Graph Neural Networks for Network Analysis

Yixuan He

Balliol College

University of Oxford

A dissertation submitted for the degree of

Doctor of Philosophy in Statistics

Trinity Term 2024

Acknowledgements

Greatest thanks to my supervisors Prof. Mihai Cucuringu and Prof. Gesine Reinert, for their insightful guidance and support. As for my source of funding, a Clarendon Scholarship from University of Oxford and a Jason Hu Scholarship from Balliol College support my research. A research placement at Amazon Shanghai AI Lablet also supports me financially, with helpful guidance from my mentor David Wipf. Amazon also supports me in the form of providing computing resources. My family and friends also offer me valuable support. Many thanks to Dr. Yu Tian, Prof. Xiaowen Dong, Prof. Yiannis Koutis, and Prof. Jiliang Tang for their constructive feedback and guidance. Besides, I am grateful to Songchao Wang and Dr. Yu Tian for their computing resources during my first year of doctoral studies. At the same time, I appreciate the collaboration with my coauthors for all my publications, especially Xitong Zhang and Prof. Michael Perlmutter.

Abstract

With an increasing number of applications where data can be represented as graphs, graph neural networks (GNNs) are a useful tool to apply deep learning to graph data. Signed and directed networks are important forms of networks that are linked to many real-world problems, such as ranking from pairwise comparisons, and angular synchronization.

In this report, we propose two spatial GNN methods for node clustering in signed and directed networks, a spectral GNN method for signed directed networks on both node clustering and link prediction, and two GNN methods for specific applications in ranking as well as angular synchronization. The methods are end-to-end in combining embedding generation and prediction without an intermediate step. Experimental results on various data sets, including several synthetic stochastic block models, random graph outlier models, and real-world data sets at different scales, demonstrate that our proposed methods can achieve satisfactory performance, for a wide range of noise and sparsity levels. The introduced models also complement existing methods through the possibility of including exogenous information, in the form of node-level features or labels.

Their contribution not only aid the analysis of data which are represented by networks, but also form a body of work which presents novel architectures and task-driven loss functions for GNNs to be used in network analysis.

Contents

List of Figures

List of Abbreviations and Some Notations

$\mathbf{A} = (A_{ij})_{i,j\in\mathcal{V}}$	Adjacency matrix
$\mathbf{C}$	Number of clusters
$\mathcal{C}_k$	Cluster k
d	Dimension
DSBM $(\mathcal{M}, \mathbb{1}(\textbf{ambient}), n, K, p, \rho, \eta)$	A Directed Stochastic Block Model (DSBM)
$\mathcal{E}$	The set of (directed) edges or links
GNN	Graph Neural Network
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, w, \mathbf{X}_\mathcal{V})$	A network with attributes. Here $\mathcal{G}$ could have self-loops but not multiple edges.
$\mathbf{K}$	Number of clusters
$\mathbf{k}$	The number of sets of angles to be recovered in the k-synchronization problem
$\mathcal{V}$	The set of nodes
$w \in (-\infty, \infty)^{\lvert\mathcal{E}\rvert}$	The set of weights of the edges
$n = \lvert\mathcal{V}\rvert$	The total number of nodes
$\mathbf{X}_\mathcal{V} \in \mathbb{R}^{n\times d_{\textbf{in}}}$	Feature matrix, whose rows are node attributes
$\mathbf{P}$	Probability matrix for cluster assignments, whose rows are cluster probabilities
$\mathbf{P}_{(:,k)}$	The k^{th} column of the matrix $\mathbf{P}$
p	Edge probability parameter in synthetic models
Pol-SSBM (n, r, p, ρ, η, N)	A polarized SSBM model
SDSBM $(\mathbf{F}, n, p, \rho, \eta)$	A Signed Directed Stochastic Block Model (SDSBM), given a meta-graph adjacency matrix $\mathbf{F} = (\mathbf{F}_{k,l})_{k,l=0,\dots,C-1}$, an edge sparsity level p, a number of nodes n, and a sign flip noise level parameter $0 \le \eta \le 0.5$

$\mathbf{SSBM}(n, K, p, \rho, \eta)$	A Signed Stochastic Block Model (SSBM)
η	Noise parameter in synthetic models, such as the sign flip probability in SSBM and Pol-SSBM.

1 Introduction

1.1 Motivation

With an increasing number of applications where data are generated from non-Euclidean domains and are represented as graphs (e.g., social networks, citation networks, and biochemical graphs), graph data, which contains rich relation information, are related to many learning tasks [1, 2]. Tasks requiring learning from graph data include, for example, predicting protein interfaces, classifying diseases, learning molecular fingerprints, and modeling physics systems [2]. Although traditional network analysis usually focuses on a single fixed simple network, which could often be represented by a symmetric adjacency matrix with nonnegative entries, more complex network types are often more realistic. In particular, signed networks, which have positive or negative edge weights, have been of interest in social network analysis and financial time-series clustering, with signs indicating positive or negative sentiments [3]. Directed networks, with asymmetric sending and receiving patterns, are also important with many applications such as clustering time-series data with lead-lag relationships, and detecting influential groups in social networks [4–6]. Dynamic networks have additionally a temporal factor, with evolving network structure and/or (node) attributes [7].

To tackle network inference tasks, graph neural networks (GNNs) are a useful tool. In essence, GNNs apply deep learning to graph data. Deep learning is very powerful in the sense that neural networks are trainable functions to make predictions. Dedicated loss functions are often constructed based on the downstream task, enabling the neural network parameters to be updated by optimizing the loss function. Neural networks are hence often flexible to train and can often attain satisfactory performance. Utilizing standard deep learning techniques such as normalization, gradient descent, and parallel computation, GNNs can be trained just like standard neural networks. By leveraging the network structure, GNNs are able to retain information from the neighborhood of nodes with long-range dependencies [2]. GNNs have a wide range of applications, such as node clustering, node embedding, link prediction, node classification, and spatial-temporal graph forecasting [1].

In this dissertation, we tackle network analysis problems in complex graphs, addressing the domain-specific challenges by incorporating customizations of the tasks at hand. By leveraging GNNs, we are able to build upon existing methods via treating their outputs as our inputs or adding learnable parameters. In this way, our GNNs can be viewed as refinements to non-GNN approaches, typically spectral methods, which cannot naturally use external information.

Here, we introduce GNN methods for node embeddings and downstream tasks such as node clustering and link prediction; for the analysis of networks, node embeddings facilitate the dimensionality reduction task and the subsequent applicability of standard machine learning methods. The new methods in this dissertation are developed around such node embeddings.

For the methods we propose, we first generate node embeddings and then transform the embeddings to fit downstream tasks, such as cluster assignment probabilities. Unlike spectral clustering methods, which generate an embedding and conduct downstream tasks such as clustering sequentially and separately, our GNN models are trained in an end-to-end manner, combining embedding generation and downstream task optimization in the same framework, with a differentiable objective function guiding the training process. The use of GNN models is often

advantageous over many spectral methods, due to the ability of GNNs to easily incorporate node features. When input node features are not given, a GNN model can also intake outputs from spectral methods as node features, such as stacking the first eigenvectors of a suitably defined Laplacian matrix. In this way, a GNN could borrow strength from such spectral methods. Another benefit of a GNN is that it could readily exploit known labels by using a supervised or semi-supervised training objective, while it is often nontrivial to incorporate supervision in spectral methods [8]. In this way, GNNs complement other methods through the possibility of incorporating exogenous information.

The proposed methods are assessed on typical tasks in network analysis. One typical task is node clustering.

Related to node clustering in networks is the task of community detection. The goal of community detection is to partition the node set of a network such that, loosely speaking, nodes within a cluster should be similar to each other, while nodes across clusters should be dissimilar [9]. Denote a (possibly signed, directed and weighted) network with node attributes as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w, \mathbf{X}_{\mathcal{V}})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of (directed) edges or links, $w \in (-\infty, \infty)^{|\mathcal{E}|}$ is the set of weights of the edges. Here $\mathcal{G}$ could have self-loops but not multiple edges. A clustering into K clusters is a partition of the node set into disjoint sets $\mathcal{V} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{K-1}$.

In a semi-supervised setting, for each of the K clusters, a fraction of training nodes are selected as seed nodes, for which the cluster membership labels are known before training. The set of seed nodes is denoted as $\mathcal{V}^{\text{seed}} \subseteq \mathcal{V}^{\text{train}} \subset \mathcal{V}$, where $\mathcal{V}^{\text{train}}$ is the set of all training nodes. The goal is then to use the embedding for assigning each node $v \in \mathcal{V}$ to a cluster containing known seed nodes. When no seeds are given, we are in a self-supervised setting, where only the number of clusters, K, is given, and the quality of the clustering may have to be judged using network information only.

The quality of a partition is often assessed through a modularity objective function [10] which compares the partition to that expected under a null model for the network, with the assumption that nodes within a cluster are relatively more densely connected than nodes across clusters. However, depending on the

task at hand, *similarity* could have different meanings. In a signed network with positive and negative edges, similarity may relate to the neighborhood of a node such as the proportion of shared friends or enemies, see Chapter 2. In a directed network, nodes may be clustered together based on their propensity to point to the same set of nodes, and be pointed by the same set of nodes, thus indicating a similar role/function in the network, see Chapter 3.

In addition to the node clustering task, the second typical task in network analysis is that of link prediction. Intuitively nodes that are close in the embedding should be more likely to be connected than nodes that are distant in the embedding. For signed and directed networks, we are faced with link prediction tasks that could relate to both link sign and directionality, as proposed in Chapter 4.

Signal recovery from pairwise measurements is related to directed graphs and could be solved by directed graph neural networks, in that the observed pairwise comparison information could be encoded in a directed network. As one real-world application of signal recovery, we are interested in recovering the global rankings based on pairwise comparisons of some competitors, where comparisons could be encoded in a directed graph. Recovering global rankings from pairwise comparisons reflecting relative latent strengths or scores is a fundamental problem in information retrieval [11, 12] and beyond. When analyzing large-scale data sets, one often seeks various forms of rankings (i.e. orderings) of the data for the purpose of identifying the most important entries, efficient computation of search & sort operations, or extracting the main features. There is a swarm of applications employing ranking techniques such as Amazon's Mechanical Turk system for crowdsourcing [13], the movie recommendation system provided by Netflix [14], and modeling outcomes of football matches [15]. Previous methods, however, are not based on neural networks. To utilize the powerful neural network architecture which is flexible to train with learnable parameters and can often achieve satisfactory performance, we hence propose a neural network method for the ranking problem. In particular, as the input data structure has an intrinsic link to a directed graph, we propose a GNN method for this real-world application, in Chapter 5.

For another application of directed graph neural networks, the group synchronization problem has received considerable attention in recent years, as a key building block of many computational problems that involve pairwise measurements with an underlying group structure. The goal of group synchronization is to estimate a collection of group elements, given a small subset of potentially noisy measurements of their pairwise ratios. An important special case is *angular synchronization*, also referred to as *phase synchronization*, which can be viewed as the group synchronization problem over the group SO(2). The angular synchronization problem aims at obtaining an accurate estimation (up to a constant additive phase) for a set of unknown angles $\theta_1, \ldots, \theta_n \in [0, 2\pi)$ from m noisy measurements of their pairwise offsets $\theta_i - \theta_j \bmod 2\pi$. This problem has a wide range of applications, such as distributed clock synchronization over wireless networks [16], image reconstruction from pairwise intensity differences [17, 18], phase retrieval [19, 20], and sensor network localization (SNL) [21]. In Chapter 6, we propose a GNN method for the angular synchronization problem as well as a heterogeneous extension.

For a GNN model, its objective plays an essential role in guiding the GNN to learn. In a node clustering/classification task, when seed nodes with known cluster labels are available, the cross-entropy loss function is usually applied. In [22], a contrastive loss function based on triplets of the nodes is used to push embeddings of nodes within clusters/classes to be closer to each other than those across clusters/classes. For self-supervised tasks, however, objectives that are independent of known labels are required. To train a GNN model, we devise task-specific loss functions. The loss function is constructed to reflect violations of rules about the data themselves: in signed clustering, positive edges should mainly exist within clusters while negative edges should mainly exist across clusters; in directed graph clustering, edges should try to adhere to the general flow in the network; in ranking from pairwise comparisons, edges should point from better players to weaker ones; in angular synchronization, edge weights should encode pairwise angular offsets, and a noiseless observed network would adhere to the cycle consistency constraint. As cluster assignment outputs can be probabilistic, a

probabilistic version of the balanced normalized cut loss is proposed in Chapter 2, while a probabilistic version of cut flow imbalance loss is introduced in Chapter 3. For other tasks such as link prediction, the cross-entropy loss function, or binary cross-entropy function, which is based on class assignment probabilities is often employed, such as in Chapter 4. For ranking, upset loss functions are proposed in Chapter 5. For angular synchronization, another upset loss function and a loss function based on cycle consistency relation are proposed in Chapter 6.

For a node clustering problem, to compare partitions, as cluster indices could be permuted, accuracy would not be an appropriate evaluation measure. Instead, the Adjusted Rand Index (ARI) [23] is often chosen. Depending on the downstream task, other measures could be used, such as the *unhappy ratio* from Chapter 2, and the *imbalance scores* from Chapter 3. For link prediction, accuracy is usually applied for evaluation, as in Chapter 4. For ranking, upset values could be considered for evaluation when no ground truth is available, and Kendall tau [24] values can be used if we have known rankings, in Chapter 5. For angular synchronization, we evaluate performance using the Mean Square Error (MSE) in general, and the Average Normalized Error (ANE) for SNL, in Chapter 6.

As for data sets to validate our proposed methods, we first generate representative synthetic data, in the form of a signed/directed stochastic block model, with possibly unequal cluster sizes but equal edge density to tackle the hardest problem of classification, as well as a polarized signed stochastic block model for signed clustering. We also propose random graph outlier models for ranking and angular synchronization. This is a way to assess the behavior of the method under a range of controlled scenarios. At the same time, we test the efficacy of our models on real-world data sets.

We have proposed a number of GNN architectures based on the task at hand. For semi-supervised node clustering in signed graphs, our method SSSNET, detailed in Chapter 2, includes a novel signed mixed-path aggregation scheme and a novel GNN architecture. For self-supervised directed graph node clustering, our approach DIGRAC, described in Chapter 3, includes a directed mixed-path aggregation

scheme. For a spectral GNN model for signed directed graphs, our model MSGNN, presented in Chapter 4, is adapted from the MagNet [6] architecture using our novel Laplacian matrix. For ranking from pairwise comparisons, our pipeline GNNRank, established in Chapter 5, borrows strength from an existing ranking baseline called SerialRank [25] and unfolds the Fiedler eigenvector computation steps. For angular synchronization, our method GNNSync, elaborated in Chapter 6, adopts ideas from GPM [26] to refine the initial angle guesses.

Theoretical guarantees provide users with a deeper understanding of the proposed methods. In Chapter 3, a type of hypothesis testing is derived to determine whether a pair of clusters is informative. In Chapter 4, we prove that our proposed magnetic signed Laplacian matrix possesses satisfactory properties. In Chapter 5, we validate the convergence of our proximal gradient steps. In Chapter 6, the behavior of the loss functions is discussed.

1.2 Contributions

Fig. 1.1 is a sketch of my doctoral contributions. My work as a graduate student includes applying GNNs to signed networks [3], directed networks [5, 6], signed directed networks [27, 28], dynamic networks [7, 29], ranking [30], angular synchronization [31], a transformer model based on energy diffusion [32], a graph sampling and filtering approach for multi-scale disentangled representations [33], and interdisciplinary collaborations including inferring metabolic states via geometric deep learning [34] and community structures in chimpanzees (work in progress). My accepted core first-author papers are: [3] by SDM 2022 on signed clustering (detailed in Chapter 2), [5] by LoG 2022 on directed clustering (detailed in Chapter 3), [27] on a spectral GNN on signed directed graphs by LoG 2022 (detailed in Chapter 4), [28] by LoG 2023 on a library and comparative survey on signed directed GNNs, [30] by ICML 2022 on recovering global rankings from pairwise comparisons (detailed in Chapter 5, and [31] by ICLR 2024 on angular synchronization (detailed in Chapter 6). In addition, I co-authored one second-author paper accepted by NeurIPS 2021 on directed networks [6], one third-author paper on temporal event

prediction accepted by LoG 2022 [29], a third-author paper on a graph sampling and filtering approach for multi-scale disentangled representations accepted by KDD 2023 [33], a third-author paper on inferring metabolic states via geometric deep learning accepted by RECOMB 2024 [34], a forth-author paper on a transformer model based on energy diffusion accepted by ICLR 2023 [32], a co-first-author paper on the generalization error of GNNs in the mean-field regime [35] accepted by ICML 2024, and one third-author paper on a temporal GNN package accepted by CIKM 2021 [7]. The temporal GNN package paper won the best paper award in CIKM 2021. The signed clustering paper [3] was accepted for an oral presentation at Complex Networks 2021. Fig. 1.1 relates different core first-author works and links them to the theme of the dissertation.

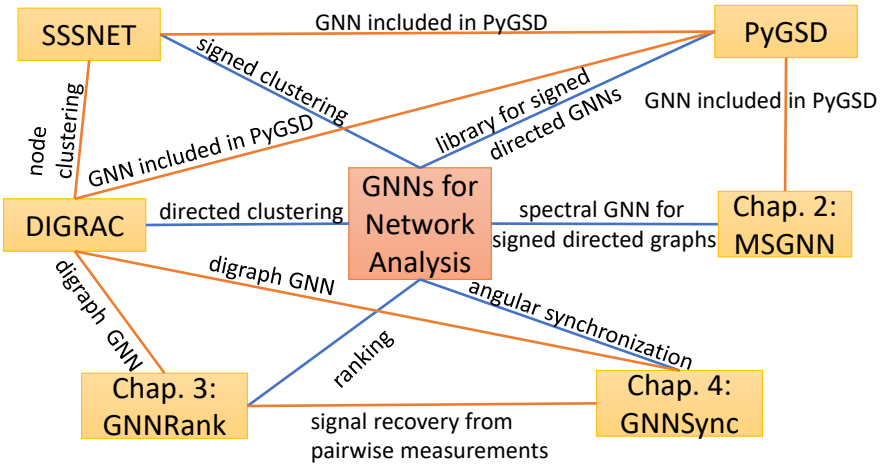


Figure 1.1: Overview of my core first-author works (orange lines) and their link to the dissertation title (blue lines).

1.3 Detailed Introduction to Doctoral Research Contributions

Next, here are more detailed backgrounds and introductions to each of the five projects presented in this thesis: SSSNET addresses the task of node clustering in signed networks; DIGRAC tackles the directed clustering task; MSGNN is a spectral GNN, designed for signed directed graphs, which is capable of solving node clustering and link prediction tasks; GNNRank is a method for recovering global rankings from pairwise comparisons; and GNNSync is an approach for angular synchronization.

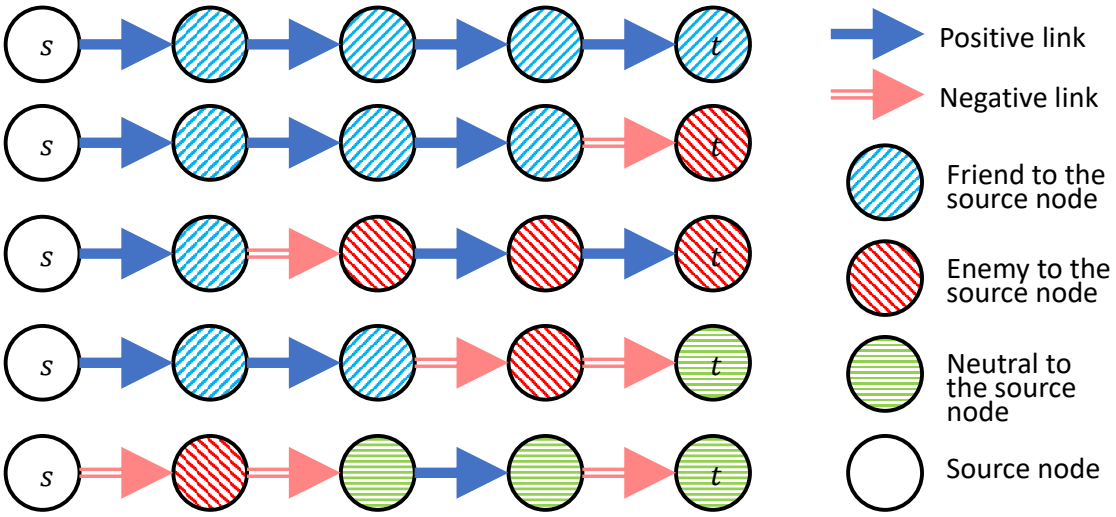
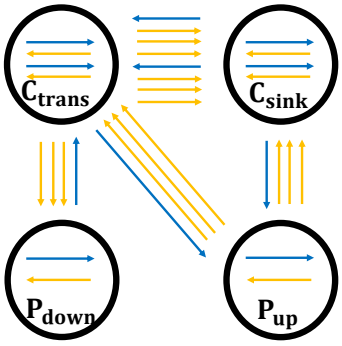


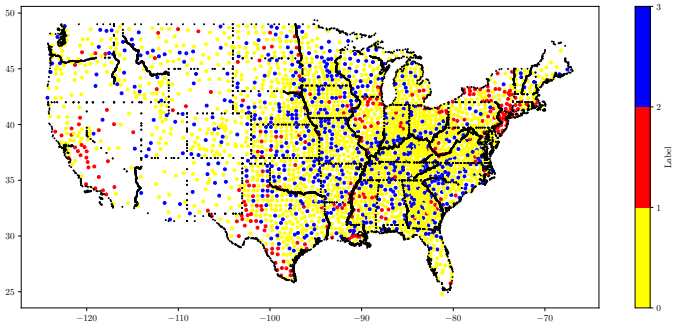
Figure 1.2: Example: five paths between the source (s) and target (t) nodes, and resulting relationships. While we assume a neutral relationship on the last two paths, social balance theory claims them as "friend" and "enemy", respectively.

1.3.1 SSSNET: Semi-Supervised Signed Network Clustering (Chapter 2)

The main novelty of our approach for signed clustering is a new take on the role of social balance theory for signed network embeddings. The standard heuristic for justifying the criteria for the embeddings hinges on the assumption that in a social network, *balanced* triangles are preferred; these are triangles such that either all three nodes are friends, or two friends have a common enemy; otherwise it would be viewed as *unbalanced*. More generally, all cycles are assumed to prefer to contain either zero or an even number of negative edges. This hypothesis is supported empirically for unsigned friendship networks, but is difficult to justify for general signed networks. For example, the relationship between trust and distrust may not be a simple negation; the enemies of enemies are not necessarily friends; an example is the social network of relations between 16 tribes of the Eastern Central Highlands of New Guinea. Besides, most state-of-the-art methods generating node embeddings of signed networks focus on link sign prediction, and those that pertain to node clustering are usually not GNN methods. In Chapter 2, we introduce a novel probabilistic balanced normalized cut loss for training nodes in a GNN framework for semi-supervised signed network clustering, called SSSNET. Figure 1.2 illustrates



(a) A meta-graph which we hypothesize to be present on *Telegram*: there are much more edge weights flow from $\mathcal{C}_{\text{trans}}$ to $\mathcal{C}_{\text{sink}}$ than the other direction.



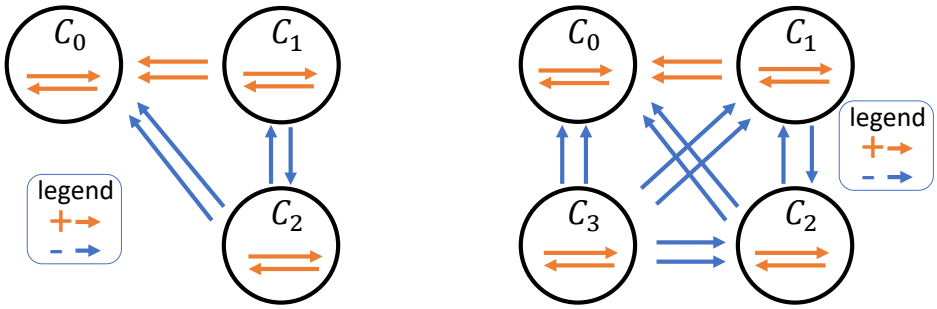
(b) Strongest imbalanced flow on *Migration* data, along with the geographic locations of the counties and state boundaries (in black): most edges flow from red (1) to blue (2).

Figure 1.3: Visualization of directed flow imbalance.

five different paths of length four, connecting the source and the target nodes. We can also obtain the relationship of a source node to a target node within a path by reversing the arrows in Figure 1.2. Note that it is possible for a node to be both a "friend" and an "enemy" to a source node simultaneously, as there might be multiple paths between them, with different resulting relationships. Our model aggregates these relationships by assigning different weights to different paths connecting two nodes. For example, the source node and target node may have all five paths shown in Figure 2.2 connecting them. Since the last two paths are neutral paths and do not cast a vote on their relationship, we only take the top three paths into account.

1.3.2 DIGRAC: Digraph Clustering Based on Flow Imbalance (Chapter 3)

The main novelty of our directed clustering approach is an objective based on flow imbalance. While most existing methods that could be applied to directed clustering use local edge densities as the main signal and directionality as an additional signal, we argue that even in the absence of any edge density differences, directionality can play a vital role in directed clustering as it can reveal latent properties of network flows. Therefore, instead of finding relatively dense groups of nodes in digraphs that have a relatively small amount of flow between the groups, our main



(a) Signed Directed Stochastic Block Model example 1. **(b)** Signed Directed Stochastic Block Model example 2.

Figure 1.4: This toy example models groups of athletes and sports fans on social media. Here, signed, directed edges represent positive or negative mentions. Cluster $\mathcal{C}_0$: players of a sports team; $\mathcal{C}_1$: a group of their fans who typically say positive things about the players; $\mathcal{C}_2$: a group of fans of a rival team; $\mathcal{C}_3$: a group of fans of a third, less important team.

goal is to recover clusters with *strong and imbalanced flow* among them, where directionality is the main signal. In contrast to standard approaches focusing on edge density, here edge directionality is not a nuisance but the main piece of information to uncover the latent structure. Figure 3.1a plots a meta-graph which we hypothesize to be present for *Telegram* [4], where most edge weights flow from the core-transient cluster ($\mathcal{C}_{\text{trans}}$) to the core-sink cluster ($\mathcal{C}_{\text{sink}}$) than the other direction. As another real-world example, Figure 3.1b shows the strongest flow imbalances between clusters in a network of US migration flow [36]; most edges flow from the red cluster (1) to the blue one (2).

1.3.3 MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian (Chapter 4)

Signed and directed networks are ubiquitous in real-world applications. However, there has been relatively little work proposing spectral GNN for such networks. In [27] we introduce a signed directed Laplacian matrix, which we call the magnetic signed Laplacian, as a natural generalization of both the signed Laplacian on signed graphs and the magnetic Laplacian [6] on directed graphs. We then use this matrix to construct a novel efficient spectral GNN architecture and conduct

extensive experiments on both node clustering and link prediction tasks. In these experiments, we consider tasks related to signed information, tasks related to directional information, and tasks related to both signed and directional information. We demonstrate that our proposed spectral GNN is effective for incorporating both signed and directional information, and attains leading performance on a wide range of data sets. Additionally, we provide a novel synthetic network model, which we refer to as the Signed Directed Stochastic Block Model (SDSBM), and several novel real-world data sets based on lead-lag relationships in financial time series.

Here we provide a toy example to better understand why considering both signed and directional information is essential. In Figure 1.4(a), $\mathcal{C}_0$ are the players of a sports team, $\mathcal{C}_1$ is a group of their fans who typically say positive things about the players, and $\mathcal{C}_2$ is a group of fans of a rival team, who typically say negative things about the players. Since they are fans of rival teams, the members of $\mathcal{C}_1$ and $\mathcal{C}_2$ both say negative things about each other. In general, fans mention the players more than players mention the fans, which leads to a net flow imbalance. In Figure 1.4(b), we add in $\mathcal{C}_3$, a group of fans of a third, less important team. This group dislikes the other two teams and disseminates negative content about $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$. However, since this third team is quite unimportant, no one comments anything back.

Notably, in both examples, as the expected edge density is identical both within and across clusters, discarding either signed or directional information will ruin the clustering structure. For instance, in both examples, if we discard directional information, then $\mathcal{C}_0$ will look identical to $\mathcal{C}_1$ in the resulting meta-graph. On the other hand, if we discard signed information, $\mathcal{C}_1$ will look identical to $\mathcal{C}_2$. This shows the necessity for a model to consider both signed and directional information simultaneously.

1.3.4 Signal Recovery with Directed Graph Interpretations (Chapter 5 and Chapter 6)

Given that pairwise comparisons could be encoded into a digraph, I have, with my collaborators, introduced two deep learning methods based on directed graph

neural networks for signal recovery problems starting from pairwise comparisons.

Ranking. Recovering global rankings from pairwise comparisons has wide applications from time synchronization to sports team ranking. Pairwise comparisons corresponding to matches in a competition can be construed as edges in a directed graph, whose nodes represent e.g. competitors with an unknown rank. In Chapter 5, we introduce neural networks into the ranking recovery problem by proposing the so-called GNNRank, a trainable GNN-based framework with digraph embedding. New objectives are devised to encode ranking upsets/violations. The framework involves a ranking score estimation approach and adds an inductive bias by unfolding the Fiedler vector computation of the graph constructed from a learnable similarity matrix. Experimental results on extensive data sets show that our methods attain competitive and often superior performance against baselines, as well as showing promising transfer ability when applying the trained model to similar data sets.

Angular Synchronization. The angular synchronization problem aims to accurately estimate (up to a constant additive phase) a set of unknown angles $\theta_1, \dots, \theta_n \in [0, 2\pi)$ from m noisy measurements of their offsets $\theta_i - \theta_j \bmod 2\pi$. Applications include, for example, sensor network localization, phase retrieval, and distributed clock synchronization. An extension of the problem to the heterogeneous setting (dubbed k-synchronization) is to estimate k groups of angles simultaneously, given noisy observations (with unknown group assignment) from each group. Existing methods for angular synchronization usually perform poorly in high-noise regimes, which are common in applications. In Chapter 6, we leverage neural networks for the angular synchronization problem, and its heterogeneous extension, by proposing GNNSync, a theoretically-grounded end-to-end trainable framework using directed graph neural networks. Novel loss functions are devised to encode synchronization objectives. Experimental results on extensive data sets demonstrate that GNNSync attains competitive, and often superior, performance against a comprehensive set of baselines for the angular synchronization problem and its extension, validating the robustness of GNNSync even at high noise levels.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows: In Chapter 2, a GNN framework named SSSNET for signed network clustering is introduced. Chapter 3 details our directed graph clustering GNN framework called DIGRAC. Chapter 4 presents a spectral GNN method for signed directed networks. In Chapter 5, a GNN model named GNNRank is described for ranking, while Chapter 6 details a GNN method called GNNSync for angular synchronization. In Chapter 7, we briefly introduce PyTorch Geometric Signed Directed, a Python library on signed directed GNNs with a comparative survey; MagNet, a spectral GNN for directed graphs based on a complex Hermitian matrix known as the magnetic Laplacian; PyTorch Geometric Temporal, a deep learning framework combining state-of-the-art machine learning algorithms for neural spatiotemporal signal processing; Difformer, a transformer model based on energy diffusion; CEP3, an event prediction method on dynamic graphs; PyGNN, a graph sampling and filtering approach for multi-scale disentangled representations; GEFMAP, a method based on geometric deep learning for predicting flux through reactions in a global metabolic network using transcriptomics data; as well as a theoretical framework for assessing the generalization error of GNNs in the over-parameterized regime. Lastly, Chapter 8 draws conclusions and discusses future directions. The appendix (supplementary information) covers implementation details, data description, variants of the models, and theoretical results, as well as extended results in the main body of this dissertation.

Chapter 2, Chapter 3, Chapter 4, Chapter 5, and Chapter 6 are stand-alone papers which are either published. The code for Chapter 2 is found at `https://github.com/SherylHYX/SSSNET_Signed_Clustering`, for Chapter 3 is found at `https://github.com/SherylHYX/DIGRAC_Directed_Clustering`, for Chapter 4 is found at `https://github.com/SherylHYX/MSGNN`, for Chapter 5 is found at `https://github.com/SherylHYX/GNNRank`, and for Chapter 6 is found at `https://github.com/SherylHYX/GNN_Sync`. References for these papers can be found before the appendix.

2

SSSNET: Semi-Supervised Signed Network Clustering

2.1 Introduction

In social network analysis, signed network clustering is an important task. Signs of edges in networks may indicate positive or negative sentiments, see for example [37]. Users may express trust-distrust or friendship-enmity. Review websites as well as online news allow users to approve or denounce others [38]. Clustering time series can be viewed as an instance of signed network clustering [39], with the empirical correlation matrix being construed as a weighted signed network. Recommendation systems provide another playground for signed networks; [40] introduced a principled approach to capturing local and global information from signed social networks mathematically, and proposed a novel recommendation framework. Furthermore, there has been a recent growing interest on the topic of polarization in social media, mainly fueled by a large variety of speeches and statements made in the pursuit of public good, and their impact on the integrity of democratic processes [41]; our work also contributes to the growing literature of polarization in signed networks.

Most competitive state-of-the-art methods generating node embeddings for signed networks focus on link sign prediction [42–48], and those that pertain to node clustering are not GNN methods [44, 49–52]. Here, we introduce a graph neural

network (GNN) framework, called SSSNET, with a *Signed Mixed-Path Aggregation* (SIMPA) scheme, to obtain node embeddings for signed clustering.

The main novelty of our approach is a new take on the role of social balance theory for signed network embeddings. The standard heuristic for justifying the criteria for the embeddings hinges on the assumption that "an enemy's enemy is a friend" [44, 48, 53–55]. This heuristic is based on social balance theory [56, 57], or *multiplicative distrust propagation* as in [58], which asserts that in a social network, in a triangle either all three nodes are friends, or two friends have a common enemy; otherwise it would be viewed as *unbalanced*. More generally, all cycles are assumed to prefer to contain either zero or an even number of negative edges. This hypothesis has been supported for the analysis of unsigned friendship networks, but is difficult to justify for general signed networks. For example, the relationship between trust and distrust may not be a simple negation; the enemies of enemies are not necessarily friends, see [58, 59]; an example is given by the social network of relations between 16 tribes of the Eastern Central Highlands of New Guinea [60]. Hence, the present work takes a neutral stance on whether or not the enemy of an enemy is a friend, thus generalizing the *atomic propagation* by [58].

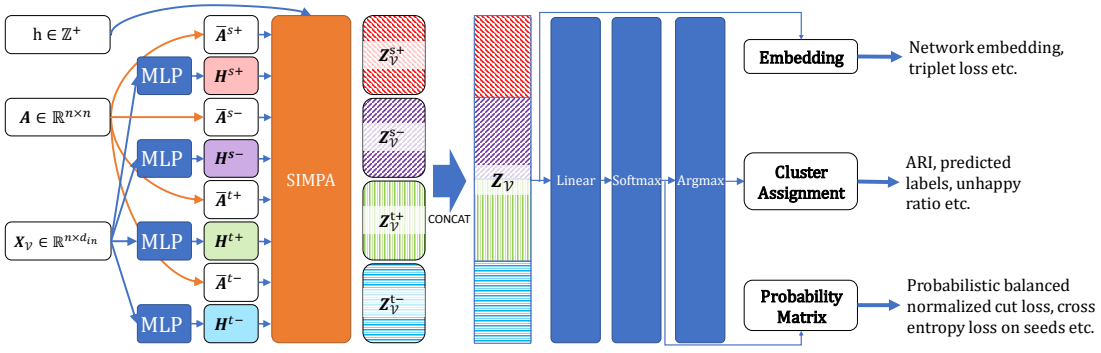


Figure 2.1: SSSNET overview: starting from feature matrix $\mathbf{X}_{\mathcal{V}}$ and adjacency matrix $\mathbf{A}$, we first compute the row-normalized adjacency matrices $\overline{\mathbf{A}}^{s+}, \overline{\mathbf{A}}^{s-}, \overline{\mathbf{A}}^{t+}, \overline{\mathbf{A}}^{t-}$. We then apply four separate MLPs on $\mathbf{X}_{\mathcal{V}}$, to obtain hidden representations $\mathbf{H}^{s+}, \mathbf{H}^{s-}, \mathbf{H}^{t+}$, $\mathbf{H}^{t-}$, respectively. Next, we compute their decoupled embeddings via Eq. (2.1) and its equivalent for negative/target embeddings. The concatenated decoupled embeddings are the final embeddings. We add another linear layer followed by a unit softmax function to obtain the probability matrix $\mathbf{P}$. Applying argmax to each row of $\mathbf{P}$ yields cluster assignments for all nodes.

From a method's viewpoint, the neutral stance is reflected in the feature aggregation in SIMPA, which provides the basis of the network embedding. Network clustering is then carried out using the node embedding as input and a loss function for training; see Figure 2.1 for an overview. To train SSSNET, the loss function consists of a self-supervised novel probabilistic balanced normalized cut loss acting on all training nodes, and a supervised loss which acts on seed nodes, if available. Experimental results at different scales demonstrate that our method achieves state-of-the-art performance on synthetic data, for a wide range of network densities, while complementing other methods through the possibility of incorporating exogenous information. Tested on real-world data for which ground truth is available, our method outperforms its competitors in terms of the Adjusted Rand Index [23].

Main contributions. Our main contributions are as follows: • (1) We propose an efficient end-to-end GNN for semi-supervised signed node clustering, based on a new variant of social balance theory. To the best of our knowledge, this is the first GNN-based method deriving node embeddings for clustering signed networks, potentially with attributes. The advantage of the ability to handle features is that we can incorporate eigen-features of other methods (such as eigenvectors of the signed Laplacian), thus borrowing strength from existing methods. • (2) We propose a *Signed Mixed-Path Aggregation* (SIMPA) framework based on our new take on social balance theory. • (3) We propose a probabilistic version of balanced normalized cut to serve as a self-supervised loss function for signed clustering. • (4) We achieve state-of-the-art performance on various signed clustering tasks, including a challenging version of a classification task, for which we customize and adapt a general definition of polarized signed stochastic block models (Pol-SSBM), to include an ambient cluster and multiple polarized SSBM communities, not necessarily of equal size, but of equal density. Code and preprocessed data are available at `https://github.com/SherylHYX/SSSNET_Signed_Clustering`.

2.2 Related Work

2.2.1 Network Embedding and Clustering

We introduce a semi-supervised method for node embeddings, which uses the idea of an aggregator and relies on powers of adjacency matrices for such aggregators. The use of an aggregation function is motivated by [61]. The use of powers of adjacency matrices for neighborhood information aggregation was sparked by a mechanism for node feature aggregation, proposed in [22], with a data-driven similarity metric during training.

Non-GNN methods have been employed for signed clustering. [50] utilizes the signed Laplacian matrix and its normalized versions for signed clustering. [62] clusters signed networks using the geometric mean of Laplacians. In [51], nodes are clustered based on optimizing the Balanced Normalized Cut and the Balanced Ratio Cut. [63] develops two normalized signed Laplacians based on so-called SNScut and BNScut. [52] relies on a generalized eigenproblem and achieves state-of-the-art performance on signed clustering. To conduct semi-supervised structural learning on signed networks, [64] uses variational Bayesian inference and [65] devises an MBO scheme. [66] tackles network sparsity. However, these prior works do not take node attributes into account. [49] exploits the network structure and node attributes simultaneously, but does not utilize known labels. [44] views relationship formation between users as the comprehensive effects of latent factors and trust transfer patterns, via social balanace theory.

Graph neural networks have also been utilized for signed network embedding tasks, but not for signed clustering. SGCN [48] utilizes social balance theory to aggregate and propagate the information across layers. Considering interactions between positive and negative edges jointly is another main inspiration for our method, but SSSNET is not driven by such social balance theory principles. Many other GNNs [43, 45, 47, 54, 55] are also based on social balance theory, usually applied to data with strong positive class imbalance. Numerous other signed network

embedding methods [42, 44, 46, 53, 67] also do not explore the node clustering problem. Hence we do not employ these methods for comparison.

2.2.2 Polarization

Opinion formation in a social network can be driven by a few small groups which have a polarized opinion, in a social network in which many agents do not (yet) have a strongly formed opinion. These small clusters could strongly influence the public discourse and even threaten the integrity of democratic processes. Detecting such small clusters of polarized agents in a network of ambient nodes is hence of interest [41]. For a network with node set $\mathcal{V}$, [41] introduces the notion of a polarized community structure $(\mathcal{C}_1, \mathcal{C}_2)$ within the wider network, as two disjoint sets of nodes $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{V}$, such that • (1) there are relatively few (resp. many) negative (resp. positive) edges within $\mathcal{C}_1$ and within $\mathcal{C}_2$; • (2) there are relatively few (resp. many) positive (resp. negative) edges across $\mathcal{C}_1$ and $\mathcal{C}_2$; • (3) there are relatively few edges (of either sign) from $\mathcal{C}_1$ and $\mathcal{C}_2$ to the rest of the graph. By allowing a subset of nodes to be neutral with respect to the polarized structure, [68] derives a formulation in which each cluster inside a polarized community is naturally characterized by the solution to the maximum discrete Rayleigh's quotient (MAX-DRQ) problem. However, this model cannot incorporate node attributes. Extending the approach in [69] and [52], here, a community structure $(\mathcal{C}_1, \mathcal{C}_2)$ is said to be *polarized* if (1) and (2) hold, while (3) is not required to hold. Moreover, our model includes different parameters for the noise and edge probability.

2.3 The SSSNET Method

2.3.1 Problem Definition

Denote a signed (possibly directed and weighted) network with node attributes as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w, \mathbf{X}_\mathcal{V})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of (directed) edges or links, $w \in (-\infty, \infty)^{|\mathcal{E}|}$ is the set of weights of the edges. Here $\mathcal{G}$ could have self-loops but not multiple edges. The total number of nodes is $n = |\mathcal{V}|$, and $\mathbf{X}_\mathcal{V} \in \mathbb{R}^{n \times d_{\text{in}}}$ is a matrix whose rows are node attributes (which could be generated

from $\mathbf{A}$). This network can be represented by the attribute matrix $\mathbf{X}_{\mathcal{V}}$ and the adjacency matrix $\mathbf{A} = (A_{ij})_{i,j\in\mathcal{V}}$, where $\mathbf{A}_{ij} = w_{ij}$, the edge weight, if there is an edge between nodes v_i and v_j; otherwise $\mathbf{A}_{ij} = 0$. We decompose the adjacency matrix $\mathbf{A}$ into positive and negative parts $\mathbf{A}^+$ and $\mathbf{A}^-$, where $\mathbf{A}^+_{ij} = \max(\mathbf{A}_{ij}, 0)$ and $\mathbf{A}^-_{ij} = -\min(\mathbf{A}_{ij}, 0)$. A clustering into K clusters is a partition of the node set into disjoint sets $\mathcal{V} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{K-1}$. Intuitively, nodes within a cluster should be similar to each other, while nodes across clusters should be dissimilar. In a semi-supervised setting, for each of the K clusters, a fraction of training nodes are selected as seed nodes, for which the cluster membership labels are known before training. The set of seed nodes is denoted as $\mathcal{V}^{\text{seed}} \subseteq \mathcal{V}^{\text{train}} \subset \mathcal{V}$, where $\mathcal{V}^{\text{train}}$ is the set of all training nodes. For this task, the goal is to use the embedding for assigning each node $v \in \mathcal{V}$ to a cluster containing known seed nodes. When no seeds are given, we are in a self-supervised setting, where only the number of clusters, K, is given.

2.3.2 Path-Based Node Relationship

Methods based on social balance theory assume that, given a negative relationship between v_1, v_2 and a negative relationship between v_2, v_3, the nodes v_1 and v_3 should be positively related. This assumption may be sensible for social networks, but in other networks such as correlation networks [39, 70], it is not obvious why it should hold. Indeed, the column $|\Delta^{\text{u}}|$ in Table 2.1 counts the number of triangles with an odd number of negative edges in eight real-world data sets and one synthetic model. We observe that $|\Delta^{\text{u}}|$ is never zero, and that in some cases, the percentage of unbalanced triangles (the last column) is quite large, such as in Sampson's network of novices and the simulated SSBM($n = 5000, K = 5, p = 0.1, \rho = 1.5$) ("Syn" in Table 2.1, SSBM is defined in Section 2.4.1). The relative high proportion of unbalanced triangles sparks our novel approach. SSSNET holds a neutral attitude towards the relationship between v_1 and v_3. In contrast to social balance theory, our definition of "friends" and "enemies" is based on the set of paths within a given length between any two nodes. For a target node v_j to be a h-hop "friend" neighbor of source node v_i *along a given path* from v_i to v_j of length h, all edges on this

path need to be positive. For a target node v_j to be an h-hop “enemy" neighbor of source node v_i along a given path from v_i to v_j of length h, exactly one edge on this path has to be negative. Otherwise, v_i and v_j are neutral to each other on this path. For directed networks, only directed paths are taken into account, and the friendship relationship is no longer symmetric.

Figure 2.2 illustrates five different paths of length four, connecting the source and the target nodes. We can also obtain the relationship of a source node to a target node within a path by reversing the arrows in Figure 2.2. Note that it is possible for a node to be both a “friend" and an “enemy" to a source node simultaneously, as there might be multiple paths between them, with different resulting relationships. Our model aggregates these relationships by assigning different weights to different paths connecting two nodes. For example, the source node and target node may have all five paths shown in Figure 2.2 connecting them. Since the last two paths are neutral paths and do not cast a vote on their relationship, we only take the top three paths into account. We refer to the long-range neighbors whose information would be considered by a node as the *contributing neighbors* with respect to the node of interest.

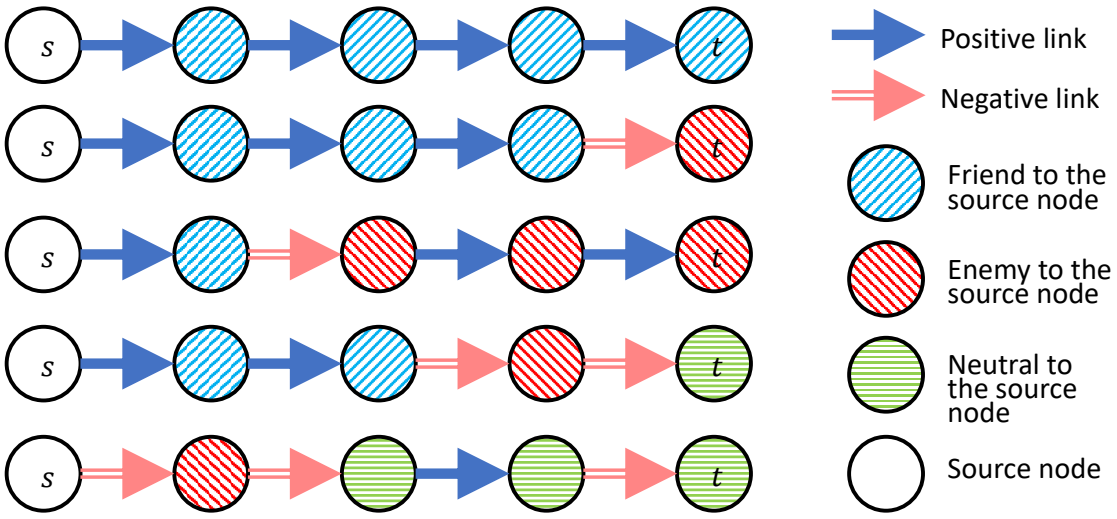


Figure 2.2: Example: five paths between the source (s) and target (t) nodes, and resulting relationships. While we assume a neutral relationship on the last two paths, social balance theory claims them as "friend" and "enemy", respectively.

2.3.3 Signed Mixed-Path Aggregation (SIMPA)

SIMPA aggregates neighbor information from contributing neighbors within h hops, by a weighted average of the embeddings of the up-to-h-hop contributing neighbors of a node, with the weights constructed in analogy to a random walk on the set of nodes.

SIMPA Matrices

First, we row-normalize the positive and negative parts of the adjacency matrix, $\mathbf{A}^+$ and $\mathbf{A}^-$, to obtain matrices $\overline{\mathbf{A}}^{s+}$ and $\overline{\mathbf{A}}^{s-}$, respectively. Inspired by the regularization discussed in [71], we add a weighted self-loop to each node and carry out the normalization by setting $\overline{\mathbf{A}}^{\mathbf{s}+} = (\tilde{\mathbf{D}}^{s+})^{-1}\tilde{\mathbf{A}}^{s+}$, where $\tilde{\mathbf{A}}^{s+} = \mathbf{A}^+ + \tau^+\mathbf{I}$ and the diagonal matrix $\tilde{\mathbf{D}}^{s+}(i,i) = \sum_j \tilde{\mathbf{A}}^{s+}(i,j)$, for some $\tau^+ \geq 0$; similarly, we row-normalize $\mathbf{A}^{s-}$ to obtain $\overline{\mathbf{A}}^{s-}$ based on τ^-. *Next*, we explore multi-hop neighbors by taking powers or mixed powers of $\overline{\mathbf{A}}^{s+}$ and $\overline{\mathbf{A}}^{s-}$. The h-hop "friend" neighborhood can be computed directly from $(\overline{\mathbf{A}}^{s+})^h$, the h power of $\overline{\mathbf{A}}^{s+}$. Similarly, the h-hop "enemy" neighborhood can be computed directly from the mixed powers of $h-1$ terms of $\overline{\mathbf{A}}^{s+}$, and exactly one term of $\overline{\mathbf{A}}^{s-}$. As multiplication of $\overline{\mathbf{A}}^{s+}$ and $\overline{\mathbf{A}}^{s-}$ may not necessarily commute, we keep all h "enemy" neighborhood matrices to aggregate "enemy" information. We denote the set of *up-to-h-hop* "friend" neighborhood matrices as $\mathcal{A}^{s+,h} = \{(\overline{\mathbf{A}}^{s+})^{h_1} : h_1 \in \{0, \cdots, h\}\}$, where $(\overline{\mathbf{A}}^{s+})^0 = \mathbf{I}$, the identity matrix, and the set of up-to-h-hop "enemy" neighborhood matrices as

$$\mathcal{A}^{s-,h} = \left\{(\overline{\mathbf{A}}^{s+})^{h_1} \cdot \overline{\mathbf{A}}^{s-} \cdot (\overline{\mathbf{A}}^{s+})^{h_2} : h_1, h_2 \in H\right\}$$

with $H = \{(h_1, h_2) : h_1, h_2 \in \{0, \ldots, h-1\}, h_1 + h_2 \leq h-1\}$. With added self-loops, any h-hop neighbor defined by our matrices *aggregates beliefs* from nearby neighbors. Since a node is not an enemy to itself, we set $\tau^- = 0$. We use $\tau^+ = \tau = 0.5$ in our experiments.

When the signed network is directed, we additionally carry out ℓ_1 row normalization and calculate mixed powers for $(\mathbf{A}^+)^T$ and $(\mathbf{A}^-)^T$. We denote the row-normalized adjacency matrices for *target* positive and negative as $\overline{\mathbf{A}}^{t+}$ and $\overline{\mathbf{A}}^{t-}$, respectively. Likewise, we denote the set of up-to-h-hop target "friend" (resp. "enemy") neighborhood matrices as $\mathcal{A}^{t+,h}$ (resp. $\mathcal{A}^{t-,h}$).

Feature Aggregation Based on SIMPA

Next, we define four feature-mapping functions for source positive, source negative, target positive and target negative embeddings, respectively. A source positive embedding of a node is the weighted combination of its contributing neighbors' hidden representations, for neighbors up to h hops away. The source positive hidden representation is denoted as $\mathbf{H}_{\mathcal{V}}^{s+} \in \mathbb{R}^{n\times d}$. Assume that each node in $\mathcal{V}$ has a vector of features and summarize these features in the input feature matrix $\mathbf{X}_{\mathcal{V}}$. The source positive embedding $\mathbf{Z}_{\mathcal{V}}^{s+}$ is given by

$$\mathbf{Z}_{\mathcal{V}}^{s+} = \sum_{\mathbf{M}\in\mathcal{A}^{s+,h}} \omega_{\mathbf{M}}^{s+} \cdot \mathbf{M} \cdot \mathbf{H}_{\mathcal{V}}^{s+} \in \mathbb{R}^{n\times d}, \tag{2.1}$$

where for each $\mathbf{M}$, $\omega_{\mathbf{M}}^{s+}$ is a learnable scalar, and d is the embedding dimension. In our experiments, we use $\mathbf{H}_{\mathcal{V}}^{s+} = \mathbf{MLP}^{(s+,l)}(\mathbf{X}_{\mathcal{V}})$. The hyperparameter l controls the number of layers in the multilayer perceptron (MLP) with ReLU activation; we fix $l = 2$ throughout. Each layer of the MLP has the same number d of hidden units.

The embeddings $\mathbf{Z}_{\mathcal{V}}^{s-}$, $\mathbf{Z}_{\mathcal{V}}^{t+}$ and $\mathbf{Z}_{\mathcal{V}}^{t-}$ for source negative embedding, target positive embedding and target negative embedding, respectively, are defined similarly. Different parameters for the MLPs for different embeddings are possible. We concatenate the embeddings to obtain the final node embedding as a $n \times (4d)$ matrix $\mathbf{Z}_{\mathcal{V}} = \text{CONCAT}\left(\mathbf{Z}_{\mathcal{V}}^{s+}, \mathbf{Z}_{\mathcal{V}}^{s-}, \mathbf{Z}_{\mathcal{V}}^{t+}, \mathbf{Z}_{\mathcal{V}}^{t-}\right)$. The embedding vector $\mathbf{z}_i$ for a node v_i, is the i^{th} row of $\mathbf{Z}_{\mathcal{V}}$, namely $\mathbf{z}_i := (\mathbf{Z}_{\mathcal{V}})_{(i,:)} \in \mathbb{R}^{4d}$. Next we apply a linear layer to $\mathbf{Z}_{\mathcal{V}}$ so that the resulting matrix has the same number of columns as the number K of clusters. We apply the unit *softmax* function to map each row to a probability vector $\mathbf{p}_i \in \mathbb{R}^K$ of length equal to the number of clusters, with entries denoting the probabilities of each node to belong to each cluster. The resulting probability matrix is denoted as $\mathbf{P} \in \mathbb{R}^{n\times K}$. If the input network is undirected, it suffices to find $\mathcal{A}^{s+,h}$ and $\mathcal{A}^{s-,h}$, and we obtain the final embedding as $\mathbf{Z}_{\mathcal{V}} = \text{CONCAT}\left(\mathbf{Z}_{\mathcal{V}}^{s+}, \mathbf{Z}_{\mathcal{V}}^{s-}\right) \in \mathbb{R}^{n\times(2d)}$.

2.3.4 Loss, Overview & Complexity Analysis

Node clustering is optimized to minimize a loss function which pushes embeddings of nodes within the same cluster close to each other, while driving apart embeddings

of nodes from different clusters. We first introduce a novel self-supervised loss function for node clustering, then discuss supervised loss functions when labels are available for some seed nodes.

Probabilistic Balanced Normalized Cut Loss

For a clustering $(\mathcal{C}_0, \dots \mathcal{C}_{K-1})$, let $\{\mathbf{x}_0, \cdots, \mathbf{x}_{K-1}\}$ denote the cluster indicator vectors so that $\mathbf{x}_k(i) = 1$ if node i is in cluster C_k, and 0 otherwise. Let $\mathbf{L}^+ = \mathbf{D}^+ - \mathbf{A}^+$ denote the unnormalized graph Laplacian for the positive part of $\mathbf{A}$, where $\mathbf{D}^+$ is a diagonal matrix whose diagonal entries are row-sums of $\mathbf{A}^+$. Then $\mathbf{x}_k^T \mathbf{L}^+ \mathbf{x}_k$ measures the total weight of positive edges linking cluster $\mathcal{C}_k$ to *other* clusters. Further, $\mathbf{x}_k^T \mathbf{A}^- \mathbf{x}_k$ measures the total weight of negative edges *within* cluster $\mathcal{C}_k$. Since $\mathbf{D}^+ - \mathbf{A} = \mathbf{D}^+ - \mathbf{A}^+ + \mathbf{A}^-$, then $\mathbf{x}_k^T(\mathbf{L}^+ + \mathbf{A}^-)\mathbf{x}_k = \mathbf{x}_k^T(\mathbf{D}^+ - \mathbf{A})\mathbf{x}_k$ measures the total weight of the *unhappy* edges with respect to cluster $\mathcal{C}_k$; "unhappy edges" violate their expected signs (positive edges across clusters or negative edges within clusters). The loss function in this paper is related to the (non-differentiable) Balanced Normalized Cut (BNC) [51]. In analogy, we introduce the differentiable *Probabilistic Balanced Normalized Cut (PBNC) loss*

$$\mathcal{L}_{\text{PBNC}} = \sum_{k=1}^{K} \frac{(\mathbf{P}_{(:,k)})^T (\mathbf{D}^+ - \mathbf{A}) \mathbf{P}_{(:,k)}}{(\mathbf{P}_{(:,k)})^T \overline{\mathbf{D}} \mathbf{P}_{(:,k)}}, \tag{2.2}$$

where $\mathbf{P}_{(:,k)}$ denotes the k^{th} column of the probability matrix $\mathbf{P}$ and $\overline{\mathbf{D}}_{ii} = \sum_{j=1}^{n} |A_{ij}|$. As column k of $\mathbf{P}$ is a relaxed version of $\mathbf{x}_k$, the numerator in Eq. (2.2) is a probabilistic count of the number of unhappy edges.

Supervised Loss

When some seed nodes have known labels, a supervised loss can be added to the loss function. For nodes in $\mathcal{V}^{\text{seed}}$, we use as a supervised loss function similar to that in [22], the sum of a cross-entropy loss $\mathcal{L}_{\text{CE}}$ and a triplet loss. The triplet loss is

$$\mathcal{L}_{\text{triplet}} = \frac{1}{|\mathcal{T}|} \sum_{(v_i, v_j, v_k) \in \mathcal{T}} ReLU(\text{CS}(\mathbf{z}_i, \mathbf{z}_j) - \text{CS}(\mathbf{z}_i, \mathbf{z}_k) + \alpha), \tag{2.3}$$

where $\mathcal{T} \subseteq \mathcal{V}^{\text{seed}} \times \mathcal{V}^{\text{seed}} \times \mathcal{V}^{\text{seed}}$ is a set of node triplets: v_i is an anchor seed node, and v_j is a seed node from the same cluster as the anchor, while v_k is from a different

cluster. Here, $\mathrm{CS}(\mathbf{z}_i, \mathbf{z}_j)$ is the cosine similarity of the embeddings of nodes v_i and v_j, chosen so as to avoid sensitivity to the magnitude of the embeddings. $\alpha \geq 0$ is the contrastive margin as in [22]. $\mathcal{L}_{\mathrm{CE}} + \gamma_t \mathcal{L}_{\mathrm{triplet}}$ forms the supervised part of the loss function for SSSNET, for a suitable parameter $\gamma_t > 0$.

Overall Objective Function and Framework Overview

By combining $\mathcal{L}_{\mathrm{CE}}$, Eq. (2.2), and Eq. (2.3), the objective function minimizes

$$\mathcal{L} = \mathcal{L}_{\mathrm{PBNC}} + \gamma_s(\mathcal{L}_{\mathrm{CE}} + \gamma_t \mathcal{L}_{\mathrm{triplet}}), \tag{2.4}$$

where $\gamma_s, \gamma_t > 0$ are weights for the supervised part of the loss and triplet loss, respectively. The final embedding can then be used, for example, for node clustering. A linear layer coupled with a unit softmax function turns the embedding into a probability matrix. A node is assigned to the cluster for which its membership probability is highest. Figure 2.1 gives an overview.

Complexity Analysis

The matrix operations in Eq. (2.1) appear to be computationally expensive and space unfriendly. However, SSSNET resolves these concerns via a sparsity-aware implementation, detailed in Algorithm 1 in SI A.3.1, without explicitly calculating the sets of powers, maintaining sparsity throughout. Therefore, for input feature dimension d_{in} and hidden dimension d, if $d' = \max(d_{\mathrm{in}}, d) \ll n$, time and space complexity of SIMPA, and implicitly SSSNET, is $\mathcal{O}(|\mathcal{E}|d'h^2 + 4nd'K)$ and $\mathcal{O}(4|\mathcal{E}| + 10nd' + nK)$, respectively [72]. For large networks, SIMPA is amenable to a more scalable version following [73].

2.4 Experiments

This section describes the synthetic and real-world data sets used in this study, and illustrates the efficacy of our method. When ground truth is available, performance is measured by the Adjusted Rand Index (ARI) [23]. When no labels are provided, we measure performance by the ratio of number of “unhappy edges" to that of all edges. Our self-supervised loss function is applied to the subgraph induced by

all training nodes. We do not report Normalized Mutual Information (NMI) [74] performance in Figure 2.5 (but reported in SI A.1.1) as it has some shortcomings [75], and results from the ARI and NMI from our synthetic experiments indeed yield almost the same ranking for the methods, with average Kendall tau 0.808 and standard deviation 0.233.

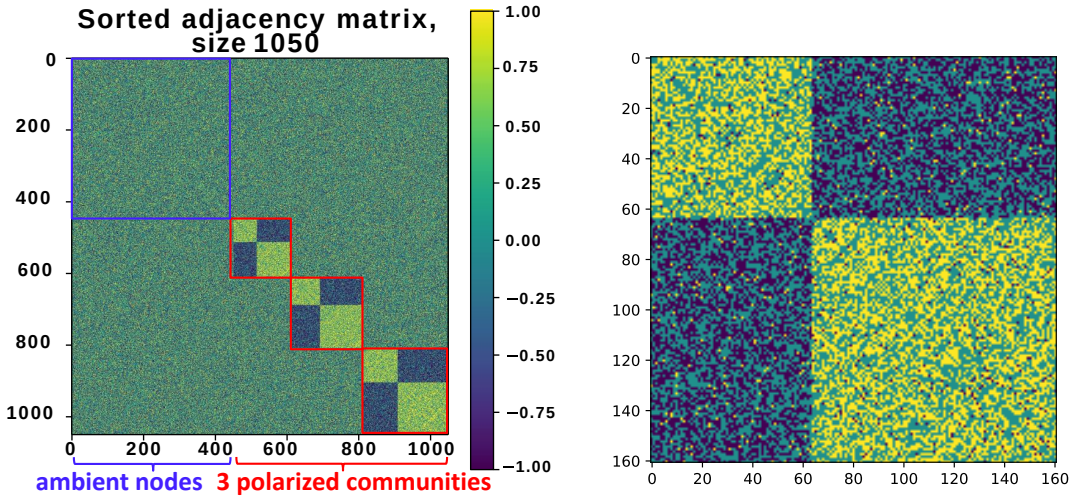
2.4.1 Data

Synthetic Data: Signed Stochastic Block Models (SSBM)

A Signed Stochastic Block Model (SSBM) for a network on n nodes with K blocks (clusters), is constructed similar to [52] but with a more general cluster size definition. In our experiments, we choose the number of clusters, the (approximate) ratio, ρ, between the largest and the smallest cluster size, sign flip probability, η, and the number, n, of nodes. To tackle the hardest clustering task, all pairs of nodes within a cluster and those between clusters have the same edge probability, with more details in SI A.2.1. Our SSBM model can be represented by SSBM(n, K, p, ρ, η).

Synthetic Data: Polarized SSBMs

In a polarized SSBM model, SSBMs are planted in an ambient network; each block of each SSBM is a cluster, and the nodes not assigned to any SSBM form an ambient cluster. The polarized SSBM model that creates communities of SSBMs, is generated as follows: • (1) Generate an Erdős-Rényi graph with n nodes and edge probability p, whose sign is set to ± 1 with equal probability 0.5. • (2) Fix n_c as the number of SSBM communities, and calculate community sizes $N_1 \leq N_2 \leq \cdots \leq N_r$, for each of the r communities as in Section 2.4.1, such that the ratio of the largest block size to the smallest block size is approximately ρ, and the total number of nodes in these SSBMs is $N \times n_c$. • (3) Generate r SSBM models, each with $K_i = 2, i = 1, \ldots, r$ blocks, number of nodes according to its community size, with the same edge probability p, size ratio ρ, and flip probability η. • (4) Place the SSBM models on disjoint subsets of the whole network; the remaining nodes not part of any SSBM are dubbed as *ambient* nodes.



(a) Sorted adjacency matrix. **(b)** Polarized community #1.

Figure 2.3: A polarized SSBM model with 1050 nodes, $r = 3$ polarized communities of sizes 161, 197, and 242; $\rho = 1.5$, default SSBM community size $N = 200, p = 0.5, \eta = 0.05$, and each SSBM has $K_1 = K_2 = K_3 = 2$ blocks, rendering $K = 7$.

Therefore, while the total number of clusters in an SSBM equals the number of blocks, the total number of clusters within a polarized SSBM model equals $K = 1 + \sum_{i=1}^{r} K_i = 1 + 2r$. In our experiments, we also assume the existence of an ambient cluster. The resulting polarized SSBM model is denoted as Pol-SSBM (n, r, p, ρ, η, N). The setting in [69] can be construed as a special case of our model, see SI B.2.

Figure 2.3 gives a visualization of a polarized SSBM model with 1050 nodes, $p = 0.5, \eta = 0.05, N = 200$, with 3 SSBMs of $K = 2$ blocks each, $\rho = 1.5$. The sorted adjacency matrix has its rows and columns sorted by cluster membership, starting with the ambient cluster. With $\rho = 1.5$, the largest SSBM community has size 242, while the smallest has size 161, as $\frac{242}{161} \approx 1.5 = \rho$. For $n = 1050$, the default size of a SSBM community is $N = 200$. For $n = 5000$ (resp, $n = 10000$) we consider $N = 500$ (resp. $N = 2000$).

Real-World Data

We perform experiments on six real-world signed network data sets (*Sampson* [37], *Rainfall* [76], *Fin-YNet*, *S&P 1500* [77], *PPI* [78], and *Wiki-Rfa* [79]), summarized in Table 2.1. *Sampson*, as the only data set with given node attributes (1D

"Cloisterville" binary attribute), cover four social relationships, which are combined into a network; *Rainfall* contains Australian rainfalls pairwise correlations; *Fin-YNet* consists of 21 yearly financial correlation networks and its results are averaged; *S&P1500* is a correlation network of stocks during 2003-2005; *PPI* is a signed protein-protein interaction network; *Wiki-Rfa* describes voting information for electing Wikipedia managers.

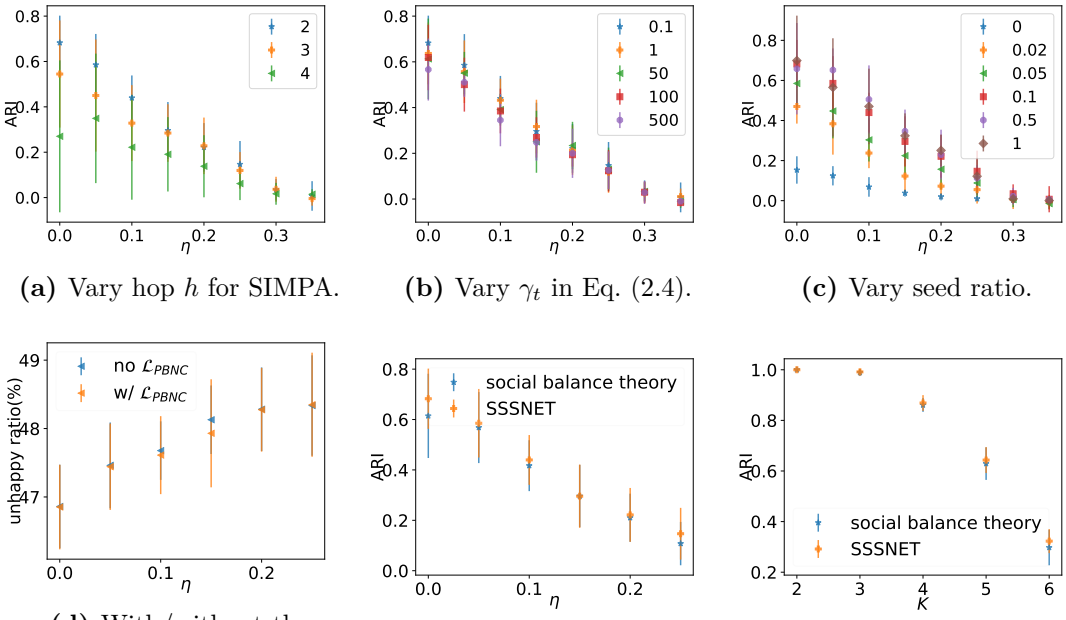
We use labels given by each data set for *Sampson* (5 clusters), and sector memberships for *S&P 1500* and *Fin-YNet* (10 clusters). For *Rainfall*, with 6 clusters, we use labels from SPONGE as proxy for ground truth to carry out semi-supervised learning. For other data sets with no "ground-truth" labels available, we train SSSNET in a self-supervised manner, using all nodes to train. By exploring performance on SPONGE, we set the number of clusters for Wiki-Rfa as three, and similarly ten for PPI. Additional details concerning the data and preprocessing steps are available in SI B.3.

Table 2.1: Summary statistics for the real-world networks and one synthetic model. Here n is the number of nodes, $|\mathcal{E}^+|$ and $|\mathcal{E}^-|$ denote the number of positive and negative edges, respectively. $|\Delta^{\mathrm{u}}|$ counts the number of unbalanced triangles (with an odd number of negative edges). The *violation ratio* $\frac{|\Delta^{\mathrm{u}}|}{|\Delta|}$ (%) is the percentage of unbalanced triangles in all triangles, i.e., 1 minus the Social Balance Factor from [80].

Data set	n	$\lvert\mathcal{E}^+\rvert$	$\lvert\mathcal{E}^-\rvert$	$\lvert\Delta^{\mathrm{u}}\rvert$	$\frac{\lvert\Delta^{\mathrm{u}}\rvert}{\lvert\Delta\rvert}$ (%)
Sampson	25	129	126	192	37.16
Rainfall	306	64,408	29,228	1,350,756	28.29
Fin-YNet	451	14,853	5,431	408,594	26.97
S&P 1500	1,193	1,069,319	353,930	199,839	28.15
PPI	3,058	7,996	3,864	94	2.45
Syn	5,000	510,586	198,6224	9,798,914	47.20
Wiki-Rfa	7,634	136,961	38,826	79,911,143	28.23

2.4.2 Experimental Results

In our experiments, we compare SSSNET against **nine** state-of-the-art spectral clustering methods in signed networks mentioned in Sec. 2.2. These methods are based on: (1) the symmetric adjacency matrix $\mathbf{A}^* = \frac{1}{2}(\mathbf{A} + (\mathbf{A})^T)$, (2) the simple normalized signed Laplacian $\bar{\mathbf{L}}_{sns} = \bar{\mathbf{D}}^{-1}(\mathbf{D}^+ - \mathbf{D}^-\mathbf{A}^*)$ and (3) the balanced normalized signed Laplacian $\bar{\mathbf{L}}_{bns} = \bar{\mathbf{D}}^{-1}(\mathbf{D}^+ - \mathbf{A}^*)$ [63], (4) the Signed Laplacian



(a) Vary hop h for SIMPA. **(b)** Vary γ_t in Eq. (2.4). **(c)** Vary seed ratio.

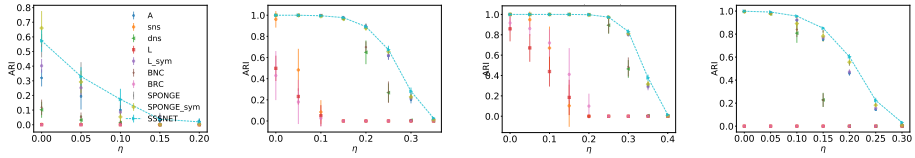
(d) With/without the self-supervised part $\mathcal{L}_{\text{PBNC}}$ in Eq. (2.4). **(e)** Social balance theory vs SSSNET (as we vary η). **(f)** Social balance theory vs SSSNET (as we vary K).

Figure 2.4: Hyperparameter analysis (a,b) and ablation study (c-f). Figures (a-e) pertain to Pol-SSBM($n = 1050, r = 2, p = 0.1, \rho = 1.5$), while Figure (f) is for an SSBM($n = 1000, \eta = 0, p = 0.01, \rho = 1.5$) model with changing K. Figure (d) compares"unhappy ratio" while the others compare the test ARI.

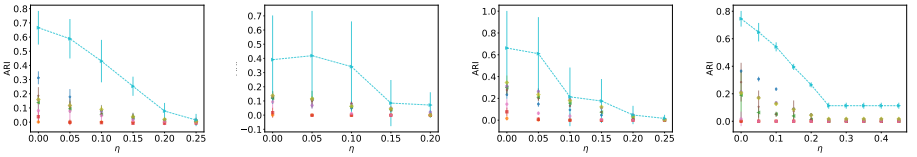
matrix $\bar{\mathbf{L}}$ of $\mathbf{A}^*$, (5) its symmetrically normalized version $\mathbf{L}_{\text{sym}}$ [50], and the two methods from [51] to optimize the (6) Balanced Normalized Cut and the (7) Balanced Ratio Cut, (8) SPONGE and (9) SPONGE_sym introduced in [52], where the diagonal matrices $\bar{\mathbf{D}}, \mathbf{D}^+$ and $\mathbf{D}^-$ have entries as row-sums of $(\mathbf{A}^*)^+ + (\mathbf{A}^*)^-, (\mathbf{A}^*)^+$ and $(\mathbf{A}^*)^-$, respectively. In our experiments, the abbreviated names of these methods are A, sns, dns, L, L_sym, BNC, BRC, SPONGE, and SPONGE_sym, respectively. The implementation details are in SI A.3.

Hyperparameter selection is done via greedy search. Figure 2.4 (a-b) compare the performance of SSSNET on a Pol-SSBM($n = 1050, r = 2, p = 0.1, \rho = 1.5$) model under different settings. We conclude from (a) that as we increase the number of hops to consider, performance drops, which might be explained by too much noise introduced. From (b), we find that the best γ_t in our candidates is 0.1. See also SI A.3.4. Our default setting is $h = 2, d = 32, \tau = 0.5, \gamma_s = 50, \gamma_t = 0.1, \alpha = 0$.

Unless specified otherwise, we use 10% of all nodes from each cluster as test nodes, 10% as validation nodes to select the model, and the remaining 80% as



(a) SSBM($n = 1000, K = 5, p = 0.01, \rho = 1.5$) **(b)** SSBM($n = 5000, K = 5, p = 0.01, \rho = 1.5$) **(c)** SSBM($n = 10000, K = 5, p = 0.01, \rho = 1.5$) **(d)** SSBM($n = 30000, K = 5, p = 0.001, \rho = 1.5$)



(e) Pol-SSBM($n = 1050, r = 2, p = 0.1, \rho = 1$) **(f)** Pol-SSBM($n = 5000, r = 3, p = 0.1, \rho = 1.5$) **(g)** Pol-SSBM($n = 5000, r = 5, p = 0.1, \rho = 1.5$) **(h)** Pol-SSBM($n = 10000, r = 2, p = 0.01, \rho = 1.5$)

Figure 2.5: Node clustering test ARI comparison on synthetic data. Dashed lines highlight SSSNET's performance. Error bars indicate one standard error.

training nodes, 10% of which as seed nodes. We train SSSNET for at most 300 epochs with a 100-epoch early-stopping scheme. As *Sampson* is small, 50% nodes are training nodes, 50% of which are seed nodes, and no validation nodes; we train 80 epochs and test on the remaining 50% nodes. For *S&P 1500*, *Fin-YNet* and *Rainfall*, we use 90% nodes for training, 10% of which as seed nodes, and no validation nodes for 300 epochs. If node attributes are missing, SSSNET stacks the eigenvectors corresponding to the largest K eigenvalues of the symmetrized adjacency matrix $\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$ as $\mathbf{X}_{\mathcal{V}}$ for synthetic data, and the eigenvectors corresponding to the smallest K eigenvalues of the symmetrically normalized Signed Laplacian [81] of $\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$ for real-world data. Numerical results are averaged over 10 runs; error bars indicate one standard error.

Node Clustering Results on Synthetic Data

Figure 2.5 compares the numerical performance of SSSNET with other methods on synthetic data. We remark that SSSNET gives state-of-the-art test ARIs on a wide range of network densities and noise levels, on various network scales, especially for polarized SSBMs. SI A.1.1 provides more results with different measures.

Table 2.2: Clustering performance on real-world data sets; best is in **bold red** , and 2nd in underline blue . The first 3 rows are test ARIs, the 4th "ARI distance to best", the rest are unhappy ratios (%).

Data set	A	sns	dns	L	L_sym	BNC	BRC	SPONGE	SPONGE_sym	SSSNET
Sampson	0.32±0.10	0.15±0.09	0.33±0.10	0.16±0.05	0.35±0.09	0.32±0.12	0.21±0.11	0.36±0.11	0.34±0.11	**0.55±0.07**
Rainfall	0.61±0.08	0.28±0.03	0.65±0.04	0.46±0.06	0.58±0.07	0.62±0.05	0.47±0.05	N/A	0.75±0.09	**0.76±0.13**
S&P 1500	0.21±0.00	0.00±0.00	0.05±0.01	0.06±0.00	0.24±0.00	0.04±0.00	0.00±0.00	0.30±0.00	0.34±0.00	**0.66±0.00**
Fin-YNet	0.22±0.09	0.37±0.12	0.32±0.10	0.33±0.10	0.22±0.09	0.32±0.09	0.33±0.11	0.20±0.08	0.16±0.07	**0.00±0.00**
PPI	57.59±0.55	46.82±0.01	46.79±0.04	46.91±0.03	47.05±0.04	46.63±0.04	52.11±0.42	47.57±0.00	46.39±0.10	**17.64±0.84**
Wiki-Rfa	50.05±0.03	23.28±0.00	23.28±0.00	23.28±0.00	36.95±0.01	23.28±0.00	23.49±0.00	29.63±0.01	**23.26±0.00**	23.27±0.14

Node Clustering Results on Real-World Data

As Table 2.2 shows, SSSNET yields the most accurate cluster assignments on *S&P 1500*, and *Sampson* in terms of test ARI, compared to baselines. When using labels from SPONGE to conduct semi-supervised training, SSSNET also achieves the most accurate test ARI. The N/A entry for SPONGE denotes that we use it as "ground-truth", so do not compare SSSNET against SPONGE on *Rainfall*. "ARI dist. to best" on *Fin-YNet* considers the average distance of test ARI performance to the best average test ARI for each of the 21 years, and then obtains mean and standard deviation over the 21 distances for each method. We conclude that SSSNET produces the highest test ARI for all 21 years. For data sets without labels, we compare SSSNET with other methods in terms of "unhappy ratio", the ratio of unhappy edges. We conclude that SSSNET gives comparable and often better results on these data sets, in a self-supervised setting. SI A.1.2 extends the results.

Ablation Study

In Figure 2.4, (c-e) rely on Pol-SSBM($n = 1050, r = 2, p = 0.1, \rho = 1.5$), while (f) is based on an SSBM($n = 1000, \eta = 0, p = 0.01, \rho = 1.5$) model with varying K. (c) explores the influence of increasing seed ratio, and validates our strength in using labels. (d) assesses the impact of removing the self-supervised loss $\mathcal{L}_{\text{PBNC}}$ from Eq. (2.4). Lower values of the "unhappy ratio" with $\mathcal{L}_{\text{PBNC}}$ reveal that including the self-supervised loss in Eq. (2.4) can be beneficial.

Figure 2.4 (e) compares the performance for $h = 2$ by replacing SIMPA with an aggregation scheme based on social balance theory, which also considers a path of length two with pure negative links as a path of friendship. The ARI for

SSSNET is larger than the one for the corresponding method which would adhere to social balance theory, thus further validating our approach. Figure 2.4 (f) further illustrates the influence of the violation ratio, as percentage (the last column in Table 2.1) on the performance gap between a variant based on social balance theory and our model. As K increases, the violation ratio grows, and we witness a larger gap in test ARI performance, suggesting that social balance theory becomes more problematic when there are more violations to its assumption that "an enemy's enemy is a friend". We conclude that this social balance theory modification not only complicates the calculation by adding one more scenario of friendship, but can also decrease the test ARI. Finally, as we increase the ratio of seed nodes, we witness an increase in test ARI performance, as expected.

2.5 Conclusion and Future Work

SSSNET provides an end-to-end pipeline to create node embeddings and carry out signed clustering, with or without available additional node features, and with an emphasis on polarization. It would be interesting to apply the method to more networks without ground truth, as is often done in community detection, and relate the resulting clusters to exogenous information. As another future direction, instead of specifying the number of clusters, we would like to extend our framework to also detect the number of clusters, see e.g., [82]. Other future research directions will address the performance in the very sparse regime, where spectral methods underperform and various regularization techniques have been proven to be effective both on the theoretical and experimental fronts; for example, see the regularization in the sparse regime for the unsigned [83, 84] and signed clustering settings [85]. Applying signed clustering to cluster multivariate time series, and leveraging the uncovered clusters for the time series prediction task, by fitting the model of choice for each individual cluster, as in [86], is a promising extension. Finally, adapting our pipeline for constrained clustering, a popular task in the semi-supervised learning [87], is worth exploring.

Statement of Authorship for joint/multi-authored papers for PGR thesis

Title of Paper:	*SSSNET: Semi-Supervised Signed Network Clustering*
Publication Status	▪ Published
Publication Details	*He, Yixuan, Gesine Reinert, Songchao Wang, and Mihai Cucuringu. "SSSNET: Semi-Supervised Signed Network Clustering." In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), pp. 244-252. Society for Industrial and Applied Mathematics, 2022.*

Student Confirmation

Student name:	Yixuan He
Contribution to the paper	*This is my work, the topic of which was raised by my supervisor Prof. Mihai Cucuringu. The idea of a new take on social balance theory was proposed by me. This project was conducted under the supervision of my supervisors Prof. Gesine Reinert and Prof. Mihai Cucuringu. I conducted the experiments with the help of my collaborator Songchao Wang and wrote the original manuscript.*
Signature: Yixuan He 何奕萱	**Date: 04/27/2024**

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

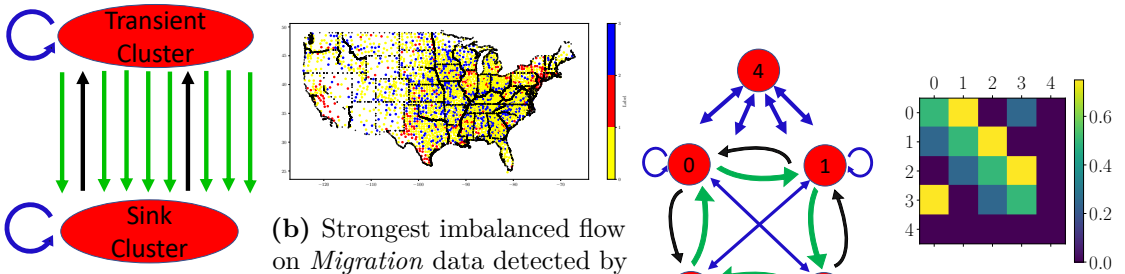
Supervisor name and title: Professor Gesine Reinert	
Supervisor comments	The description is fine.
Signature: Gesine Reinert	**Date:** 29 April 2024

3

DIGRAC: Digraph Clustering Based on Flow Imbalance

3.1 Introduction

Revealing an underlying community structure of *directed* networks (*digraphs*) is an important problem in many applications, see for example [88] and [89], such as detecting influential social groups [4] and analyzing migration patterns [36]. While most existing methods that could be applied to directed clustering use local edge densities as main signal and directionality (i.e, edge orientation) as additional signal, we argue that even in the absence of any edge density differences, directionality can play a vital role in directed clustering as it can reveal latent properties of network flows. The underlying intuition is that homogeneous clusters of nodes form *meta-nodes* in a *meta-graph*, with the meta-graph directing the flow between clusters; directed core-periphery structure is such an example [90]. Loosely speaking, a meta-node is a collection of nodes, and a meta-graph is a graph on such meta-nodes, with weighted edges collecting the overall sum of edge weights between the meta-nodes. Fig. 3.1(a) is an example of flow imbalance between two clusters, here on an unweighted network for simplicity: while 80% of the edges flow from the *Transient* cluster to the *Sink* cluster, only 20% flow in the other direction. As a real-world example, Fig. 3.1(b) shows the strongest flow imbalances between clusters detected



(a) Flow imbalance example: most edges flow from *Transient* to *Sink*.

(b) Strongest imbalanced flow on *Migration* data detected by DIGRAC, along with the geographic locations of the counties and state boundaries (in black).

(c) Meta-graph.

(d) Meta-graph adjacency matrix $\mathbf{F}$.

Figure 3.1: Visualization of cut flow imbalance and meta-graph: (a) 80% of edges flow from Transient to Sink, while 20% of edges flow in the opposite direction; (b) top pair imbalanced flow on *Migration* data [36]: most edges flow from red (1) to blue (2); (c) & (d) are for a Directed Stochastic Block Model with a cycle meta-graph with ambient nodes, for a total of 5 clusters. Most edges flow in direction $0 \to 1 \to 2 \to 3 \to 0$, while few flow in the opposite direction. Cluster 4 is the ambient cluster. In (a) and (c), blue lines indicate flows with random, equally likely directions; these flows do not exist in the meta-graph adjacency matrix $\mathbf{F}$. For (d), the lighter the color, the stronger the flow.

by our method in a network of US migration flow [36]; most edges flow from the red cluster (label 1) to the blue one (label 2). Figures 3.1(c-d) show examples on a synthetic meta-graph. We could also think of a social network in which a set of fake accounts $\mathcal{A}$ have been created, and these target another subset $\mathcal{B}$ of real accounts by sending them messages. Most likely, there would be many more messages from $\mathcal{A}$ to $\mathcal{B}$ than from $\mathcal{B}$ to $\mathcal{A}$, hinting that $\mathcal{A}$ is most likely comprised of fake accounts.

Thus, instead of finding relatively dense groups of nodes in digraphs with a relatively small amount of flow between the groups, as in [91–96], our main goal is to recover clusters with *strongly imbalanced flow* among them, in the spirit of [97, 98], where directionality is the main signal. This task is not addressed by most methods for node clustering in digraphs, including community detection methods. Those methods that do lay emphasis on directionality are usually spectral methods, for which incorporating features is non-trivial, or graph neural network (GNN) methods that require labeling information. An exception is the network community detection method InfoMap [99] which uses directed random walks; however, it still relies on some edge density information within clusters, as a walk is more likely to happen

when the density is higher. [100] and [101] employ Markov chains, but we pick InfoMap as a representative for methods based on information theory/Markov chains.

Here we introduce DIGRAC, a GNN framework to obtain node embeddings for clustering digraphs (allowing weighted edges and self-loops but no multiple edges). In a self-supervised manner, a novel *probabilistic imbalance loss* is proposed to act on the digraph induced by all training nodes. The global imbalance score, one minus whom is the self-supervised loss function, is aggregated from pairwise normalized cut imbalances. The method is end-to-end in combining embedding generation and clustering without an intermediate step. To the best of our knowledge, this is the first GNN method which derives node embeddings for digraphs that directly maximizes flow imbalance between pairs of clusters. With an emphasis on the use of a direction-based flow imbalance objective, experimental results on synthetic data and real-world data at different scales demonstrate that our method can achieve leading performance for a wide range of network densities and topologies.

DIGRAC's main novelty is the ability to cluster based on direction-based flow imbalance, instead of using classical criteria such as maximizing relative densities within clusters. Compared with prior methods that focus on directionality, DIGRAC can easily consider node features and also does not require known node clustering labels. DIGRAC complements existing approaches in various aspects: (1) Our results show that DIGRAC complements classical community detection by detecting alternative patterns in the data, such as meta-graph structures, which are otherwise not detectable by existing methods. This aspect of detecting novel structures in directed graphs has also been emphasized in [97]. (2) DIGRAC complements existing spectral methods, through the possibility of including exogenous information, in the form of node-level features or labels, thus borrowing their strength. (3) DIGRAC complements existing GNN methods by introducing an imbalance-based objective. (4) DIGRAC introduces imbalance measures for evaluation when ground-truth is unavailable.

DIGRAC's applicability extends beyond settings where the input data is a digraph: with time series data as input, the digraph construction mechanism

can accommodate any procedure that encodes a pairwise directional association between the corresponding time series, such as lead-lag relationships and Granger causality [102], with applications such as in the analysis of information flow in brain networks [103], biology [104], finance [105, 106] and earth sciences [107]. DIGRAC could also facilitate tasks in ranking [30] and anomaly detection [108–110], as it allows one to extrapolate from *local* pairwise (directed) interactions to a *global* structure inference, in the high-dimensional low signal-to-noise ratio regime.

Main contributions. Our main contributions are as follows.

1. We propose a GNN framework for self-supervised end-to-end node clustering on (possibly attributed and weighted) digraphs explicitly taking into account the directed flow imbalance.

2. We propose a family of probabilistic global imbalance scores to serve as the self-supervised loss function and evaluation objective, including one based on hypothesis testing for directionality signal. To the best of our knowledge, this is the first method directly maximizing flow imbalance for node clustering in digraphs using GNNs.

3. We extend our method to the semi-supervised setting when label information is available.

3.2 Related Work

Directed clustering has been explored by non-GNN methods. [111] performs directed clustering that hinges on symmetrizations of the adjacency matrix, but is not scalable as it requires large matrix multiplications. [112] proposes a spectral co-clustering algorithm for asymmetry discovery that relies on in-degree and out-degree. Whenever direction is the sole information, such as in a complete network with a lead-lag structure derived from time series [105], a purely degree-based method cannot detect the clusters. While [9] produces two partitions of the node set, one based on out-degree and one based on in-degree, our partition simultaneously takes

both directions into account. The *directed graph Laplacians* introduced by [89] are only applicable to strongly connected digraphs, which is rarely the case in sparse networks arising in applications. InfoMap by [99] assumes that there is a "map" underlying the network, similar to a meta-graph in DIGRAC. InfoMap aims to minimize the expected description length of a random walk and is recommended for networks where edges encode patterns of movement among nodes. While related to DIGRAC, InfoMap still relies on some amount of density-based signal being present within each of the modules. [97] seeks to uncover clusters characterized by a strongly imbalanced flow circulating among them, based on eigenvectors of the Hermitian matrix $(\mathbf{A} - \mathbf{A}^T) \cdot i$, where $\mathbf{A}$ is the (normalized) adjacency matrix and i the imaginary unit. [97] is a purely spectral-based method and is not able to naturally incorporate any available node features or label information; in contrast, DIGRAC is a GNN-based method that is naturally able to account for such information. Moreover, [97] is not driven by an optimization function, but only proposes evaluation metrics that capture the imbalance of the pairs of clusters. In contrast, inspired by [97], in DIGRAC a family of novel imbalance loss functions is proposed, with a probabilistic interpretation, rendering DIGRAC a fully trainable end-to-end pipeline. Furthermore, the rich class of imbalance evaluation and training objectives/losses proposed in this paper go far beyond the evaluation metrics considered in [97]. [98] uncovers higher-order structural information among clusters in digraphs, while maximizing the imbalance of the edge directions, but its definition of the flow ratio restricts the underlying meta-graph to a path.

GNNs have been applied to digraph node classification, which is similar to digraph clustering but requires known clustering labels. [113] uses first and second-order proximity, and constructs three Laplacians, but the method is space and speed-inefficient. [114] simplifies [113], builds a directed Laplacian based on PageRank, and aggregates information dependent on higher-order proximity. Building on [97, 115], [6] constructs a Hermitian matrix that encodes undirected geometric structure in the magnitude of its entries, and directional information in their phase. [116] introduces a digraph data augmentation method called *Laplacian perturbation*

and conducts digraph contrastive learning. [117] proposes a spectral-based graph convolution network for digraphs, yet is restricted to strongly connected digraphs that are usually not realistic. [118] utilizes convolution-like anisotropic filters based on local subgraph structures (motifs) for semi-supervised node classification tasks in digraphs, but relies on pre-defined structures and fails to handle complex networks.

In particular, [6, 113, 114, 116, 118] all require known labels, which are not generally available for real-world data. [89, 97, 98, 111, 112] could not trivially incorporate node attributes or node labels. In contrast, we propose an efficient GNN-based method that maximizes a probablistic flow imbalance objective, in a self-supervised manner, and which can naturally analyze attributed weighted digraphs.

To avoid potential misunderstanding, we briefly mention several related works that we are aware of but do not compare against in our experiments in the main text. While DIGRAC addresses the task of partitioning the nodes into disjoint sets, [119] locates a certain community within a network. In particular, [119] proposes a local algorithm while this paper proposes a global one. OSLOM by [120] is very flexible but based on a density heuristic and hence a comparison to DIGRAC on networks without density signal would not be fair, to begin with. [121] introduces directionality in the Louvain algorithm. This algorithm optimizes a modularity-type function that compares the number of edges within communities to the expected number of edges under a specified model. It is thus an approach that aims to find denser-than-expected groups of vertices. When all groups have the same density, as in our synthetic data sets, and the only structure lies in the directionality of the edges, this method simply cannot be expected to perform well. The Leiden algorithm in [10] also builds on the Louvain method, again optimizing a modularity-type function that compares the number of edges within communities to the expected number of edges under a specified model. It is a powerful method for that task, but cannot be fairly compared to DIGRAC which is tailored to find imbalances. As confirmed by our experiments in Appendix (App.) B.5, comparing these methods to DIGRAC is not appropriate.

We also do not compare DIGRAC against graph pooling methods [122], which are inspired by pooling in CNNs and developed to discard information that is superfluous for the task at hand, as a partition of the nodes which can be interpreted as clustering is only a byproduct. Moreover, graph pooling methods are usually developed only for undirected networks. While graph matching as in [123–125] and [126] can be viewed as a clustering method of networks, matching the graph of interest to a disconnected graph by connecting each node in the observed graph with an isolated node of the disconnected graph, this approach is not developed for directed networks. The underlying idea of these papers is complementary to the meta-graph idea which underpins DIGRAC; in the meta-graph, the components are connected, and estimating the directionality of these connections is the main focus. Hence this work addresses a very different task. We emphasize that these are all excellent methods, but they address different objectives and tasks. DIGRAC is tailored to detect an imbalance signal in directed networks, and such a signal cannot be present in an undirected network. As it is based on imbalance, DIGRAC will not be able to detect a signal in an undirected network, thus rendering it not applicable to undirected networks.

3.3 The DIGRAC Framework

Problem definition. Denote a (possibly weighted) digraph with node attributes as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w, \mathbf{X})$, with $\mathcal{V}$ the set of nodes, $\mathcal{E}$ the set of directed edges or links, and $w \in [0, \infty)^{|\mathcal{E}|}$ the set of edge weights. $\mathcal{G}$ may have self-loops, but no multiple edges. The number of nodes is $n = |\mathcal{V}|$, and $\mathbf{X} \in \mathbb{R}^{n \times d_{\text{in}}}$ is a matrix whose rows encode the nodes' attributes. Such a network can be represented by the attribute matrix $\mathbf{X}$ and the adjacency matrix $\mathbf{A} = (A_{ij})_{i,j \in \mathcal{V}}$, with $\mathbf{A}_{ij} = 0$ if no edge exists from v_i to v_j; if there is an edge e from v_i to v_j, we set $A_{ij} = w_e$, the edge weight.

Digraphs often lend themselves to interpreting weighted directed edges as flows, with a meta-graph on clusters of vertices describing the overall flow directions; see Fig. 3.1. A clustering is a partition of the set of nodes into K disjoint sets (clusters) $\mathcal{V} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{K-1}$ (ideally, $K \geq 2$). Intuitively, nodes within

a cluster should be similar to each other with respect to flow directions, while nodes across clusters should be dissimilar. In a self-supervised setting, only the number of clusters K is given. In a semi-supervised setting, for each of the K clusters, a fraction set $\mathcal{V}^{\text{seed}} \subseteq \mathcal{V}^{\text{train}} \subset \mathcal{V}$ of the set $\mathcal{V}^{\text{train}}$ of all training nodes is selected to serve as the set of seed nodes, for which the cluster membership labels are known before training. The goal of semi-supervised clustering is to assign each node $v \in \mathcal{V}$ to a cluster containing some known seed nodes, without knowledge of the underlying flow meta-graph. The corresponding self-supervised clustering task does not use seed nodes.

3.3.1 Self-Supervised Loss for Clustering

Our self-supervised loss function is inspired by [97], aiming to cluster the nodes by maximizing a normalized form of cut imbalance across clusters. We first define probabilistic versions of cuts, imbalance flows, and probabilistic volumes. For K clusters, the *assignment probability matrix* $\mathbf{P} \in \mathbb{R}^{n \times K}$ has as row i the probability vector $\mathbf{P}_{(\mathbf{i},:)} \in \mathbb{R}^K$ with entries denoting the probabilities of each node to belong to each cluster; its k^{th} column is denoted by $\mathbf{P}_{(:,k)}$.

• $\forall k, l \in \{0, \ldots, K-1\}$ where $K \geq 2$, the **probabilistic cut** from cluster $\mathcal{C}_k$ to $\mathcal{C}_l$ is defined as

$$W(\mathcal{C}_k, \mathcal{C}_l) = \sum_{i,j} \mathbf{A}_{i,j} \cdot \mathbf{P}_{i,k} \cdot \mathbf{P}_{j,l} = (\mathbf{P}_{(:,k)})^T \mathbf{A} \mathbf{P}_{(:,l)}.$$

• The **imbalance flow** between $\mathcal{C}_k$ and $\mathcal{C}_l$ is defined as $|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|$.

For interpretability and ease of comparison, we normalize the imbalance flows to obtain an imbalance score with values in $[0, 1]$ as follows (we defer additional details to App. B.2.2).

• The **probabilistic volume** for cluster $\mathcal{C}_k$ is defined as

$$\begin{aligned} VOL(\mathcal{C}_k) &= VOL^{(\text{out})}(\mathcal{C}_k) + VOL^{(\text{in})}(\mathcal{C}_k) \\ &= \sum_{i,j} (\mathbf{A}_{j,i} + \mathbf{A}_{i,j}) \cdot \mathbf{P}_{j,k} \end{aligned}$$

Then $VOL(\mathcal{C}_k) \geq W(\mathcal{C}_k, \mathcal{C}_l)$ for all $l = 1, \ldots, K-1$ and

$$\min(VOL(\mathcal{C}_k), VOL(\mathcal{C}_l)) \geq |W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|. \tag{3.1}$$

The imbalance term, which is used in most of our experiments, denoted $\mathrm{CI}^{\mathrm{vol\_sum}}$, is defined as

$$\mathrm{CI}^{\mathrm{vol\_sum}}(k,l) = 2\frac{|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|}{VOL(\mathcal{C}_k) + VOL(\mathcal{C}_l)} \in [0,1]. \tag{3.2}$$

In particular, for $K = n$, every node is a single cluster, and $\mathrm{CI}^{\mathrm{vol\_sum}}(k,l)$ =1, but then the partition is not informative. The aim is to find a partition that maximizes the imbalance flow under the constraint that the partition has at least two sets, to capture groups of nodes that could be viewed as representing clusters in the meta-graph. The normalization by the volumes penalizes partitions that put most nodes into a single cluster. The range $[0,1]$ follows from Eq. (3.1). Other variants are discussed in App. B.2.3.

To obtain a **global probabilistic imbalance score**, based on $\mathrm{CI}^{\mathrm{vol\_sum}}$ from Eq. (3.2), we average over pairwise imbalance scores of different pairs of clusters. Since the scores discussed are symmetric and the cut difference before taking absolute value is skew-symmetric, we only need to consider the pairs in the set $\mathcal{T} = \{(\mathcal{C}_k, \mathcal{C}_l) : 0 \leq k < l \leq K-1, k, l \in \mathbb{Z}\}$.

A naive approach, which we call the **"*naive*"** variant, considers all possible $\binom{K}{2}$ pairwise cut imbalance values. However, due to potentially high noise levels in certain data sets, one may only be interested in pairs that are not just noise but exhibit true signals. To this end, we introduce a **"*std*"** variant, which only considers pairwise cut imbalance values that are 3 standard deviations away from the observed purely noisy imbalance values; the standard deviation is calculated under the null hypothesis that the between-cluster relationship has no direction preference, i.e. $\mathbf{F}_{k,l} = \mathbf{F}_{l,k}$ (entries of the meta-graph adjacency matrix $\mathbf{F}$ to be introduced later in this section), as follows.

Suppose two clusters $\mathcal{C}_k$ and $\mathcal{C}_l$ have only noisy links between them, with no edge in the meta-graph $\mathbf{F}$, i.e. $\mathbf{F}_{kl} = 0$. Assume also that the underlying

network is fixed in terms of the number of nodes and locations of edges; the only randomness stems from the direction of the edges. Then we can provide the following theoretical guarantee.

Proposition 1. *Suppose that $\mathcal{C}_k$ and $\mathcal{C}_l$ are two clusters of n_k and n_l nodes, respectively, with $m(k,l)$ edges between them, edge weights $w_{ij} = w_{ji} \in [0,1]$ and edge direction drawn independently at random with equal probability $\frac{1}{2}$ for each direction. We assume that the edge weights satisfy $\max_e |w_e|(\sum_e w_e^2)^{-\frac{1}{2}} = o(m(k,l))$. Then $W(\mathcal{C}_k,\mathcal{C}_l) - W(\mathcal{C}_l,\mathcal{C}_k)$ is approximately normally distributed with mean 0 and variance $||w||^2$ as $m(k,l) \to \infty$.*

A consequence of Proposition 1, which is proved in App. B.2.1, is that under its assumptions, approximately 99.7 % of the observations fall within 3 standard deviations from 0. While Proposition 1 makes many assumptions and ignores reciprocal edges, the resulting threshold is still a useful guideline for restricting attention to pairwise imbalance values which are very likely to capture a true signal. In particular, we use it as motivation for our "std" variant to pick cluster pairs from $\mathcal{T}$ that satisfy $\left(W(\mathcal{C}_k,\mathcal{C}_l) - W(\mathcal{C}_l,\mathcal{C}_k)\right)^2 > 9\left(W(\mathcal{C}_k,\mathcal{C}_l) + W(\mathcal{C}_l,\mathcal{C}_k)\right)$.

As we are mainly concerned about the top pairs (i.e., those exhibiting the largest imbalance flow), another option is the **"*sort*"** variant, which selects the largest β pairwise cut imbalance values, where β is half of the number of nonzero entries in the off-diagonal entries of the meta-graph adjacency matrix $\mathbf{F}$, if the meta-graph is known or can be approximated. For example, for a "cycle" meta-graph with three clusters and no ambient nodes, $\beta = 3$. When the meta-graph is a "path" with three clusters and ambient nodes, then $\beta = 1$. When considering the "sort" variant, with $\mathcal{T}(\beta) = \{(\mathcal{C}_k,\mathcal{C}_l) \in \mathcal{T} : \mathrm{CI}^{\mathrm{vol\_sum}}(k,l)$ is among the top β values$\}$, where $1 \le \beta \le \binom{K}{2}$, we set

$$\mathcal{O}^{\mathrm{sort}}_{\mathrm{vol\_sum}} = \frac{1}{\beta} \sum_{(\mathcal{C}_k,\mathcal{C}_l)\in\mathcal{T}(\beta)} \mathrm{CI}^{\mathrm{vol\_sum}}(k,l), \quad \text{and} \quad \mathcal{L}^{\mathrm{sort}}_{\mathrm{vol\_sum}} = 1 - \mathcal{O}^{\mathrm{sort}}_{\mathrm{vol\_sum}}, \tag{3.3}$$

as the corresponding loss function. Definitions of meta-graph structures are discussed in Section 3.4.1. For the other variants, the corresponding scores and loss

functions are defined analogously. We apply the “std" variant when we have no prior knowledge of the meta-graph structure during training, and the “sort" variant when we have information on the number of pairs to count.

When using the “std" variant for training, for the initial 50 epochs, we apply the “sort" variant with $\beta = 3$ for a reasonable starting clustering probability matrix for training, as otherwise during the initial training epochs possibly no pairs could be picked out. During the epochs actually utilizing this “std" variant, if no pairs could be picked out, we temporarily switch to the “naive" variant for that epoch.

Regarding complexity, the objective mainly contains matrix-vector multiplications and element-wise matrix divisions, which are at most quadratic in the number of nodes, but usually faster with our sparsity-aware implementation.

3.3.2 Instantiation of DIGRAC

To instantiate DIGRAC, any aggregation scheme able to take directionality into account could be incorporated into our general framework, as long as it can output the node embedding matrix $\mathbf{Z}$. Here, by default, we adapt the Signed Mixed Path Aggregation (SIMPA) scheme from [3]. We remove the signed parts and devise a simple yet effective directed mixed path aggregation scheme, which we call Directed Mixed Path Aggregation (DIMPA), to obtain the probability assignment matrix $\mathbf{P}$

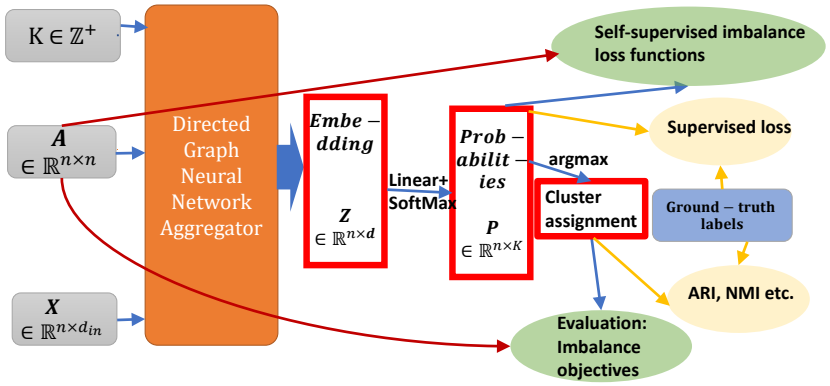


Figure 3.2: DIGRAC overview: from feature matrix $\mathbf{X}$, adjacency matrix $\mathbf{A}$ and number of clusters K, we first apply a directed GNN aggregator to obtain the node embedding matrix $\mathbf{Z}$, then apply a linear layer followed by a unit softmax function to get the probability matrix $\mathbf{P}$. Applying *argmax* on each row of $\mathbf{P}$ yields node cluster assignments. Green circles involve our proposed imbalance objective, while the yellow circles can only be used when ground-truth labels are provided.

by applying a linear layer followed by a unit softmax function to the embedding generated, and feed it to the loss function. Details of DIMPA are provided in App. B.1. A framework diagram is provided in Fig. 3.2, and an instantiation using DIMPA is visualized in Fig. B.1.

3.4 Experiments

In our synthetic experiments, when by design ground truth is available, performance is assessed by the Adjusted Rand Index (ARI) [23]. Normalized Mutual Information (NMI) results give almost the same ranking for the best-performing methods as the ARI, with an average Kendall tau value of 83.8% and standard deviation 24.9%, for pairwise ranking comparison, on the methods compared in our experiments. We do not focus on NMI in the main text due to its shortcomings [127], see also App. B.3.9.

Clustering tasks will have different ground truths, depending on the pattern they are trying to detect. Many network clustering methods focus on detecting relatively dense clusters, and try to optimize classical network clustering measures, such as directed modularity or partition density. Ground truth for these clustering algorithms then relates to relatively densely connected subgroups in the data. DIGRAC is a novel method that addresses a novel task, namely that of detecting flow imbalances. To the best of our knowledge, real-world data sets with ground-truth flow imbalances are not available to date, and hence we introduce normalized imbalance scores to evaluate clustering performance based on flow imbalance. As ARI and NMI require ground-truth labels, they thus cannot be applied to the available real-world data sets. To address this shortcoming, for the real-world data sets, in Table 3.1, we include three performance measures which we introduce in the paper, and the appendix contains an additional 11 performance measures. Implementation details are provided in App. B.3. Codes and preprocessed data are available at `https://github.com/SherylHYX/DIGRAC_Directed_Clustering` and have been included in the open-source library *PyTorch Geometric Signed Directed* [28].

We compare DIGRAC against the most recent related methods from the literature for clustering digraphs. The **10** methods are • (1) InfoMap [99], •

(2) Bibliometric and • (3) Degree-discounted introduced in [111], • (4) DI_SIM [112], • (5) Herm and • (6) Herm_sym introduced in [97], • (7) MagNet [6], • (8) DGCN [113], • (9) DiGCN [114], and • (10) DiGCL [116]. The abbreviations of these methods, when reported in the numerical experiments, are InfoMap, Bi_sym, DD_sym, DISG_LR, Herm, Herm_sym, MagNet, DGCN, DiGCN, DiGCL, respectively. DGCN is the least efficient method in terms of speed and space complexity, followed by DiGCN which involves the so-called *inception blocks*. We use the same hyperparameter settings stated in these papers. Methods (7), (8), (9), (10) are GNN methods which are trained with 80% nodes under label supervision, while all the other methods are trained without label supervision. DIGRAC further restricts itself to be trained on the subgraph induced by only the training nodes. All methods are designed for directed graphs, and all except Infomap require K to be known. Runtime comparison is provided in App. B.3.2, illustrating that DIGRAC is among the fastest among competing GNNs. Implementation details for competitors are provided in App. B.3.7.

3.4.1 Data Sets

Synthetic data: Directed Stochastic Block Models A standard directed stochastic block model (DSBM) is often used to represent a network cluster structure, see for example [88]. Its parameters are the number K of clusters and the edge probabilities; given the cluster assignment of the nodes, the edge indicators are independent. The DSBMs used in our experiments also depend on a meta-graph adjacency matrix $\mathbf{F} = (\mathbf{F}_{k,l})_{k,l=0,\dots,K-1}$ and a *filled* version of it, $\tilde{\mathbf{F}} = (\tilde{\mathbf{F}}_{k,l})_{k,l=0,\dots,K-1}$, and on a noise level parameter $\eta \leq 0.5$. The meta-graph adjacency matrix $\mathbf{F}$ is generated from the given meta-graph structure, called $\mathcal{M}$. To include an ambient background, the filled meta-graph adjacency matrix $\tilde{\mathbf{F}}$ replaces every zero in $\mathbf{F}$ that is not part of the imbalance structure by 0.5. The filled meta-graph thus creates a number of *ambient nodes* which correspond to entries which are not part of $\mathcal{M}$ and thus are not part of a meaningful cluster; this set of *ambient nodes* is also called the *ambient cluster*. First, we provide examples of structures of $\mathbf{F}$ without any ambient

nodes, where $\mathbb{1}$ denotes the indicator function.

• (1) "*cycle*": $\mathbf{F}_{k,l} = (1-\eta)\mathbb{1}(l = ((k+1) \mod K)) + \eta\mathbb{1}(l = ((k-1) \mod K)) + \frac{1}{2}\mathbb{1}(l = k)$. • (2) "*path*": $\mathbf{F}_{k,l} = (1-\eta)\mathbb{1}(l = k+1) + \eta\mathbb{1}(l = k-1) + \frac{1}{2}\mathbb{1}(l = k)$.

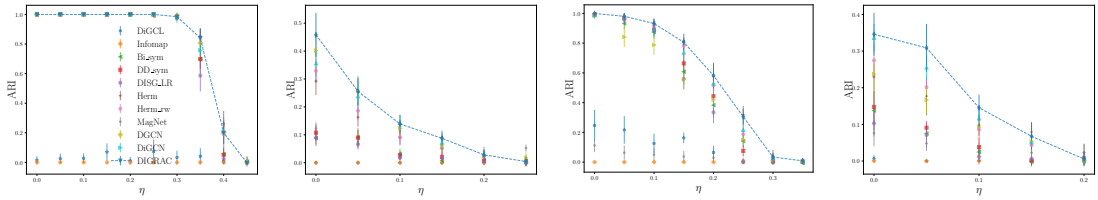
• (3) "*complete*": assign diagonal entries $\frac{1}{2}$. For each pair (k, l) with $k < l$, let $\mathbf{F}_{k,l}$ be η and $1-\eta$ with equal probability, then assign $\mathbf{F}_{l,k} = 1 - \mathbf{F}_{k,l}$.

• (4) "*star*", following [128]: select the center node as $\omega = \lfloor \frac{K-1}{2} \rfloor$ and set $\mathbf{F}_{k,l} = (1-\eta)\mathbb{1}(k = \omega, l \text{ odd}) + \eta\mathbb{1}(k = \omega, l \text{ even}) + (1-\eta)\mathbb{1}(l = \omega, k \text{ odd}) + \eta\mathbb{1}(l = \omega, l \text{ even})$.

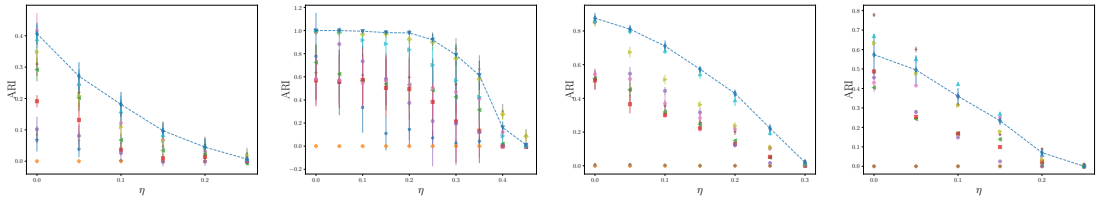
When ambient nodes are present, the construction involves two steps, with the first step the same as the above, but with the following changes: For "*cycle*" meta-graph structure, $\mathbf{F}_{k,l} = (1-\eta)\mathbb{1}(l = ((k+1) \mod (K-1))) + \eta\mathbb{1}(l = ((k-1) \mod (K-1))) + 0.5\,\mathbb{1}(l = k)$. The second step is to assign 0 (0.5, resp.) to the last row and the last column of $\mathbf{F}$ ($\tilde{\mathbf{F}}$, resp.). Figures 3.1(c-d) display a "*cycle*" meta-graph structure with ambient nodes (in cluster 4). The majority of edges flow in the form $0 \to 1 \to 2 \to 3 \to 0$, while few flow from the opposite direction. Fig. 3.1(d) illustrates the meta-graph adjacency matrix corresponding to this $\mathbf{F}$.

In our experiments, we choose the number of clusters, the (approximate) ratio, ρ, between the largest and the smallest cluster size, and the number, n, of nodes. To tackle the hardest clustering task and also focus on directionality, all pairs of nodes within a cluster and all pairs of nodes between clusters have the same edge probability, p. Note that for $\mathcal{M}$ ="*cycle*", even the expected in-degree and out-degree of all nodes are identical. Our DSBM, which we denote by DSBM $(\mathcal{M}, \mathbb{1}(\text{ambient}), n, K, p, \rho, \eta)$, is built similarly to [97] but with possibly unequal cluster sizes, with more details in App. B.3.3. For each node $v_i \in \mathcal{C}_k$, and each node $v_j \in \mathcal{C}_l$, independently sample an edge from node v_i to node v_j with probability $p \cdot \tilde{\mathbf{F}}_{k,l}$. The parameter settings in our experiments are $p \in \{0.001, 0.01, 0.02, 0.1\}$, $\rho \in \{1, 1.5\}$, $K \in \{3, 5, 10\}$, $\mathbb{1}(\text{ambient}) \in \{\text{T, F}\}$ (True and False), $n \in \{1000, 5000, 30000\}$, and we also vary the direction flip probability η from 0 to 0.45, with a 0.05 step size.

Real-world data We perform experiments on five real-world digraph data sets with sizes ranging from 245 to over 2 million nodes: *Telegram* [4], *Blog* [129], *Migration* [36], *WikiTalk* [38], and *Lead-Lag* [105], with details in App. B.3.3. We



(a) DSBM("cycle", F, $n = 1000, K = 3, p = 0.1, \rho = 1$) **(b)** DSBM("complete", T, $n = 1000, K = 10, p = 0.02, \rho = 1$) **(c)** DSBM("cycle", F, $n = 1000, K = 3, p = 0.02, \rho = 1.5$) **(d)** DSBM("path", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)



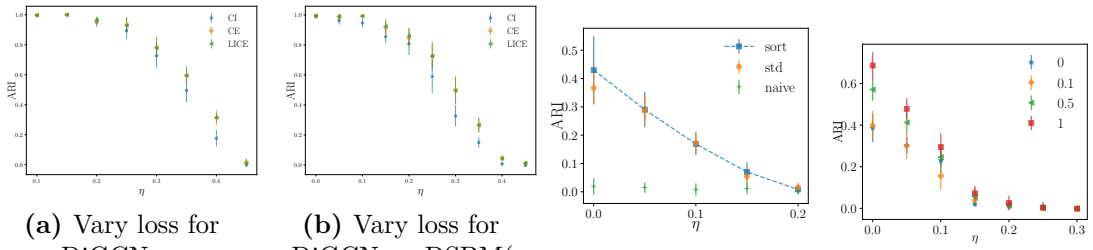
(e) DSBM("star", F, $n = 1000, K = 5, p = 0.02, \rho = 1$) **(f)** DSBM("complete", F, $n = 1000, K = 3, p = 0.1, \rho = 1$) **(g)** DSBM("cycle", T, $n = 5000, K = 5, p = 0.01, \rho = 1.5$) **(h)** DSBM("cycle", F, $n = 30000, K = 5, p = 0.001, \rho = 1$)

Figure 3.3: Test ARI comparison on synthetic data. Dashed lines highlight DIGRAC's performance. Error bars are given by one standard error.

Table 3.1: Performance comparison on real-world data sets. The best is marked in **bold red** and the second best is marked in underline blue . The objectives are defined in Section 3.3.1.

Metric	Data set	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{\text{sort}}_{\text{vol\_sum}}$	*Telegram*	0.04±0.00	0.21±0.0	0.21±0.0	0.21±0.01	0.2±0.01	0.14±0.0	**0.32±0.01**
	Blog	0.07±0.00	0.07±0.0	0.0±0.0	0.05±0.0	0.37±0.0	0.0±0.0	**0.44±0.0**
	Migration	N/A	0.03±0.00	0.01±0.00	0.02±0.00	0.04±0.00	0.02±0.00	**0.05±0.00**
	WikiTalk	N/A	N/A	N/A	0.18±0.03	0.15±0.02	0.0±0.0	**0.24±0.05**
	Lead-Lag	N/A	0.07±0.01	0.07±0.01	0.07±0.01	0.07±0.02	0.07±0.02	**0.15±0.03**
$\mathcal{O}^{\text{std}}_{\text{vol\_sum}}$	*Telegram*	0.01±0.00	0.26±0.00	0.26±0.00	0.26±0.01	0.25±0.02	**0.35±0.00**	0.28±0.01
	Blog	0.00±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
	Migration	N/A	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.02±0.00	**0.04±0.01**
	WikiTalk	N/A	N/A	N/A	**0.17±0.04**	0.06±0.01	0.01±0.00	0.14±0.02
	Lead-Lag	N/A	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	**0.12±0.03**
$\mathcal{O}^{\text{naive}}_{\text{vol\_sum}}$	*Telegram*	0.01±0.00	0.26±0.0	0.26±0.0	0.26±0.01	0.25±0.02	0.23±0.0	**0.27±0.01**
	Blog	0.00±0.00	0.07±0.0	0.0±0.0	0.05±0.0	0.37±0.0	0.0±0.0	**0.44±0.0**
	Migration	N/A	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.01±0.00	**0.04±0.01**
	WikiTalk	N/A	N/A	N/A	0.1±0.02	0.04±0.0	0.0±0.0	**0.12±0.01**
	Lead-Lag	N/A	0.30±0.06	0.28±0.06	0.27±0.06	0.29±0.05	0.29±0.05	**0.32±0.11**

set the number of clusters K to be 4, 2, 10, 10, 10, respectively, and values of β to be 5, 1, 9, 10, 3, respectively. Note that *Lead-Lag* comprises of 19 separate networks constructed from yearly financial time series, rendering a total of 23 real-world networks.



(a) Vary loss for DiGCN on DSBM("complete", F, $n = 1000, K = 3, p = 0.1, \rho = 1$).

(b) Vary loss for DiGCN on DSBM("complete", T, $n = 1000, K = 5, p = 0.1, \rho = 1.5$).

(c) Vary selection methods of imbalance pairs.

(d) Vary seed ratio in a semi-supervised setting.

Figure 3.4: Ablation study. (c-d) are on DSBM("cycle", F, $n = 1000, K = 5, p = 0.02, \rho = 1$).

3.4.2 Experimental results

Training Set-Up

As training setup, we use 10% of all nodes from each cluster as test nodes, 10% as validation nodes to select the model, and the remaining 80% as training nodes. In each setting, unless otherwise stated, we carry out 10 experiments with different data splits. Error bars are given by one standard error. When no node attributes are given, the matrix $\mathbf{X}$ for DIGRAC is taken as the stacked eigenvectors corresponding to the largest K eigenvalues of the random-walk symmetrized Hermitian matrix used in the comparison method Herm_rw. The imbalance loss function acts on the subgraph induced by the training nodes. To further clarify the training setup, DIGRAC uses 0% of the labels in training. As DIGRAC is a self-supervised method, in principle, we could use all nodes for training. However for a fair comparison with other GNN methods we use only 80% of the nodes for training. For supervised methods our split of 80% - 10% - 10% is a standard split. For the non-GNN methods, all nodes are used for training. The default loss function for DIGRAC is $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$.

Results on Synthetic Data

Fig. 3.3 compares the numerical performance of DIGRAC with other methods on synthetic data. For this Fig. we generate 5 DSBM networks under each parameter setting and use 10 different data splits for each network, then average

over the 50 runs. Error bars are given by one standard error. App. B.3 provides additional implementation details.

We conclude that DIGRAC compares favorably against state-of-the-art methods, on a wide range of network densities and noise levels, on different network sizes, and with different underlying meta-graph structures, with and without ambient nodes. Being a self-supervised method, DIGRAC even attains comparable or better performance than fully-supervised GNN competitors.

Results on Real-World Data

For our real-world data sets, the node in- and out-degrees may not be identical across clusters. Moreover, as these data sets do not contain node attributes, DIGRAC considers the eigenvectors corresponding to the largest K eigenvalues of the Hermitian matrix from [97] to construct an input feature matrix. Table 3.1 reveals that DIGRAC provides competitive global imbalance scores in three objectives discussed and across all real-world data sets, and outperforms all other methods in 13 out of 15 instances, while attains the second-best performance for the remaining two instances. The N/A entries for *WikiTalk* are caused by memory error, and the N/A entries for InfoMap on *Migration* and *Lead-Lag* are due to its prediction of only one single cluster. For *Migration*, as detailed in Fig. 3.1(b) and App. B.4.4, DIGRAC is able to uncover nontrivial migration patterns, such as migration from California to Arizona, as discovered by [36]. *Lead-Lag* results in each year are averaged over ten runs, while the mean and standard deviation values are calculated with respect to the 19 years. The experiments indicate that edge directionality contains an important signal that DIGRAC is able to capture. As App. B.4.2 illustrates, DIGRAC is able to provide comparable or higher pairwise imbalance scores for the leading pairs. The fitted meta-graph plots in App. B.4.3 reveal that DIGRAC is able to recover a directed flow imbalance between clusters in all of the selected data sets. A comprehensive numerical comparison in App. B.4 reveals similar conclusions.

3.4.3 Ablation Study

Figures 3.4(a-b) compare the performance of DiGCN replacing the loss function by $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$ from Eq. (3.3), indicated by "CI" (self-supervised loss only), or "LICE" (sum of supervised and self-supervised loss), on two synthetic models. We find that replacing the supervised loss function with $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$ leads to comparable results, and that adding $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$ to the loss could be beneficial, indicating that the imbalance objectives are more general than only applicable to DIMPA. Fig. 3.4(c) compares the test ARI performance using three variants of loss functions on the same digraph. The current choice "*sort*" performs best among these variants, indicating a benefit in only considering top pairs of individual imbalance scores. The "std" variant is comparable with the "sort" variant, but the "sort" variant performs the best with prior knowledge on the network structure. More details on loss functions, comparison with other variants, and evaluation on additional metrics are discussed in App. B.2, with similar conclusions. As illustrated in Fig. 3.4(d), again on the same digraph, we also experiment on adding seeds, with the seed ratio defined as the ratio of the number of seed nodes to the number of training nodes. A supervised loss, following [3], is then applied to these seeds; App. B.3.5 contains additional details. In conclusion, seed nodes with a supervised loss function enhance performance, and we infer that our model can further boost its performance when additional label information is available.

3.5 Conclusion, Limitations and Outlook

DIGRAC provides an end-to-end pipeline to create node embeddings and perform directed clustering, with or without available additional node features or cluster labels. We illustrate DIGRAC on publicly available data without any personally identifiable information. DIGRAC could potentially have societal impact, for example, in detecting clusters of fake accounts in social networks. While we do not envision our work to have any negative societal impact, vigilance is of course required.

Current limitations that could be addressed by future work include detecting the number of clusters [82, 95], instead of specifying it a-priori, as this is typically not available in real-world applications. The relatively small sizes of the networks used in the paper (the largest has 2 million nodes) also opens future direction in adapting our pipeline to extremely large networks, possibly combined with sampling methods or mini-batch [61], rendering DIGRAC applicable to large scale industrial applications. We also intent to further explore the effect of normalization terms in our objectives, and to design more powerful objectives that could explicitly account for varying edge density.

Another future direction pertains to additional experiments in the semi-supervised setting, when there exist seed nodes with known cluster labels, or when additional information is available in the form of *must-link* and *cannot-link* constraints, popular in the *constrained clustering* literature [130, 131]. Further research directions will also address the performance in the sparse regime, where spectral methods are known to underperform, and various regularizations have been proven to be effective theoretically and empirically; e.g., see regularization in the sparse regime for the undirected settings [83–85].

Statement of Authorship for joint/multi-authored papers for PGR thesis

Title of Paper:	*DIGRAC: Digraph Clustering Based on Flow Imbalance*
Publication Status	▪ Published
Publication Details	*He, Yixuan, Gesine Reinert, and Mihai Cucuringu. "DIGRAC: Digraph Clustering Based on Flow Imbalance." In Learning on Graphs Conference, pp. 21-1. PMLR, 2022.*

Student Confirmation

Student name:	Yixuan He
Contribution to the paper	*This is my work, the topic of which was raised by my supervisor Prof. Mihai Cucuringu. The idea of the loss functions and their normalizations came from discussions between my supervisors (Prof. Gesine Reinert and Prof. Mihai Cucuringu) and me. The theoretical results were from my supervisor Prof. Gesine Reinert and me. This project was conducted under the supervision of my supervisors Prof. Gesine Reinert and Prof. Mihai Cucuringu. I conducted the experiments and wrote the original manuscript.*
Signature: Yixuan He 何奕萱	**Date: 04/27/2024**

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Gesine Reinert	
Supervisor comments	The description is fine.
Signature: Gesine Reinert	**Date:** 29 April 2024

4

MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian

4.1 Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for extracting information from graph-structured data and have achieved state-of-the-art performance on a variety of machine learning tasks. However, compared to research on constructing GNNs for unsigned and undirected graphs, and graphs with multiple types of edges, GNNs for graphs where the edges have a natural notion of sign, direction, or both, have received relatively little attention.

There is a demand for such tools because many important and interesting phenomena are naturally modeled as signed and/or directed graphs, i.e., graphs in which objects may have either positive or negative relationships, and/or in which such relationships are not necessarily symmetric [28]. For example, in the analysis of social networks, positive and negative edges could model friendship or enmity, and directional information could model the influence of one person on another [132, 133]. Signed/directed networks also arise when analyzing time-series data with lead-lag relationships [105], detecting influential groups in social networks [5], and

computing rankings from pairwise comparisons [30]. Additionally, signed and directed networks are a natural model for group conflict analysis [134], modeling the interaction network of the agents during a rumor spreading process [135], and maximizing positive influence while formulating opinions [136].

In general, most GNNs are either spectral or spatial. Spatial methods typically define convolution on graphs as a localized aggregation whereas spectral methods rely on the eigen-decomposition of a suitable graph Laplacian. Our goal is to introduce a novel Laplacian and an associated GNN for signed directed graphs. While several spatial GNNs exist, such as SDGNN [133], SiGAT [137], SNEA [54], and SSSNET [3] for signed (and possibly directed) networks, this is one of the first works to propose a spectral GNN for such networks. We devote special attention to the concurrent preprint SigMaNet [138] which also constructs a spectral GNN based on a different Laplacian.

A principal challenge in extending traditional spectral GNNs to this setting is to define a proper notion of the signed, directed graph Laplacian. Such a Laplacian should be positive semidefinite, have a bounded spectrum when properly normalized, and encode information about both the sign and direction of each edge. Here, we unify the magnetic Laplacian, which has been used in [6] to construct a GNN on an (unsigned) directed graph, with a signed Laplacian which has been used for a variety of data science tasks on (undirected) signed graphs [50, 62, 63, 85]. Importantly, our proposed matrix, which we refer to as the *magnetic signed Laplacian*, reduces to either the magnetic Laplacian or the signed Laplacian when the graph is directed, but not signed, or signed, but not directed.

Although this magnetic signed Laplacian is fairly straightforward to obtain, it is novel and surprisingly powerful: We show that our proposed ***M****agnetic* ***S****igned* ***GNN*** *(MSGNN)* is effective for a variety of node clustering and link prediction tasks. Specifically, we consider several variations of the link prediction task, some of which prioritize signed information over directional information, some of which prioritize directional information over signed information, while others emphasize the method's ability to extract both signed and directional information simultaneously.

In addition to testing MSGNN on established data sets, we also devise a novel synthetic model which we call the *Signed Directed Stochastic Block Model (SDSBM)*, which generalizes both the (undirected) Signed Stochastic Block Model from [3] and the (unsigned) Directed Stochastic Block Model from [5]. Analogous to these previous models, our SDSBM can be defined by a meta-graph structure and additional parameters describing density and noise levels. We also introduce a number of signed directed networks for link prediction tasks using lead-lag relationships in real-world financial time series.

Main Contributions. The main contributions of our work are:

1. We devise a novel matrix called the magnetic signed Laplacian, which can naturally be applied to signed and directed networks. The magnetic signed Laplacian is Hermitian, positive semidefinite, and the eigenvalues of its normalized counterpart lie in $[0, 2]$. Our proposed Laplacian matrix and its counterpart reduce to existing Laplacians when the network is unsigned and/or undirected.

2. We propose an efficient spectral graph neural network architecture, MSGNN, based on this magnetic signed Laplacian, which attains leading performance on extensive node clustering and link prediction tasks, including novel tasks that consider edge sign and directionality jointly. To the best of our knowledge, this is the first work to evaluate GNNs on tasks that are related to both edge sign and directionality.[1]

3. We introduce a novel synthetic model for signed and directed networks, called Signed Directed Stochastic Block Model (SDSBM), and also contribute a number of new real-world data sets constructed from lead-lag relationships of financial time series data.

[1]Some previous work, such as [133], evaluates GNNs on signed and directed graphs. However, they focus on tasks where either only signed information is important, or where only directional information is important.

4.2 Related Work

In this section, we review related work constructing neural networks for directed graphs and signed graphs. We refer the reader to [28] for more background information.

Several works have aimed to define neural networks on directed graphs by constructing various directed graph Laplacians and defining convolution as multiplication in the associated eigenbasis. [117] defines a directed graph Laplacian by generalizing identities involving the undirected graph Laplacian and the stationary distribution of a random walk. [113] uses a similar idea, but with PageRank in place of a random walk. [114] constructs three different first- and second-order symmetric adjacency matrices and uses these adjacency matrices to define associated Laplacians. Similarly, [118] uses several different graph Laplacians based on various graph motifs.

Quite closely related to our work, [6] constructs a graph neural network using the magnetic Laplacian. Indeed, in the case where all edge weights are positive, our GNN exactly reduces to the one proposed in [6]. Importantly, unlike the other directed graph Laplacians mentioned above, the magnetic Laplacian is a complex, Hermitian matrix rather than a real, symmetric matrix. We also note [5], which constructs a GNN for node clustering on directed graphs based on flow imbalance.

All of the above works are restricted to unsigned graphs, i.e., graphs with positive edge weights. However, there are also a number of neural networks introduced for signed (and possibly also directed) graphs, mostly focusing on the task of link sign prediction, i.e., predicting whether a link between two nodes will be positive or negative. SGCN by [48] is one of the first graph neural network methods to be applicable to signed networks, using an approach based on balance theory [56]. However, its design is mainly aimed at undirected graphs. SiGAT [137] utilizes a graph attention mechanism based on [139] to learn node embeddings for signed, directed graphs, using a novel motif-based GNN architecture based on balance theory and status theory [38]. Subsequently, SDGNN by [133] builds upon this work by increasing its efficiency and proposing a new objective function. In a similar vein, SNEA [54] proposes a signed graph neural network for link sign prediction based on

a novel objective function. In a different line of work, [3] proposes SSSNET, a GNN not based on balance theory designed for semi-supervised node clustering in signed (and possibly directed) graphs. A concurrent preprint, SigMaNet [138], proposes a signed magnetic Laplacian to construct a spectral GNN.[2] Additionally, several GNNs [141–143] have been introduced for multi-relational graphs, i.e., graphs with different types of edges. In such networks, the number of learnable parameters typically increases linearly with the number of edge types. Signed graphs, at least if the graph is unweighted or the weighting function w only takes finitely many values, can be thought of as special cases of multi-relational graphs. However, in the context of (possibly weighted) signed graphs, there is an implicit relationship between the different edge types, namely that a negative edge is interpreted as the opposite of a positive edge and that edges with large weights are deemed more important than edges with small weights. These relationships will allow us to construct a network with significantly fewer trainable parameters than if we were considering an arbitrary multi-relational graph.

4.3 Proposed Method

4.3.1 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w, \mathbf{X}_{\mathcal{V}})$ denote a signed, and possibly directed, weighted graph with node attributes, where $\mathcal{V}$ is the set of nodes (or vertices), $\mathcal{E}$ is the set of (directed) edges (or links), and $w : \mathcal{E} \to (-\infty, \infty) \setminus \{0\}$ is the weighting function. Let $\mathcal{E}^+ = \{e \in \mathcal{E} : w(e) > 0\}$ denote the set of positive edges and let $\mathcal{E}^- = \{e \in \mathcal{E} : w(e) < 0\}$ denote the set of negative edges so that $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$. Here, we do allow self-loops but not multiple edges; if $v_i, v_j \in \mathcal{V}$, there is at most one edge $e \in \mathcal{E}$ from v_i to v_j. Let $n = |\mathcal{V}|$, and let d_{in} be the number of attributes at each node, so that $\mathbf{X}_{\mathcal{V}}$ is an $n \times d_{\text{in}}$ matrix whose rows are the attributes of each node. We let $\mathbf{A} = (A_{ij})_{i,j \in \mathcal{V}}$ denote the weighted, signed adjacency matrix where $\mathbf{A}_{i,j} = w_{i,j}$ if $(v_i, v_j) \in \mathcal{E}$, and $\mathbf{A}_{i,j} = 0$ otherwise.

[2] After the initial submission of this work we became aware of a concurrent work [140] by another research group which developed a network similar to ours independently at the same time.

4.3.2 Magnetic Signed Laplacian

In this section, we define Hermitian matrices $\mathbf{L}_U^{(q)}$ and $\mathbf{L}_N^{(q)}$ which we refer to as the unnormalized and normalized magnetic signed Laplacian matrices, respectively. We first define a symmetrized adjacency matrix and an absolute degree matrix by

$$\tilde{\mathbf{A}}_{i,j} \coloneqq \frac{1}{2}(\mathbf{A}_{i,j} + \mathbf{A}_{j,i}), \;\; 1 \leq i,j \leq n, \;\; \tilde{\mathbf{D}}_{i,i} \coloneqq \frac{1}{2}\sum_{j=1}^{n}(|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|), \;\; 1 \leq i \leq n,$$

with $\tilde{\mathbf{D}}_{i,j} = 0$ for $i \neq j$. Importantly, the use of absolute values ensures that the entries of $\tilde{\mathbf{D}}$ are non-negative. Furthermore, it ensures that all $\tilde{\mathbf{D}}_{i,i}$ will be strictly positive if the graph is connected. This is in contrast to the construction in [138] which will give a node degree zero if it has an equal number of positive and negative neighbors (for unweighted networks). To capture directional information, we next define a phase matrix $\mathbf{\Theta}^{(q)}$ by $\mathbf{\Theta}_{i,j}^{(q)} \coloneqq 2\pi q(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})$, where $q \in \mathbb{R}$ is the so-called "charge parameter." In our experiments, for simplicity, we set $q = 0$ when the task at hand is unrelated to directionality, or when the underlying graph is undirected, and we set $q = q_0 \coloneqq 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$ (so that $\Theta^{(q)}$ has entries $\in [-\pi, \pi]$) for all the other tasks (except in an ablation study on the role of q). With $\odot$ denoting elementwise multiplication, and $\mathbb{i}$ denoting the imaginary unit, we now construct a complex Hermitian matrix $\mathbf{H}^{(q)}$ by

$$\mathbf{H}^{(q)} \coloneqq \tilde{\mathbf{A}} \odot \exp(\mathbb{i}\mathbf{\Theta}^{(q)})$$

where $\exp(\mathbb{i}\mathbf{\Theta}^{(q)})$ is defined elementwise by $\exp(\mathbb{i}\mathbf{\Theta}^{(q)})_{i,j} \coloneqq \exp(\mathbb{i}\mathbf{\Theta}_{i,j}^{(q)})$.

Note that $\mathbf{H}^{(q)}$ is Hermitian, as $\tilde{\mathbf{A}}$ is symmetric and $\mathbf{\Theta}^{(q)}$ is skew-symmetric. In particular, when $q = 0$, we have $\mathbf{H}^{(0)} = \tilde{\mathbf{A}}$. Therefore, setting $q = 0$ is equivalent to making the input graph symmetric and discarding directional information. In general, however, $\mathbf{H}^{(q)}$ captures information about a link's sign, through $\tilde{\mathbf{A}}$, and about its direction, through $\mathbf{\Theta}^{(q)}$.

We observe that flipping the direction of an edge, i.e., replacing a positive or negative link from v_i to v_j with a link of the same sign from v_j to v_i corresponds to complex conjugation of $\mathbf{H}_{i,j}^{(q)}$ (assuming either that there is not already a link

from v_j to v_i or that we also flip the direction of that link if there is one). We also note that if $q = 0.25$, $\mathbf{A}_{i,j} = \pm 1$, and $\mathbf{A}_{j,i} = 0$, we have

$$\mathbf{H}^{(0.25)}_{i,j} = \pm\frac{i}{2} = -\mathbf{H}^{(0.25)}_{j,i}.$$

Thus, a unit-weight edge from v_i to v_j is treated as the opposite of a unit-weight edge from v_j to v_i.

Given $\mathbf{H}^{(q)}$, we next define the unnormalized magnetic signed Laplacian by

$$\mathbf{L}^{(q)}_U \coloneqq \tilde{\mathbf{D}} - \mathbf{H}^{(q)} = \tilde{\mathbf{D}} - \tilde{\mathbf{A}} \odot \exp(\mathbb{i}\boldsymbol{\Theta}^{(q)}), \tag{4.1}$$

and also define the normalized magnetic signed Laplacian by

$$\mathbf{L}^{(q)}_N \coloneqq \mathbf{I} - \left(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\right) \odot \exp(\mathbb{i}\boldsymbol{\Theta}^{(q)}). \tag{4.2}$$

When the graph $\mathcal{G}$ is directed, but not signed, $\mathbf{L}^{(q)}_U$ and $\mathbf{L}^{(q)}_N$ reduce to the magnetic Laplacians utilized in works such as [6, 144, 145] and [146]. Similarly, when $\mathcal{G}$ is signed, but not directed, $\mathbf{L}^{(q)}_U$ and $\mathbf{L}^{(q)}_N$ reduce to the signed Laplacian matrices considered in e.g., [50, 85] and [147]. Additionally, when the graph is neither signed nor directed, they reduce to the standard normalized and unnormalized graph Laplacians [148]. The following theorems show that $\mathbf{L}^{(q)}_U$ and $\mathbf{L}^{(q)}_N$ satisfy properties analogous to the traditional graph Laplacians. The proofs are in Appendix C.1.

Theorem 1. *For any signed directed graph $\mathcal{G}$ defined in Sec. 4.3.1, $\forall q \in \mathbb{R}$, both the unnormalized magnetic signed Laplacian $\mathbf{L}^{(q)}_U$ and its normalized counterpart $\mathbf{L}^{(q)}_N$ are positive semidefinite.*

Theorem 2. *For any signed directed graph $\mathcal{G}$ defined in Sec. 4.3.1, $\forall q \in \mathbb{R}$, the eigenvalues of the normalized magnetic signed Laplacian $\mathbf{L}^{(q)}_N$ are contained in the interval $[0, 2]$.*

By construction, $\mathbf{L}^{(q)}_U$ and $\mathbf{L}^{(q)}_N$ are Hermitian, and Theorem 1 shows they are positive semidefinite. In particular, they are diagonalizable by an orthonormal basis of complex eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ associated to real, nonnegative eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n = \lambda_{\max}$. Thus, similar to the traditional normalized Laplacian, we

may factor $\mathbf{L}_N^{(q)} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\dagger$, where $\mathbf{U}$ is an $n \times n$ matrix whose k-th column is $\mathbf{u}_k$, for $1 \leq k \leq n$, $\mathbf{\Lambda}$ is a diagonal matrix with $\mathbf{\Lambda}_{k,k} = \lambda_k$, and $\mathbf{U}^\dagger$ is the conjugate transpose of $\mathbf{U}$. A similar formula holds for $\mathbf{L}_U^{(q)}$.

We conclude this subsection with a comparison to SigMaNet, proposed in the concurrent preprint [138]. SigMaNet also constructs a GNN based on a signed magnetic Laplacian, which is different from the magnetic signed Laplacian proposed here. The claimed advantage of SigMaNet is that it does not require the tuning of a charge parameter q and is invariant to, e.g., doubling the weight of every edge. In our work, for the sake of simplicity, we usually set $q = 0.25$, except for when the graph is undirected (in which case we set $q = 0$). However, a user may choose to also tune q through a standard cross-validation procedure as in [6]. Moreover, one can readily address the latter issue by normalizing the adjacency matrix via a preprocessing step (e.g., [149]). In contrast to our magnetic signed Laplacian, in the case where the graph is not signed but is weighted and directed, the matrix proposed in [138] does not reduce to the magnetic Laplacian considered in [6]. For example, denoting the graph adjacency matrix by $\mathbf{A}$, consider the case where $0 < \mathbf{A}_{j,i} < \mathbf{A}_{i,j}$. Let $m = \frac{1}{2}(\mathbf{A}_{i,j} + \mathbf{A}_{j,i})$, $\delta = \mathbf{A}_{i,j} - \mathbf{A}_{j,i}$, and let $\mathbb{i}$ denote the imaginary unit. Then the (i,j)-th entry of the matrix $\mathbf{L}^\sigma$ proposed in [138] is given by $\mathbf{L}^\sigma_{i,j} = m\mathbb{i}$, whereas the corresponding entry of the unnormalized magnetic Laplacian is given by $(\mathbf{L}_U^{(q)})_{i,j} = m\exp(2\pi\mathbb{i}q\delta)$. Moreover, while SigMaNet is in principle well-defined on signed and directed graphs, the experiments in [138] are restricted to tasks where only signed or directional information is important (but not both). In our experiments, we find that our proposed method outperforms SigMaNet on a variety of tasks on signed and/or directed networks. Moreover, we observe that the signed magnetic Laplacian $\mathbf{L}^\sigma$ proposed in [138] has an undesirable property when the graph is unweighted — a node is assigned to have degree zero if it has an equal number of positive and negative connections. Our proposed Laplacian does not suffer from this issue.

4.3.3 Spectral Convolution via the Magnetic Signed Laplacian

In this section, we show how to use a Hermitian, positive semidefinite matrix $\mathbf{L}$ such as the normalized or unnormalized magnetic signed Laplacian introduced in Sec. 4.3.2, to define convolution on a signed directed graph. This method is similar to the ones proposed for unsigned (possibly directed) graphs in, e.g., [71, 148, 150] and [6], but we provide details in order to keep our work reasonably self-contained.

Given $\mathbf{L}$, let $\mathbf{u}_1 \dots, \mathbf{u}_n$ be an orthonormal basis of eigenvectors such that $\mathbf{L}\mathbf{u}_k = \lambda_k \mathbf{u}_k$, and let $\mathbf{U}$ be an $n \times n$ matrix whose k-th column is $\mathbf{u}_k$, for $1 \leq k \leq n$. For a signal $\mathbf{x} : \mathcal{V} \to \mathbb{C}$, we define its Fourier transform $\widehat{\mathbf{x}} \in \mathbb{C}^n$ by $\widehat{\mathbf{x}}(k) = \langle \mathbf{x}, \mathbf{u}_k \rangle \coloneqq \mathbf{u}_k^\dagger \mathbf{x}$, and equivalently, $\widehat{\mathbf{x}} = \mathbf{U}^\dagger \mathbf{x}$. Since $\mathbf{U}$ is unitary, we readily obtain the Fourier inversion formula

$$\mathbf{x} = \mathbf{U}\widehat{\mathbf{x}} = \sum_{k=1}^{n} \widehat{\mathbf{x}}(k)\mathbf{u}_k \,. \tag{4.3}$$

Analogous to the well-known convolution theorem in Euclidean domains, we define the convolution of $\mathbf{x}$ with a filter $\mathbf{y}$ as multiplication in the Fourier domain, i.e., $\widehat{\mathbf{y} * \mathbf{x}}(k) = \widehat{\mathbf{y}}(k)\widehat{\mathbf{x}}(k)$. By equation 4.3, this implies $\mathbf{y} * \mathbf{x} = \mathbf{U}\mathrm{Diag}(\widehat{\mathbf{y}})\widehat{\mathbf{x}} = (\mathbf{U}\mathrm{Diag}(\widehat{\mathbf{y}})\mathbf{U}^\dagger)\mathbf{x}$, where $\mathrm{Diag}(\mathbf{z})$ denotes a diagonal matrix with the vector $\mathbf{z}$ on its diagonal. Therefore, we say that $\mathbf{Y}$ is a *generalized convolution matrix* if

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\dagger \,, \tag{4.4}$$

for a diagonal matrix $\mathbf{\Sigma}$. This is a natural generalization of the class of convolutions used in [151].

A main purpose of using a graph filter as in equation 4.4 is to reduce the number of parameters while maintaining permutation invariance. Some potential drawbacks exist when defining a convolution via equation 4.4. First, it requires one to compute the eigen-decomposition of $\mathbf{L}$ which is expensive for large graphs. Second, the number of trainable parameters equals the size of the graph (the number of nodes), rendering GNNs constructed via equation 4.4 prone to overfitting. To remedy these issues, we follow [148] (see also [150]) and observe that spectral convolution may

also be implemented in the spatial domain via polynomials of $\mathbf{L}$ by setting $\mathbf{\Sigma}$ equal to a polynomial of $\mathbf{\Lambda}$. This reduces the number of trainable parameters from the size of the graph to the degree of the polynomial and also enhances robustness to perturbations [152]. As in [148], we let $\widetilde{\mathbf{\Lambda}} = \frac{2}{\lambda_{\max}}\mathbf{\Lambda} - \mathbf{I}$ denote the normalized eigenvalue matrix (with entries in $[-1, 1]$) and choose $\mathbf{\Sigma} = \sum_{k=0}^{K} \theta_k T_k(\widetilde{\mathbf{\Lambda}})$, for some $\theta_1, \ldots, \theta_k \in \mathbb{R}$ where for $0 \leq k \leq K$, T_k is the Chebyshev polynomials defined by $T_0(x) = 1, T_1(x) = x$, and $T_k(x) = 2xT_{k-1}(x) + T_{k-2}(x)$ for $k \geq 2$. Since $\mathbf{U}$ is unitary, we have $(\mathbf{U}\widetilde{\mathbf{\Lambda}}\mathbf{U}^\dagger)^k = \mathbf{U}\widetilde{\mathbf{\Lambda}}^k\mathbf{U}^\dagger$, and thus, letting $\widetilde{\mathbf{L}} \coloneqq \frac{2}{\lambda_{\max}}\mathbf{L} - \mathbf{I}$, we have

$$\mathbf{Yx} = \mathbf{U}\sum_{k=0}^{K} \theta_k T_k(\widetilde{\mathbf{\Lambda}})\mathbf{U}^\dagger\mathbf{x} = \sum_{k=0}^{K} \theta_k T_k(\widetilde{\mathbf{L}})\mathbf{x}\,. \tag{4.5}$$

This is the class of convolutional filters we will use in our experiments. However, one could also imitate Sec. 3.1 on [6] to produce a class of filters based on [71] rather than [148].

It is important to note that $\widetilde{\mathbf{L}}$ is constructed so that, in equation 4.5, $(\mathbf{Yx})_i$ depends on all nodes within K-hops from v_i on the undirected, unsigned counterpart of $\mathcal{G}$, i.e. the graph whose adjacency matrix is given by $\mathbf{A}'_{i,j} = \frac{1}{2}(|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|)$. Therefore, this notion of convolution does not favor "outgoing neighbors" $\{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$ over "incoming neighbors" $\{v_j \in V : (v_j, v_i) \in \mathcal{E}\}$ (or vice versa). This is important since for a given node v_i, both sets may contain different, useful information. Furthermore, since the phase matrix $\mathbf{\Theta}^{(q)}$ encodes an outgoing edge and an incoming edge differently, the filter matrix $\mathbf{Y}$ is also able to aggregate information from these two sets in different ways. Regarding computational complexity, we note that while the matrix $\exp(\mathbb{i}\mathbf{\Theta}^{(q)})$ is dense in theory, in practice, one only needs to compute a small fraction of its entries corresponding to the nonzero entries of $\tilde{\mathbf{A}}$ (which is sparse for most real-world data sets). Thus, the computational complexity of the convolution proposed here is equivalent to that of its undirected, unsigned counterparts.

4.3.4 The MSGNN architecture

We now define our network, MSGNN. Let $\mathbf{X}^{(0)}$ be an $n \times F_0$ input matrix with columns $\mathbf{x}_1^{(0)}, \ldots \mathbf{x}_{F_0}^{(0)}$, and L denote the number of convolution layers. As in [6], we use a complex version of the Rectified Linear Unit defined by $\sigma(z) = z$, if $-\pi/2 \leq \arg(z) < \pi/2$, and $\sigma(z) = 0$ otherwise, where $\arg(\cdot)$ is the complex argument of $z \in \mathbb{C}$. Let F_ℓ be the number of channels in the ℓ-th layer. For $1 \leq \ell \leq L$, $1 \leq i \leq F_\ell$, and $1 \leq j \leq F_{\ell-1}$, let $\mathbf{Y}_{ij}^{(\ell)}$ be a convolution matrix defined by equation 4.4 or equation 4.5. Given the $(\ell-1)$-st layer hidden representation matrix $\mathbf{X}^{(\ell-1)}$, we define $\mathbf{X}^{(\ell)}$ columnwise by

$$\mathbf{x}_j^{(\ell)} = \sigma\left(\sum_{i=1}^{F_{\ell-1}} \mathbf{Y}_{ij}^{(\ell)} \mathbf{x}_i^{(\ell-1)} + \mathbf{b}_j^{(\ell)}\right), \tag{4.6}$$

where $\mathbf{b}_j^{(\ell)}$ is a bias vector with equal real and imaginary parts, $\text{Real}(\mathbf{b}_j^{(\ell)}) = \text{Imag}(\mathbf{b}_j^{(\ell)})$. In matrix form we write $\mathbf{X}^{(\ell)} = \mathbf{Z}^{(\ell)}\left(\mathbf{X}^{(\ell-1)}\right)$, where $\mathbf{Z}^{(\ell)}$ is a hidden layer of the form equation 4.6. In our experiments, we utilize convolutions of the form equation 4.5 with $\mathbf{L} = \mathbf{L}_N^{(q)}$ and set $K = 1$, in which case we obtain

$$\mathbf{X}^{(\ell)} = \sigma\left(\mathbf{X}^{(\ell-1)}\mathbf{W}_{\text{self}}^{(\ell)} + \widetilde{\mathbf{L}}_N^{(q)}\mathbf{X}^{(\ell-1)}\mathbf{W}_{\text{neigh}}^{(\ell)} + \mathbf{B}^{(\ell)}\right),$$

where $\mathbf{W}_{\text{self}}^{(\ell)}$ and $\mathbf{W}_{\text{neigh}}^{(\ell)}$ are learned weight matrices corresponding to the filter weights of different channels and $\mathbf{B}^{(\ell)} = (\mathbf{b}_1^{(\ell)}, \ldots, \mathbf{b}_{F_\ell}^{(\ell)})$. After the convolutional layers, we unwind the complex matrix $\mathbf{X}^{(L)}$ into a real-valued $n \times 2F_L$ matrix. For node clustering, we then apply a fully connected layer followed by the softmax function. By default, we set $L = 2$, in which case, our network is given by

$$\text{softmax}(\text{unwind}(\mathbf{Z}^{(2)}(\mathbf{Z}^{(1)}(\mathbf{X}^{(0)})))\mathbf{W}^{(3)}).$$

For link prediction, we apply the same method, except we concatenate rows corresponding to pairs of nodes after the unwind layer before applying the linear layer and softmax.

4.4 Experiments

4.4.1 Tasks and Evaluation Metrics

Node Clustering In the node clustering task, one aims to partition the nodes of the graph into the disjoint union of C sets $\mathcal{C}_0, \ldots, \mathcal{C}_{C-1}$. Typically in an unsigned, undirected network, one aims to choose the $\mathcal{C}_i$'s so that there are many links within each cluster and comparably few links between clusters, in which case nodes within each cluster are *similar* due to dense connections. In general, however, similarity could be defined differently [153]. In a signed graph, clusters can be formed by grouping together nodes with positive links and separating nodes with negative links (see [3]). In a directed graph, clusters can be determined by a directed flow on the network (see [5]). More generally, we can define clusters based on an underlying meta-graph, where meta-nodes, each of which corresponds to a cluster in the network, can be distinguished based on either signed or directional information (e.g., flow imbalance [5]). This general meta-graph idea motivates our introduction of a novel synthetic network model, which we will define in Sec. 4.4.2, driven by both link sign and directionality. All of our node clustering experiments are done in the semi-supervised setting, where one selects a fraction of the nodes in each cluster as seed nodes, with known cluster membership labels. In all of our node clustering tasks, we measure our performance using the Adjusted Rand Index (ARI) [23].

Link Prediction On undirected, unsigned graphs, link prediction is simply the task of predicting whether or not there is a link between a pair of nodes. Here, we consider five different variations of the link prediction task for *signed and/or directed* networks. In our first task, link sign prediction (SP), one assumes that there is a link from v_i to v_j and aims to predict whether that link is positive or negative, i.e., whether $(v_i, v_j) \in \mathcal{E}^+$ or $(v_i, v_j) \in \mathcal{E}^-$. Our second task, direction prediction (DP), one aims to predict whether $(v_i, v_j) \in \mathcal{E}$ or $(v_j, v_i) \in \mathcal{E}$ under the assumption that exactly one of these two conditions holds. We also consider three-, four-, and five-class prediction problems. In the three-class problem (3C), the possibilities are $(v_i, v_j) \in \mathcal{E}$, $(v_j, v_i) \in \mathcal{E}$, or that neither (v_i, v_j) nor (v_j, v_i) are in

$\mathcal{E}$. For the four-class problem (4C), the possibilities are $(v_i, v_j) \in \mathcal{E}^+$, $(v_i, v_j) \in \mathcal{E}^-$, $(v_j, v_i) \in \mathcal{E}^+$, and $(v_j, v_i) \in \mathcal{E}^-$. For the five-class problem (5C), we also add in the possibility that neither (v_i, v_j) nor (v_j, v_i) are in $\mathcal{E}$. For all tasks, we evaluate the performance with classification accuracy. Notably, while (SP), (DP), and (3C) only require a method to be able to extract signed *or* directed information, the tasks (4C) and (5C) require it to be able to effectively process both sign *and* directional information. Also, we discard those edges that satisfy more than one condition in the possibilities for training and evaluation, but these edges are kept in the input network which is observed during training.

4.4.2 Synthetic Data for Node Clustering

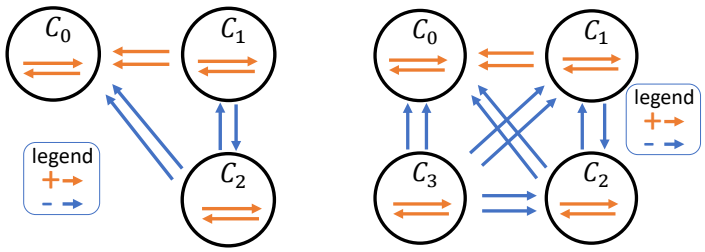
Established Synthetic Models We conduct experiments on the Signed Stochastic Block Models (SSBMs) and polarized SSBMs (POL-SSBMs) introduced in [3], which are signed but undirected. In the SSBM(n, C, p, ρ, η) model, n represents the number of nodes, C is the number of clusters, p is the probability that there is a link (of either sign) between two nodes, ρ is the approximate ratio between the largest cluster size and the smallest cluster size, and η is the probability that an edge will have the "wrong" sign, i.e., that an intra-cluster edge will be negative or an inter-cluster edge will be positive. POL-SSBM (n, r, p, ρ, η, N) is a hierarchical variation of the SSBM model consisting of r communities, each of which is itself an SSBM. We refer the reader to [3] for details of both models.

A novel Synthetic Model: Signed Directed Stochastic Block Model (SDSBM) Given a meta-graph adjacency matrix $\mathbf{F} = (\mathbf{F}_{k,l})_{k,l=0,\dots,C-1}$, an edge sparsity level p, a number of nodes n, and a sign flip noise level parameter $0 \leq \eta \leq 0.5$, we defined a SDSBM model, denoted by SDSBM $(\mathbf{F}, n, p, \rho, \eta)$, as follows: (1) Assign block sizes $n_0 \leq n_1 \leq \cdots \leq n_{C-1}$ based on a parameter $\rho \geq 1$, which approximately represents the ratio between the size of largest block and the size of the smallest block, using the same method as in [3]. (2) Assign each node to one of the C blocks, so that each block C_i has size n_i. (3) For nodes $v_i \in \mathcal{C}_k$, and

$v_j \in \mathcal{C}_l$, independently sample an edge from v_i to v_j with probability $p \cdot |\mathbf{F}_{k,l}|$. Give this edge weight 1 if $F_{k,l} \geq 0$ and weight -1 if $F_{k,l} < 0$. (4) Flip the sign of all the edges in the generated graph with sign-flip probability η.

In our experiments, we use two sets of specific meta-graph structures $\{\mathbf{F}_1(\gamma)\}, \{\mathbf{F}_2(\gamma)\}$, with three and four clusters, respectively, where $0 \leq \gamma \leq 0.5$ is the directional noise level. Specifically, we are interested in SDSBM $(\mathbf{F}_1(\gamma), n, p, \rho, \eta)$ and SDSBM $(\mathbf{F}_2(\gamma), n, p, \rho, \eta)$ models with varying γ where

$$\mathbf{F}_1(\gamma) = \begin{bmatrix} 0.5 & \gamma & -\gamma \\ 1-\gamma & 0.5 & -0.5 \\ -1+\gamma & -0.5 & 0.5 \end{bmatrix}, \mathbf{F}_2(\gamma) = \begin{bmatrix} 0.5 & \gamma & -\gamma & -\gamma \\ 1-\gamma & 0.5 & -0.5 & -\gamma \\ -1+\gamma & -0.5 & 0.5 & -\gamma \\ -1+\gamma & -1+\gamma & -1+\gamma & 0.5 \end{bmatrix}.$$



(a) Three-cluster model $\mathbf{F}_1$. **(b)** Four-cluster model $\mathbf{F}_2$.

Figure 4.1: SDSBM illustration.

To better understand the above SDSBM models, toy examples are provided. We consider the following toy examples for our proposed synthetic data in Figure 4.1, which models groups of athletes and sports fans on social media. Here, signed, directed edges represent positive or negative mentions. In Figure 4.1(a), $\mathcal{C}_0$ are the players of a sports team, $\mathcal{C}_1$ is a group of their fans who typically say positive things about the players, and $\mathcal{C}_2$ is a group of fans of a rival team, who typically say negative things about the players. Since they are fans of rival teams, the members of $\mathcal{C}_1$ and $\mathcal{C}_2$ both say negative things about each other. In general, fans mention the players more than players mention fans, which leads to net flow imbalance. In Figure 4.1(b), we add in $\mathcal{C}_3$, a group of fans of a third, less important team. This group dislikes the other two teams and disseminates negative content about $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$. However, since this third team is quite unimportant, no one comments anything back.

Notably, in both examples, as the expected edge density is identical both within and across clusters, discarding either signed or directional information will ruin the clustering structure. For instance, in both examples, if we discard directional information, then $\mathcal{C}_0$ will look identical to $\mathcal{C}_1$ in the resulting meta-graph. On the other hand, if we discard signed information, $\mathcal{C}_1$ will look identical to $\mathcal{C}_2$.

We also note that the SDSBM model proposed here is a generalization of both the SSBM model from [3] and the Directed Stochastic Block Model from [5] when we have suitable meta-graph structures.

4.4.3 Real-World Data for Link Prediction

Standard Real-World Data Sets We consider four standard real-world signed and directed data sets. *BitCoin-Alpha* and *BitCoin-OTC* [132] describe bitcoin trading. *Slashdot* [154] is related to a technology news website, and *Epinions* [155] describes consumer reviews. These networks range in size from 3783 to 131580 nodes. Only *Slashdot* and *Epinions* are unweighted ($|w_{i,j}| = 1, \forall (v_i, v_j) \in \mathcal{E}$).

Novel Financial Data Sets from Stock Returns Using financial time series data, we build signed directed networks where the weighted edges encode lead-lag relationships inherent in the financial market, for each year in the interval 2000-2020. The lead-lag matrices are built from time series of daily price returns.[3] We refer to these networks as our **Fi**ancial **L**ead-**L**ag (FiLL) data sets. For each year in the data set, we build a signed directed graph (*FiLL-pvCLCL*) based on the price return of 444 stocks at market close times on consecutive days. We also build another graph (*FiLL-OPCL*), based on the price return of 430 stocks from market open to close. The difference between 444 versus 430 stems from the non-availability of certain open and close prices on some days for certain stocks. The lead-lag metric that is captured by the entry $\mathbf{A}_{i,j}$ in each network encodes a measure that quantifies the extent to which stock v_i leads stock v_j, and is obtained by computing the linear regression coefficient when regressing the time series (of length 245) of

[3]Raw CRSP data accessed through `https://wrds-www.wharton.upenn.edu/`.

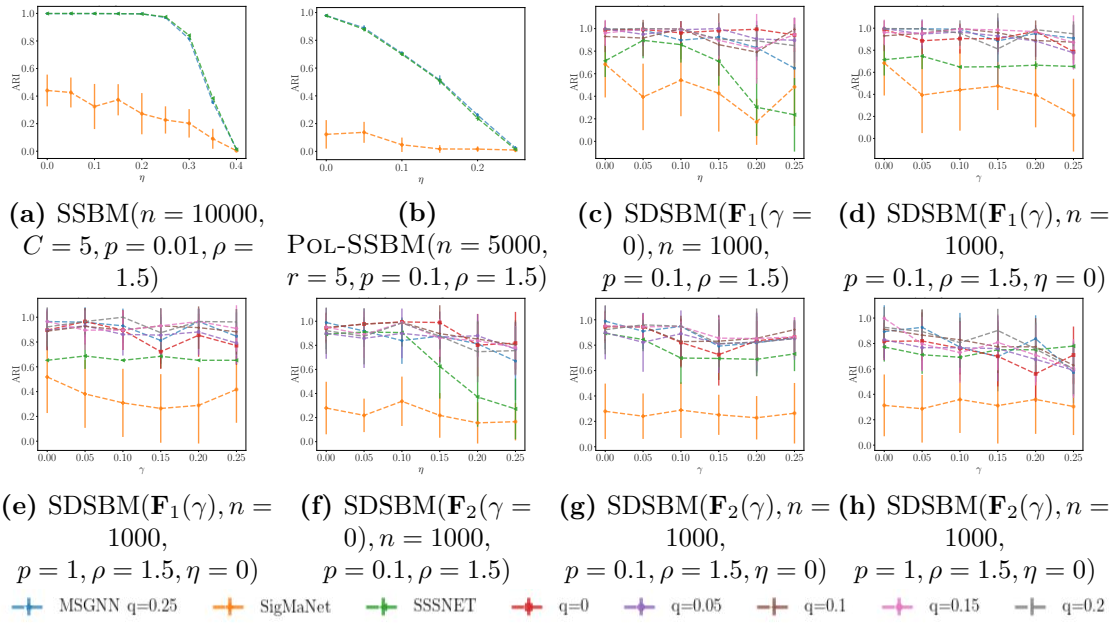
daily returns of stock v_i against the lag-one version of the time series (of length 245) of the daily returns of stock v_j. Specifically, we use the beta coefficient of the corresponding simple linear regression, to serve as the one-day lead-lag metric. The resulting matrix is asymmetric and signed, rendering it amenable to a signed, directed network interpretation. The initial matrix is dense, with nonzero entries outside the main diagonal, since we do not consider the own auto-correlation of each stock. Note that an alternative approach to building the directed network could be based on Granger causality [156, 157], or other measures that quantify the lead-lag between a pair of time series, potentially while accounting for nonlinearity, such as second-order log signatures from rough paths theory as in [105].

Next, we sparsify each network, keeping only 20% of the edges with the largest magnitudes. We also report the average results across the all the yearly data sets (a total of 42 networks) where the data set is denoted by *FiLL (avg.)*. To facilitate future research using these data sets as benchmarks, both the dense lead-lag matrices and their sparsified counterparts have been made publicly available.

4.4.4 Experimental Results

We compare MSGNN against representative GNNs which are described in Section 4.2. The six methods we consider are (1) SGCN [48], (2) SDGNN [133], (3) SiGAT [137], (4) SNEA [54], (5) SSSNET [3], and (6) SigMaNet [138]. For all link prediction tasks, comparisons are carried out on all baselines; for the node clustering tasks, we only compare MSGNN against SSSNET and SigMaNet as adapting the other methods to this task is nontrivial. In all of our experiments, we use the normalized Magnetic signed Laplacian, $\mathbf{L}_N^q$, unless otherwise stated. Implementation details are provided in Appendix C.2, along with a runtime comparison which shows that MSGNN is generally the fastest method, see Table C.1 in Appendix C.2. Extended results are in Appendix C.3 and C.4. Code and preprocessed data are available at `https://github.com/SherylHYX/MSGNN` and have been included in the open-source library *PyTorch Geometric Signed Directed* [28].

Node Clustering



(a) SSBM($n = 10000$, $C = 5, p = 0.01, \rho = 1.5$) **(b)** Pol-SSBM($n = 5000$, $r = 5, p = 0.1, \rho = 1.5$) **(c)** SDSBM($\mathbf{F}_1(\gamma = 0), n = 1000$, $p = 0.1, \rho = 1.5$) **(d)** SDSBM($\mathbf{F}_1(\gamma), n = 1000$, $p = 0.1, \rho = 1.5, \eta = 0$)

(e) SDSBM($\mathbf{F}_1(\gamma), n = 1000$, $p = 1, \rho = 1.5, \eta = 0$) **(f)** SDSBM($\mathbf{F}_2(\gamma = 0), n = 1000$, $p = 0.1, \rho = 1.5$) **(g)** SDSBM($\mathbf{F}_2(\gamma), n = 1000$, $p = 0.1, \rho = 1.5, \eta = 0$) **(h)** SDSBM($\mathbf{F}_2(\gamma), n = 1000$, $p = 1, \rho = 1.5, \eta = 0$)

Figure 4.2: Node clustering test ARI comparison on synthetic data. Error bars indicate one standard error. Results are averaged over ten runs — five different networks, each with two distinct data splits.

Figure 4.2 compares the node clustering performance of MSGNN with two other signed GNNs on synthetic data, and against variants of MSGNN on SDSBMs. For signed, undirected networks q does not has an effect, and hence we only report one MSGNN variant. Error bars are given by one standard error. We conclude that MSGNN outperforms SigMaNet on all data sets and is competitive with SSSNET.

On the majority of data sets, MSGNN achieves leading performance, whereas on some signed undirected networks (SSBM and Pol-SSBM) it is slightly outperformed by SSSNET. On these relatively small data sets, MSGNN and SSSNET have comparable runtime and are faster than SigMaNet. Comparing the MSGNN variants, we conclude that the directional information in these SDSBM models plays a vital role since MSGNN usually performs better with nonzero q.

Link Prediction

Our results for link prediction in Table 4.1 indicate that MSGNN is the top performing method, achieving the highest accuracy in **all** 25 cases. SNEA is among

Table 4.1: Test accuracy (%) comparison the signed and directed link prediction tasks introduced in Sec. 4.4.1. The best is marked in **bold red** and the second best is marked in underline blue .

Data Set	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
BitCoin-Alpha	SP	64.7±0.9	64.5±1.1	62.9±0.9	64.1±1.3	67.4±1.1	47.8±3.9	**71.3±1.2**
	DP	60.4±1.7	61.5±1.0	61.9±1.9	60.9±1.7	68.1±2.3	49.4±3.1	**72.5±1.5**
	3C	81.4±0.5	79.2±0.9	77.1±0.7	83.2±0.5	78.3±4.7	37.4±16.7	**84.4±0.6**
	4C	51.1±0.8	52.5±1.1	49.3±0.7	52.4±1.8	54.3±2.9	20.6±6.3	**58.5±0.7**
	5C	79.5±0.3	78.2±0.5	76.5±0.3	81.1±0.3	77.9±0.3	34.2±6.5	**81.9±0.9**
BitCoin-OTC	SP	65.6±0.9	65.3±1.2	62.8±1.3	67.7±0.5	70.1±1.2	50.0±2.3	**73.0±1.4**
	DP	63.8±1.2	63.2±1.5	64.0±2.0	65.3±1.2	69.6±1.0	48.4±4.9	**71.8±1.1**
	3C	79.0±0.7	77.3±0.7	73.6±0.7	82.2±0.4	76.9±1.1	26.8±10.9	**83.3±0.7**
	4C	51.5±0.4	55.3±0.8	51.2±1.8	56.9±0.7	57.0±2.0	23.3±7.4	**59.8±0.7**
	5C	77.4±0.7	77.3±0.8	74.1±0.5	80.5±0.5	74.0±1.6	25.9±6.2	**80.9±0.9**
Slashdot	SP	74.7±0.5	74.1±0.7	64.0±1.3	70.6±1.0	86.6±2.2	57.9±5.3	**92.4±0.2**
	DP	74.8±0.9	74.2±1.4	62.8±0.9	71.1±1.1	87.8±1.0	53.0±4.0	**93.1±0.1**
	3C	69.7±0.3	66.3±1.8	49.1±1.2	72.5±0.7	79.3±1.2	42.0±7.9	**86.1±0.3**
	4C	63.2±0.3	64.0±0.7	53.4±0.2	60.5±0.6	72.7±0.6	25.7±8.9	**78.2±0.3**
	5C	64.4±0.3	62.6±2.0	44.4±1.4	66.4±0.5	70.4±0.7	19.3±8.6	**76.8±0.6**
Epinions	SP	62.9±0.5	67.7±0.8	63.6±0.5	66.5±1.0	78.5±2.1	53.3±10.6	**85.4±0.5**
	DP	61.7±0.5	67.9±0.6	63.6±0.8	66.4±1.2	73.9±6.2	49.0±3.2	**86.3±0.3**
	3C	70.3±0.8	73.2±0.8	52.3±1.3	72.8±0.2	72.7±2.0	30.5±8.3	**83.1±0.5**
	4C	66.7±1.2	71.0±0.6	62.3±0.5	69.5±0.7	70.2±5.2	29.9±6.4	**78.7±0.9**
	5C	73.5±0.8	76.6±0.7	52.9±0.7	74.2±0.1	70.3±4.6	22.1±6.1	**80.5±0.5**
FiLL (avg.)	SP	88.4±0.0	82.0±0.3	76.9±0.1	90.0±0.0	88.7±0.3	50.4±1.8	**90.8±0.0**
	DP	88.5±0.1	82.0±0.2	76.9±0.1	90.0±0.0	88.8±0.3	48.0±2.7	**90.9±0.0**
	3C	63.0±0.1	59.3±0.0	55.3±0.1	64.3±0.1	62.2±0.3	33.7±1.3	**66.1±0.1**
	4C	81.7±0.0	78.8±0.1	70.5±0.1	83.2±0.1	80.0±0.3	24.9±0.9	**83.3±0.0**
	5C	63.8±0.0	61.1±0.1	55.5±0.1	**64.8±0.1**	60.4±0.4	19.8±1.1	**64.8±0.1**

the best performing methods, but is the least efficient in speed due to its use of graph attention, see runtime comparison in Appendix C.2. Specifically, the "avg." results for the novel financial data sets first average the accuracy values across all individual networks (a total of 42 networks), then report the mean and standard deviation over the five runs. Results for individual *FiLL* networks are reported in Appendix C.4. Note that ±0.0 in the result tables indicates that the standard deviation is less than 0.05%.

Ablation Study and Discussion

Table C.2 in Appendix C.3 compares different variants of MSGNN on the link prediction tasks, with respect to (1) whether we set $q = 0$ or use a value $q = q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$ which strongly emphaszies directional information; (2) whether to include sign in input node features (if False, then only in- and out-degrees are computed like in [6] regardless of edge signs, otherwise features

are constructed based on the positive and negative subgraphs separately); and (3) whether we take edge weights into account (if False, we view all edge weights as having magnitude one). Taking the standard errors into account, we find that incorporating directionality into the Laplacian matrix (i.e., having nonzero q) typically leads to slightly better performance in the directionality-related tasks (DP, 3C, 4C, 5C). Although, for *FiLL*, the $q = q_0$ values are within one standard deviation of the $q = 0$ ones for (T,T). The only example where $q = 0$ is clearly better is *Epinions* with (T,T) and task 4C. Hence, recommending $q = q_0$ is sensible. A further comparison of the role of q is provided in Table C.3 and shows that nonzero q values usually deliver superior performance.

Moreover, signed features are in general helpful for tasks involving sign prediction. For constructing weighted features we see no significant difference in simply summing up entries in the adjacency matrix compared to summing the absolute values of the entries. Besides, calculating degrees regardless of edge weight magnitudes could be helpful for the first four data sets but not for *FiLL*. In the first four data sets the standard errors are much larger than the averages of the sums of the features, whereas in the *FiLL* data sets, the standard errors are much smaller than the average, see Table C.16. Hence this feature may not show enough variability in the *FiLL* data sets to be very informative. Treating negative edge weights as the negation of positive ones is also not helpful (by not having separate degree features for the positive and negative subgraphs), which may explain why SigMaNet performs poorly in most scenarios due to its undesirable property. Surprisingly often, including only signed information but not weighted features does well. To conclude, constructing features based on the positive and negative subgraphs separately is helpful, and including directional information is generally beneficial.

More discussions on using instead the unnormalized Laplacian, or including more layers, and exploration of some properties of our proposed Laplacian, are provided in Appendix C.3.

4.5 Conclusion and Outlook

In this paper, we propose a spectral GNN based on a novel magnetic signed Laplacian matrix, introduce a novel synthetic network model and new real-world data sets, and conduct experiments on node clustering and link prediction tasks that are not restricted to considering either link sign or directionality alone. MSGNN performs as well or better than leading GNNs, while being considerably faster on real-world data sets. Future plans include investigating more properties of the proposed Laplacian, and an extension to temporal/dynamic graphs, where node features and/or edge information could evolve over time [7, 158]. We are also interested in extending our work to being able to encode nontrivial edge features, to develop objectives which explicitly handle heterogeneous edge densities throughout the graph, and to extend our approach to hypergraphs and other complex network structures.

Statement of Authorship for joint/multi-authored papers for PGR thesis

Title of Paper:	*MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian*
Publication Status	▪ Published
Publication Details	*He, Yixuan, Michael Perlmutter, Gesine Reinert, and Mihai Cucuringu. "MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian." In Learning on Graphs Conference, pp. 40-1. PMLR, 2022.*

Student Confirmation

Student name:	Yixuan He
Contribution to the paper	*This is my work, the topic of which was raised by my collaborator Prof. Michael Perlmutter. The idea of the Laplacian formulation came from discussions between my collaborator Prof. Michael Perlmutter and me. The theoretical results were from my collaborator Prof. Michael Perlmutter and me. This project was conducted under the supervision of my supervisors Prof. Gesine Reinert and Prof. Mihai Cucuringu. I conducted the experiments and wrote the original manuscript.*
Signature: Yixuan He 何奕萱	**Date: 04/27/2024**

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Gesine Reinert	
Supervisor comments	The description is fine.
Signature: Gesine Reinert	**Date:** 29 April 2024

5

GNNRank: Learning Global Rankings from Pairwise Comparisons via Directed Graph Neural Networks

5.1 Introduction

Recovering global rankings from pairwise comparisons reflecting relative latent strengths or scores is a fundamental problem in information retrieval [11, 12] and beyond. When analyzing large-scale data sets, one often seeks various forms of rankings (i.e. orderings) of the data for the purpose of identifying the most important entries, efficient computation of search & sort operations, or for extracting the main features. There is a swarm of applications employing ranking techniques ranging from Amazon's Mechanical Turk system for crowdsourcing [13] to the movie recommendation system provided by Netflix [14], and modeling outcomes of football matches [15].

A very rich literature on ranking traces back to [159], who studied recovering the ranking of a set of players from pairwise comparisons reflecting a total ordering. The last decades have seen a flurry of methods for ranking from pairwise comparisons, mostly based on spectral methods leveraging the eigenvectors of suitably defined matrix operators built directly from the data, which will be detailed in the related

work. In particular, SerialRank [25] shows promising ability in ranking for a set of totally ordered items without ties, by introducing a specific inductive bias from the so-called *seriation* problem [160]: a Fiedler vector of a certain similarity matrix could recover true rankings given enough pairwise comparisons in the noiseless setting. The *Fiedler value* of a symmetric and nonnegative matrix (or a graph) is defined to be the second smallest eigenvalue of the combinatorial Laplacian of the matrix [161], and its corresponding eigenvector is called a *Fiedler vector*. The Fiedler vector is able to encode useful information of a graph, such as bi-partitions of a graph [161] and the seriation problem [160], and can be employed for partitioning hypergraphs [162].

Moreover, there has been promising progress on combinatorial optimization in machine learning [163, 164], especially graph neural networks (GNNs) [1, 2], due to their potential in data exploration. Compared with their great success in many combinatorial tasks, the capability of GNNs in ranking tasks is not well developed. The few existing works are restricted to specific settings e.g. top-n personalized recommendations [163], and approximating centrality measures [164]. Another technical gap is the inability to learn a model by directly optimizing the ranking objective, which we aim to fill in our work.

We propose GNNRank, an end-to-end ranking framework compatible with existing GNN models e.g. [5, 114] that is able to learn directed graph (digraph) node embeddings. We devise differentiable objectives to encode ranking upsets/violations. Following the standard protocol in [165], an upset of an edge is a ranking violation; the inferred relative ranking is in the opposite direction to what the original measurement indicates. GNNRank consists of a family of ranking score estimation techniques, and adds an inductive bias by unfolding the Fiedler vector computation of the graph constructed from a *learnable* similarity matrix.

Our main contributions are as follows:

- To the best of our knowledge, this is *the first neural network framework* (specifically GNN) to recover global rankings from pairwise comparisons whereby a direct optimization of the ranking objectives is enabled, without supervision. Our method differs from the learning-free methods including

SerialRank because we design two novel differentiable losses tailored to neural networks, while the metric $\mathcal{L}_{upset,naive}$ used in previous works is piecewise constant and only used for post-evaluation instead of training the algorithms. Along the way, we adapt the widely used $\mathcal{L}_{upset,naive}$ metric to a more fined-grained evaluation method. Thus the motivation is also different from existing works.

- For jointly solving global ranking and GNN training, we design and introduce an inductive bias as achieved by the proximal gradient steps to our neural network, as unfolded from the Fiedler vector calculation, as an effective way of encoding latent orderings. The technique of constructing the Q matrix and the transformation of the optimization problem for differentiable computation of Fiedler eigenvectors as part of a deep learning model is nontrivial and, to the best of our knowledge, novel.

- Our methods empirically attain competitive, and often superior, accuracy compared to state-of-the-art methods, and its cost-effectiveness is especially pronounced when the trained GNN model is transferred to new datasets for ranking recovery. Compared to some existing methods e.g. Minimum Violation Rank (MVR) [166] with expensive optimizations, our methods have better asymptotic time complexity.

- From a computational perspective, when the data may have a temporal dimension or when there are similarities across different data sets, one can apply an already trained model to new data sets that are similar to the one the model has been trained on.

- We provide theoretical convergence guarantees for our method, with a technically novel proof, which in our view is a considerable advantage compared to other, ad-hoc, neural solvers.

5.2 Related Work

Ranking Methods One of the most popular models in the ranking literature is the Bradley-Terry-Luce (BTL) model [167], [168]. In its basic version, the probability that player i beats player j is given by $P_{ij} = \frac{w_i}{w_i+w_j}$, where the parameter vector $w \in \mathbb{R}^n_+$ is estimated from the data, and w_i is a proxy for the strength of player i. [169] facilitates the specification and fitting of Bradley-Terry logit models to pairwise comparisons.

Employing the stationary distribution of a suitably defined Markov chain for the ranking task traces its roots in early work on the topic of *network centrality*. Such measures have been designed to quantify the extent to which nodes of the graph (or other network structures) are most important [170]; see for example [171], [172], and [173].

David's Score [174] computes rankings from proportions of wins and losses. Minimum Violation Rank (MVR) [166] encompasses a suite of methods that aim to directly minimize certain penalty functions at the level of each upset. In our experiments, we compare against the algorithm of [175], that considers a linear relaxation of an integer program that minimizes a so-called agony loss. However, MVR is computationally expensive as also will be shown in our experiments.

SyncRank [176] formulates the ranking problem with incomplete noisy pairwise information as an instance of the group synchronization problem over the group SO(2) of planar rotations [177], which has attracted significant attention in recent years. The SpringRank algorithm of [178] borrows intuition from statistical physics, and proposes to infer hierarchical rankings in directed networks by solving a linear system of equations. [165] introduces simple algorithms for ranking and synchronization based on singular value decompositions, with theoretical guarantees. SerialRank [25] first computes the Laplacian from a certain similarity matrix $\mathbf{S}'$. The corresponding Fiedler vector of $\mathbf{S}'$ then serves as the final ranking estimate. The intuition is that the more similar two players are (in terms of the pattern of incoming/outgoing edges), the more similar their ranking should be; indeed, if two players defeat, and are defeated, by the same set of other players, then they

are likely to have a similar ranking/strength. In this classical ordering problem (called *seriation* [160]), one is given a similarity matrix between a set of items and assumes that the items can be ordered along a chain such that the similarity between items decreases with their distance in the chain.

Apart from the above classical learning-free optimizers for ranking problems, there also exist learning-based (mostly neural) models for similar tasks. The early work [179] applies a certain form of GNN to ranking web pages, while it requires label supervision for ground-truth ranks for training. While the authors in [180] propose the family of so-called RankGNNs, their competitors are graphs. [181] applies a neural network approach for preference learning, [182] generalizes [183], but these methods require queries as input, which solve a different problem from ours. [164] proposes the first GNN-based model to approximate betweenness and closeness centrality, facilitating locating influential nodes in the graphs in terms of information spread and connectivity. The pairwise direction is rarely considered in these works but it is important for the problem studied in this paper. Thus, our task differs fundamentally from the (abundant) learning-to-rank literature.

Directed Graph Neural Networks Directed GNNs are useful in learning digraph node embeddings. [114] builds aggregators based on higher-order proximity. [6] constructs a complex Hermitian Laplacian matrix. [5] introduces imbalance objectives for digraph clustering. In GNNRank, existing digraph neural networks can be readily incorporated.

Unfolding Techiques Algorithm unfolding [184] was first introduced to unfold the iterations as a cascade of layers while adding learnable parameters. The unfolding idea has later been applied to problems such as semantic segmentation [185] and efficient power allocation [186]. The work [187] discusses a new family of GNN layers designed to mimic and integrate the update rules of classical iterative algorithms. The unfolding idea inspires us to add a useful inductive bias from the calculation of the Fiedler vector via our bi-level optimization pipeline [188]. Most importantly, unfolding allows us to pass gradients through the optimization of a useful function.

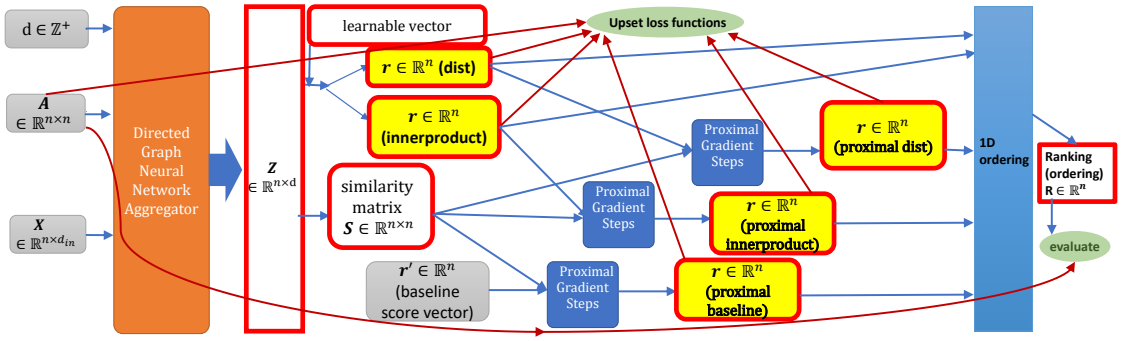


Figure 5.1: Overview of GNNRank based on directed graph neural networks and the proximal gradient steps corresponding to Algo. 2: starting from an adjacency matrix $\mathbf{A}$ which encodes pairwise comparisons, input feature matrix $\mathbf{X}$ and embedding dimension d, GNNRank first applies a directed graph neural network model to learn node embeddings $\mathbf{Z}$ for each competitor (node). Then it calculates the inner product or the similarity score with respect to a learnable vector to produce non-proximal outcomes for ranking scores ("innerproduct" or "dist"). Proximal variants start from a similarity matrix constructed from the learnable embeddings $\mathbf{Z}$, then utilize proximal gradient steps to output ranking scores. Depending on the initial guess score vector $\mathbf{r}'$, the proximal variants have names "proximal innerproduct", "proximal dist" or "proximal baseline". Ordering the scores in the score vector $\mathbf{r}$ induces the final ranking/ordering vector $\mathbf{R} \in \mathbb{R}^n$. The loss function is applied to a variant's output score vector $\mathbf{r}$, given the input adjacency matrix $\mathbf{A}$, while the final evaluation is based on $\mathbf{R}$ and $\mathbf{A}$. Red frames indicate trainable tensors/vectors/matrices. Grey squares correspond to fixed inputs.

5.3 Approach

5.3.1 Problem Definition

Without loss of generality, we consider pairwise comparisons in a competition, which can be encoded in a directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V}$ denotes competitors, and the edge set $\mathcal{E}$ represents pairwise comparisons. The outcomes of the matches can be captured by the adjacency matrix $\mathbf{A}$. A single edge $e \in \mathcal{E}$ from node v_i to node v_j, with edge weight $\mathbf{A}_{i,j} \geq 0$ that is not reciprocal, denotes that node v_i is stronger than node v_j by $\mathbf{A}_{i,j}$ For a reciprocal edge, $\mathbf{A}_{i,j}$ and $\mathbf{A}_{j,i}$ could be different; they could denote the (sum of) match scores for both competitors in matches between them, or the sum of absolute wins across different matches between them, where $\mathbf{A}_{i,j}$ is the sum of absolute wins for v_i. Recovering global rankings from pairwise comparisons amounts to assigning an integer R_i to each node $v_i \in \mathcal{V}$, denoting its position among competitors, where

the lower the rank, the stronger the node is. To this end, many existing methods (including our proposed methods) first learn a real-valued ranking score r_i for v_i, where the higher the score, the stronger the node is. The scores are then ordered to provide the final integer ranking.

5.3.2 Motivation and Connection to SerialRank

The whole GNNRank framework, described in Algo. 2, uses steps from Algo. 1 for the lower-level optimization within the whole bi-level optimization pipeline; a diagram is provided in Fig. 5.1. Details are provided later in this section.

Indeed, we highlight an intrinsic connection to SerialRank [25], which operates by first computing the Laplacian from a certain similarity matrix $\mathbf{S}' = \frac{1}{2}(n\mathbf{1}\mathbf{1}^\top + \mathbf{C}\mathbf{C}^\top)$, where $\mathbf{C}$ is the binary comparison matrix with $\mathbf{C}_{i,j} = \text{sign}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})$, and $\mathbf{A}$ the digraph adjacency matrix. The corresponding Fiedler vector of $\mathbf{S}'$ then serves as the final ranking estimate, after a global sign reconciliation. While often effective in practice, SerialRank is heavily dependent on the quality of the underlying similarity matrix $\mathbf{S}'$.

To address this issue, we introduce a parameterized GNN model that allows us to compute trainable measures of similarity that are useful for subsequent ranking. However, for training purposes, we of course need to somehow back-propagate gradients through the computation of a Fiedler vector to update the GNN parameters. Because it is generally difficult to directly pass gradients through eigenvector computations, we instead express the Fiedler vector as the solution of a constrained optimization problem. We then approximate the solution of this problem using proximal gradient steps, each of which are themselves differentiable with respect to the underlying optimization variables, and ultimately by the chain rule, the GNN parameters. Note that Algo. 1 can be viewed as a *differentiable* function that inputs $\mathbf{r}'$ and $\mathbf{L}$, and outputs $\mathbf{r}$. Therefore, the gradients can backpropagate *through* Algo. 1 into $\mathbf{L}$ and $\mathbf{r}'$, hence the similarity matrix $\mathbf{S}$ and the rest of the model parameters.

In broader contexts, this process is sometimes referred to as *unfolded optimization* [184], and is applicable in situations whereby a high-level loss function (in our case

a ranking loss) is defined with respect to the minimization of some lower-level, parameter-dependent optimization problem (e.g., Fiedler vector computation) that has been unfolded across differentiable iterations/updates. Within our proposed framework, this process allows to combine the inductive bias of Fiedler-vector-based rankings with the flexibility of GNNs for modeling relations between entities. Also, SerialRank can be viewed as a special case of GNNRank; $\mathbf{S}'_{i,j}$ counts the number of competitors that beat both v_i and v_j, plus that of competitors beaten by both v_i and v_j, minus that of competitors beating one of v_i and v_j but beaten by the other, plus half the number of nodes. Information from common neighbors could be aggregated by directed GNNs, and a kernel such as a Gaussian RBF kernel applied to the digraph embeddings could approximate $\mathbf{S}'$.

5.3.3 Loss Functions and Objectives

Define the skew-symmetric matrix $\mathbf{M}' = \mathbf{A} - \mathbf{A}^\top$, and let t be the number of nonzero elements in $\mathbf{M}'$. For a vector $\mathbf{r} = [r_1, \ldots, r_n]^\top$ with *real-valued* ranking scores (often viewed as skill levels) as entries, a naive upset is defined as the fraction of relationships disagreeing with their expected sign, in the spirit of [165]. Formally, let $\mathbf{1}$ be an all-one column vector, and define the matrix $\mathbf{T}' = \mathbf{r}\mathbf{1}^\top - \mathbf{1}\mathbf{r}^\top \in \mathbb{R}^{n\times n}$. Then we have $\mathbf{T}'_{i,j} = r_i - r_j, \forall i, j \in \{1, \cdots, n\}$. To not penalize entries where the initial pairwise rankings are not available, we only compare $\mathbf{T}'$ with $\mathbf{M}'$ at locations where $\mathbf{M}'$ has nonzero entries. A naive upset loss is defined as

$$\mathcal{L}_{\text{upset, naive}} = \sum_{i,j:\mathbf{M}'_{i,j}\neq 0} (\text{sign}(\mathbf{T}'_{i,j}) \neq \text{sign}(\mathbf{M}'_{i,j}))/t. \tag{5.1}$$

Being piecewise constant, $\mathcal{L}_{\text{upset, naive}}$ is not useful in gradient descent. To account for the difference in the scaling between the output scores and the input adjacency matrix, we define element-wise divisions $\mathbf{M} = \frac{\mathbf{A}-\mathbf{A}^\top}{\mathbf{A}+\mathbf{A}^\top}$, and $\mathbf{T} = \frac{\mathbf{r}\mathbf{1}^\top-\mathbf{1}\mathbf{r}^\top}{\mathbf{r}\mathbf{1}^\top+\mathbf{1}\mathbf{r}^\top} \in \mathbb{R}^{n\times n}$. Then $\mathbf{T}_{i,j} = \frac{r_i-r_j}{r_i+r_j}, \forall i, j \in \{1, \cdots, n\}$. Similarly, we only compare $\mathbf{T}$ with $\mathbf{M}$ at locations where $\mathbf{M}$ has nonzero entries.

With $\tilde{\mathbf{T}}$ where $\tilde{\mathbf{T}}_{i,j} = \mathbf{T}_{i,j}$ if $\mathbf{M}_{i,j} \neq 0$ and $\tilde{\mathbf{T}}_{i,j} = 0$ if $\mathbf{M}_{i,j} = 0$, the differentiable upset loss is then defined as

$$\mathcal{L}_{\text{upset, ratio}} = \left\|\tilde{\mathbf{T}} - \mathbf{M}\right\|_F^2 / t(\mathbf{M}), \tag{5.2}$$

where the subscript F means Frobenius norm, and $t(\mathbf{M})$ is the number of nonzero elements in $\mathbf{M}$.

Note that $\mathcal{L}_{\text{upset, ratio}}$ requires $r_i \geq 0$ for each r_i to represent the skill level. Hence we use transformations e.g. $r_i \leftarrow \text{sigmoid}(r_i)$ for general $r_i \in \mathbb{R}$, and $r_i \leftarrow \frac{r_i+1}{2}$ when we know $r_i \geq -1$, e.g. when the score $\mathbf{r} = (r_i)$ has unit norm.

Another choice is the margin loss $\mathcal{L}_{\text{upset, margin}}$, with a tunable nonnegative parameter $\epsilon \geq 0$ (default 0.01), defined as

$$\mathcal{L}_{\text{upset, margin}} = \sum_{i,j} (\mathbf{M}_{i,j} + |\mathbf{M}_{i,j}|) \cdot \text{ReLU}(r_j - r_i + \epsilon)/t(\mathbf{M}), \tag{5.3}$$

where ReLU is the Rectified Linear Unit.

The training loss used is either $\mathcal{L}_{\text{upset, ratio}}$, $\mathcal{L}_{\text{upset, margin}}$ or their sum, where the choice is set as a hyperparameter.

Evaluation. For evaluation, in addition to $\mathcal{L}_{\text{upset, naive}}$, we introduce another objective similar to $\mathcal{L}_{\text{upset, naive}}$ but penalizing predicted signs opposite to the actual signs more than predicted signs being zero, i.e. we distinguish predictions as ties or being opposite signs, which can also take integer rankings as input

$$\mathcal{L}_{\text{upset, simple}} = \|\text{sign}(\mathbf{T}') - \text{sign}(\mathbf{M}')\|_F^2 / t, \tag{5.4}$$

where the sign function acts element-wise, and t is the number of nonzero elements in $\mathbf{M}'$. However, although $\mathcal{L}_{\text{upset, simple}}$ distinguishes ties and opposite signs, this loss can easily be made equal to one by assigning the same score to all nodes (i.e., making $\mathbf{r}$ a constant vector which corresponds to all ties), which means that whenever we achieve a value larger than one, the model performs even worse than trivial guess.

When ground-truth rankings are available, we use the Kendall tau [24] values for evaluation, and select the model based on the lowest $\mathcal{L}_{\text{upset, simple}}$.

5.3.4 Obtaining Directed Graph Embeddings

For obtaining a directed graph embedding, any GNN method which can take into account directionality and output node embeddings could be applied, e.g. DIMPA by [5], the inception block model (IB) [114], and MagNet [6]. In our experiments, we employ DIMPA and IB, to aid in the ranking task. Denoting the final node embedding by $\mathbf{Z} \in \mathbb{R}^{n\times d}$, the embedding vector $\mathbf{z}_i$ for a node v_i is $\mathbf{z}_i = (\mathbf{Z})_{(i,:)} \in \mathbb{R}^d$, the i^{th} row of $\mathbf{Z}$.

5.3.5 Obtaining Final Scores and Rankings

To obtain the final ranking score, we unfold the calculation of a Fiedler vector for the graph constructed from our symmetric similarity matrix $\mathbf{S}$ with proximal gradient steps.

Obtaining the similarity matrix. From the high-dimensional embedding matrix $\mathbf{Z}$, we calculate the symmetric similarity matrix $\mathbf{S}$ with $\mathbf{S}_{i,j} = \exp(-|\mathbf{z}_j - \mathbf{z}_i|_2^2/(\sigma^2 d))$ where $\sigma \in \mathbb{R}$ is the same trainable parameter as in "dist". Denote by $\mathbf{D}$ the diagonal matrix with $\mathbf{D}_{i,i} = \sum_j \mathbf{S}_{i,j}$. We consider the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$, and apply proximal gradient to approximate a Fiedler vector of $\mathbf{S}$, which then serves as $\mathbf{r}$.

Transformation of the Optimization Problem. Computing a Fiedler vector of the similarity matrix $\mathbf{S}$ is equivalent to solving the optimization problem [189]

$$\min_{\mathbf{r}} \mathbf{r}^\top \mathbf{L}\mathbf{r} \text{ s.t.} \quad \|\mathbf{r}\|_2^2 = 1, \quad \mathbf{r}^\top \mathbf{1} = 0, \tag{5.5}$$

where $\mathbf{L}$ is the graph Laplacian matrix. We observe that the constraints describe an intersection of a unit sphere and a hyperplane. By rotating the problem and the constraints so that the hyperplane becomes cardinal, we can effectively fix one dimension to zero and solve

$$\min_{\mathbf{y}} \mathbf{y}^\top \mathbf{Q}\mathbf{L}\mathbf{Q}^\top \mathbf{y} \text{ s.t.} \quad \|\mathbf{y}\|_2^2 = 1, y_1 = 0. \tag{5.6}$$

Here $\mathbf{Q}$ is an orthogonal matrix. We choose a $\mathbf{Q}$ such that $\mathbf{Q}\mathbf{1} = \sqrt{n}\mathbf{e}_1$, where $\mathbf{e}_1 = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}^\top$. Let $\mathbf{y} = \mathbf{Q}\mathbf{r}$, and thus $\mathbf{r} = \mathbf{Q}^\top \mathbf{y}$. The problem becomes

$$\min_{\mathbf{y}} \mathbf{y}^\top \mathbf{Q}\mathbf{L}\mathbf{Q}^\top \mathbf{y}$$
$$\text{s.t.} \quad \|\mathbf{r}\|_2^2 = 1 \qquad \mathbf{r}^\top \mathbf{1} = \sqrt{n}\mathbf{y}^\top \mathbf{e}_1 = \sqrt{n} y_1 = 0.$$

Since we fix $y_1 = 0$, this is equivalent to

$$\min_{\mathbf{y}' \in \mathbb{R}^{n-1}} \mathbf{y}'^\top \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{2:n,2:n} \mathbf{y}' \quad \text{s.t.} \quad \|\mathbf{y}'\|_2^2 = 1, \tag{5.7}$$

where $[\cdot]_{2:n,2:n}$ represents the matrix with its first row and first column removed. To illuminate this equivalence, since the constraint $\mathbf{r}^\top \mathbf{1} = 0$ is equivalent to $y_1 = 0$, we need to ensure that $y_1 = 0$ is maintained throughout. If we start with $\mathbf{y} \in \mathbb{R}^n$ where $y_1 = 0$, and let $\mathbf{y}' = [y_2, \ldots, y_n]^\top \in \mathbb{R}^{n-1}$ then

$$\mathbf{y}^\top \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{1:n,1:n} \mathbf{y} = \sum_{1 \le i \le n, 1 \le j \le n} \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{i,j} y_i y_j$$
$$= \sum_{2 \le i \le n, 2 \le j \le n} \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{i,j} y_i y_j = \mathbf{y}'^\top \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{2:n,2:n} \mathbf{y}'.$$

One possible $\mathbf{Q}$, with details of the construction given in Appendix D.2, is the following upper Hessenberg matrix, which can be efficiently precomputed

$$\mathbf{Q}_{ij} = \begin{cases} \sqrt{\frac{1}{n}} & i = 1 \\ -\sqrt{\frac{n-i+1}{n-i+2}} & i \ge 2, j = i-1 \\ \sqrt{\frac{1}{(n-i+1)(n-i+2)}} & i \ge 2, j \ge i \\ 0 & \text{otherwise.} \end{cases}$$

Proximal Gradient Steps. To enforce a zero in the first entry of the initial guess, we zero-center the input score vector, then left multiply it by $\mathbf{Q}$, so this resulting vector has 0 in its first entry. We then remove the first entry of the resulting vector, and discard the first row and first column for the matrix $\mathbf{Q}\mathbf{L}\mathbf{Q}^\top$. The gradient of the objective $\mathbf{y}^\top \widetilde{\mathbf{L}} \mathbf{y}$ with respect to $\mathbf{y}$ is $(\widetilde{\mathbf{L}} + \widetilde{\mathbf{L}}^\top)\mathbf{y} = 2\widetilde{\mathbf{L}}\mathbf{y}$ since $\widetilde{\mathbf{L}}$ is symmetric. We set the number of proximal steps as $\Gamma = 5$, and initial learning rates inside proximal gradient steps $\alpha_\gamma = 1, \gamma = 1, \ldots, \Gamma$. Define the spherical projection operation $\mathcal{P}_{\mathcal{S}^{n-1}}(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ by $\mathcal{P}_{\mathcal{S}^{n-1}}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ if $\mathbf{x} \neq \mathbf{0}$ and $\mathcal{P}_{\mathcal{S}^{n-1}}(\mathbf{0}) = \mathbf{e}_1$, where $\mathbf{0} = [0, \ldots, 0]^\top$. Algo. 1 details the proximal gradient steps, with proximal operator $\mathcal{P}_{\mathcal{S}^{n-2}}$, which guarantees descent of the optimization objective in equation 5.5 given suitable α's (see Thm. 3).

Algorithm 1: Proximal Gradient Steps

Input: Initial score $\mathbf{r}' \in \mathbb{R}^n$, Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$ and $\mathbf{Q} \in \mathbb{R}^{n \times n}$
Parameter: (Initial) learning rate set $\{\alpha_\gamma > 0\}_{\gamma=1}^{\Gamma}$ that could either be fixed or trainable (default: trainable).
Output: Updated score vector $\mathbf{r} = [r_1, \ldots, r_n]^\top$.

Let $\mathbf{y} = \mathbf{r}' - \sum_{i=1}^n r_i'/n$;
$\mathbf{y} \leftarrow \mathbf{Q}'\mathbf{y} \in \mathbb{R}^{n-1}$ ($\mathbf{Q}'$ is $\mathbf{Q}$ with the first row removed);
$\mathbf{y} \leftarrow \mathcal{P}_{\mathcal{S}^{n-1}}(\mathbf{y})$ to have unit 2-norm;
$\widetilde{\mathbf{L}} \leftarrow \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{2:n,2:n}$.
for $\gamma < \Gamma$ **do**
    $\mathbf{y} \leftarrow \mathbf{y} - \alpha_\gamma(2\widetilde{\mathbf{L}})\mathbf{y}/n$;
    $\mathbf{y} \leftarrow \mathcal{P}_{\mathcal{S}^{n-2}}(\mathbf{y})$ to have unit 2-norm;
    $\gamma \leftarrow \gamma + 1$.
end for
$\mathbf{y} \leftarrow \text{CONCAT}(0, \mathbf{y}) \in \mathbb{R}^n$;
$\mathbf{r} = \mathbf{Q}^\top\mathbf{y}$
return $\mathbf{r}$;

Algorithm 2: GNNRank: Proposed Ranking Framework

Input: Digraph adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, node feature matrix $\mathbf{X}$, variant name *var*, (optional) initial guess from baseline $\mathbf{r}'$.
Parameter: Learnable vector $\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}, \sigma \in \mathbb{R}$, GNN parameters, parameters from Algo. 1.
Output: Score vector $\mathbf{r}$.

$\mathbf{Z} \in \mathbb{R}^{n \times d} = \text{GNN}(\mathbf{A}, \mathbf{X})$;
if *var* ∈{"dist", "proximal dist"} **then**
    Compute $\mathbf{r} : r_i = \exp(-|\mathbf{a} - \mathbf{z}_i|_2^2/(\sigma^2 d))$;
else if *var* ∈{"innerproduct", "proximal innerproduct"} **then**
    Compute $\mathbf{r} : r_i = \text{sigmoid}(z_i \cdot \mathbf{a} + b)$;
end if
if *var* ∈{"proximal dist", "proximal innerproduct", "proximal baseline"} **then**
    Compute $\mathbf{S} : \mathbf{S}_{i,j} = \exp(-|\mathbf{z}_j - \mathbf{z}_i|_2^2/(\sigma^2 d))$;
    Compute Laplacian $\mathbf{L}$ and $\mathbf{Q}$ by Sec. 5.3.5;
    if *var* ==={"proximal baseline"} **then**
        $\mathbf{r} = \mathbf{r}'$;
    end if
    $\mathbf{r} \leftarrow$ Proximal Gradient Steps$(\mathbf{r}, \mathbf{L}, \mathbf{Q})$ from Algo. 1;
end if
return $\mathbf{r}$;

5.3.6 Initialization and Pretraining Considerations

Algo. 1 requires an initial guess $\mathbf{r}' \in \mathbb{R}^n$. To achieve this, we introduce two non-proximal variants, whose output score vector could serve as $\mathbf{r}'$. These non-proximal

variants are also evaluated in our experiments.

• (1) *"innerproduct"* variant: With a trainable vector $\mathbf{a}$ that has its dimension equal to the embedding dimension, we obtain the scores by the inner product of $\mathbf{z}_i$ with $\mathbf{a}$, plus a trainable bias b, followed by a sigmoid layer to force positive score values: $r_i = \text{sigmoid}(\mathbf{z}_i \cdot \mathbf{a} + b)$.

• (2) *"dist"* variant: $r_i = \exp(-|\mathbf{a} - \mathbf{z}_i|_2^2/(\sigma^2 d))$ where $\mathbf{a} \in \mathbb{R}^d, \sigma \in \mathbb{R}$ are trainable parameters. This variant applies a trainable Gaussian RBF kernel to describe the scores.

For each non-proximal variant, the corresponding proximal variant is called • (3) *"proximal dist"* or • (4) *"proximal innerproduct"*. We can also adopt the initial guess from a certain existing baseline's output, with variant name • (5) *"proximal baseline"*.

For a reasonable similarity matrix to start with, we apply some pretraining. One option is to train the ranking model with a non-proximal variant, "dist" or "innerproduct", for the early training epochs. As the proximal variants are inspired by unfolding the Fiedler vector calculation introduced in SerialRank [25], another option is to add a term to the loss function in early training epochs that compares the similarity matrix constructed by $\mathbf{Z}$ using a GNN with the normalized version (divided by the max entry so that the maximum is 1) of $\mathbf{S}'$, the similarity matrix from SerialRank. The corresponding additional term added to the loss function $\mathcal{L}_{\text{upset, ratio}}$ from Eq. equation 5.2 or $\mathcal{L}_{\text{upset, margin}}$ from Eq. equation 5.3, or their sum, is $\mathcal{L}_{\text{similarity}} = n^{-2}\|\mathbf{S} - \mathbf{S}'/\max(\{\mathbf{S}'_{i,j}\})\|_F^2$.

5.3.7 Convergence Analysis

An analysis of the convergence of our proximal gradient steps is provided in Appendix D.3, along with additional theoretical and practical considerations. We summarize our main result below; the proof is available in Appendix D.3.

Theorem 3. *Let $\{\alpha_\gamma > 0\}_{\gamma=1}^{\Gamma}$ in Algo. 1 be fixed (equal to α) and let ρ be the Fiedler eigenvalue of $\mathbf{S}$. Denote a Fiedler eigenvector by $\mathbf{r}^*$. Assume that $\mathbf{r}^*$ is a strict local minimizer of problem (5.5). If $0 < \alpha < \frac{1}{4(n-1)}$, then with our definition of $\mathbf{S}$, Algo. 1 converges locally uniformly to $\mathbf{r}^*$.*

5.4 Experiments

Implementation details are provided in Appendix D.1. Experiments were conducted on a compute node with 8 Nvidia Tesla T4, 96 Intel Xeon Platinum 8259CL CPUs @ 2.50GHz and 378GB RAM.

5.4.1 Data sets and Protocol

Real-World Data. We consider 10 real-world data sets (78 digraphs constructed in total). Digraphs based on pairwise comparisons are constructed as follows in most cases: for each match, an edge is added if the match result is not a tie, with the edge weight the difference between scores (absolute wins, that is, the absolute difference of the scores), and ties are treated the same as no match between the pair of players.

When raw data with individual match scores on both competitors are available, we in addition construct a *"finer"* version which takes ties into account, as follows: for each match, we have a reciprocal edge with potentially different weights on the two directions. If there are multiple matches, weights are added. To distinguish zero (a tie) from no game, for our methods, we add a small value (here 0.1) to every existing edge.

Real-world data sets, detailed in Appendix D.4, include

- NCAA College Basketball[1] (used to construct the networks *Basketball* and *Basketball finer*) are the outcomes of US College basketball matches from seasons 1985-2014. We construct two separate digraphs for each season, a regular version and a "finer" version introduced in the last paragraph.

- England Football Premier League[2] (*Football* and *Football finer*) are Premier League football match results for six seasons, from 2009 to 2014, between 20 teams. As in NCAA College Basketball, we construct two separate digraphs for each season.

[1] https://www.ncaa.com/sports/basketball-men/d1
[2] https://www.premierleague.com/

Table 5.1: Summary statistics for the real-world networks.

Data	n	$\lvert\mathcal{E}\rvert$	density	$\lvert\mathcal{E}^r\rvert$	$\frac{\lvert\mathcal{E}^r\rvert}{\lvert\mathcal{E}\rvert}(\%)$	K	d
HeadToHead	602	5010	1.38e-02	464	9.26	48	32
Finance	1315	1729225	1.00e+00	1729225	100	20	64
Animal	21	193	4.60e-01	64	33.16	3	8
Faculty:Business	113	1787	1.41e-01	0	0.00	5	16
Faculty:CS	206	1407	3.33e-02	0	0.00	9	16
Faculty:History	145	1204	5.77e-02	0	0.00	12	16
Football(avg)	20	201	5.29e-01	71	32.17	9	8
Basketball(avg)	316	3506	3.51e-02	986	28.57	20	16
Football finer (avg)	20	367	9.65e-01	367	100	9	8
Basketball finer (avg)	316	6139	6.12e-02	6139	100	20	16

- The Animal Dominance Network (*Animal*) [190] describes the number of net aggressive wins of 21 captive monk parakeets.
- Microsoft Halo 2 Tournament on Head-to-Head Games (*HeadToHead*) collects game outcomes during the Beta testing period for the Xbox game Halo 2[3].
- Faculty Hiring Networks (*Faculty: Business*, *Faculty: CS* and *Faculty: History*) [191] contains three North American academic hiring networks tracking the flow of academics between universities.
- Lead-Lag Relationships on Stocks (*Finance*) [105] contains lead-lag relationships on 1315 stocks from 2001-2019. An edge (i, j) in the digraph encodes the t-value of the coefficient in the regression of the daily returns of stock i on the lag-1 (previous day) returns of stock j.

Table 5.1 gives the number of nodes (n), the number of directed edges ($|\mathcal{E}|$), the number of reciprocal edges ($|\mathcal{E}^r|$) (self-loops are counted once and for $u \neq v$, a reciprocal edge $u \to v, v \to u$ is counted twice) and their percentage among all edges, for the real-world networks, illustrating the variability in network size and density (defined as $|\mathcal{E}|/[n(n-1)]$). When input features are unavailable, we stack the real and imaginary parts of the top K eigenvectors of $(\mathbf{A} - \mathbf{A}^\top) \cdot i$ in line with the protocol in [97] for GNNs. We report the embedding dimension d.

[3]Credits for using the Halo 2 Beta data set are given to Microsoft Research Ltd. and Bungie.

Synthetic Data We perform experiments on graphs with $n = 350$ nodes for Erdős-Rényi Outlier (ERO) models as in [165], with edge density $p \in \{0.05, 1\}$, noise level $\eta \in \{0, 0.1, \ldots, 0.8\}$ (corresponding to γ in [165]) and style "uniform" or "gamma" depending on the distribution from which the ground-truth scores are generated. [4] We fix $K = 5, d = 16$ for the ERO.

5.4.2 Main Experimental Results

In our numerical experiments, we compare against **11 baselines**, where results are averaged over 10 runs: • Eigenvector Centrality (Eig.Cent.) [172], • PageRank [171], • Rank Centrality (RankCent.) [173], • Minimum Violation Rank (MVR) [166], • SerialRank [25], • SyncRank [176], • SVD_NRS and SVD_RS by [165], • Bradley-Terry-Luce (BTL) model [169], • David's Score (DavidScore) [174], and • SpringRank [178]. Models are selected based on the lowest $\mathcal{L}_{\text{upset, simple}}$ obtained without label supervision, where non-proximal results (for "dist" and "innerproduct" variants) are listed in the "GNNRank-N" column and proximal methods (for "proximal dist", "proximal innerproduct" and "proximal baseline" variants) listed with "GNNRank-P", in all tables. We report the best-performing variant for each data set within GNNRank-N and GNNRank-P, respectively. Performance with respect to each of the GNNRank variants, for each individual input digraph, and on different objectives, are given in Appendix D.5. Table 5.2 compares our two groups of methods against baselines on 10 real-world data sets, where basketball data sets are averaged over 30 seasons, and football ones are averaged over 6 seasons. Our best-performing variant of the proximal method also outperforms its inspiration SerialRank [25] on all real-world data sets by including more trainable parameters. Our proximal method achieves state-of-the-art performance when using a typically good baseline, such as SyncRank, as an initial guess to the score vector before applying proximal gradient steps.

Table 5.3 shows Kendall tau selected from the lowest $\mathcal{L}_{\text{upset, naive}}$ on synthetic models. For some dense digraphs, SerialRank (which motivated our proximal

[4] This synthetic graph is an ER graph with tunable noise level, a test case which is informative without being too unrealistic.

Table 5.2: Performance on $\mathcal{L}_{\text{upset, simple}}$ (top half) and $\mathcal{L}_{\text{upset, naive}}$ (bottom half), averaged over 10 runs with one standard deviation. "avg" for time series first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best is marked in **bold red** while the second best is in underline blue. When MVR does not generate results after one week, we fill in "NAN".

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	MVR	GNNRank-N	GNNRank-P
HeadToHead	1.00±0.00	1.94±0.00	2.01±0.00	1.12±0.01	1.16±0.00	1.47±0.00	1.36±0.00	2.00±0.02	1.79±0.00	1.42±0.00	nan±nan	0.99±0.00	**0.96±0.00**
Finance	1.63±0.00	1.98±0.00	1.61±0.00	1.78±0.01	1.63±0.00	1.74±0.00	1.75±0.00	1.88±0.00	1.64±0.00	1.64±0.00	nan±nan	**1.00±0.00**	**1.00±0.00**
Animal	0.50±0.00	1.62±0.24	1.98±0.48	0.45±0.00	0.33±0.00	0.55±0.00	0.63±0.00	1.96±0.00	1.03±0.00	0.53±0.00	2.02±0.32	0.41±0.09	**0.25±0.00**
Faculty: Business	0.41±0.00	0.83±0.00	1.19±0.00	0.41±0.01	0.49±0.00	0.49±0.00	0.49±0.00	2.01±0.03	0.68±0.00	0.46±0.00	0.78±0.05	0.38±0.01	**0.36±0.00**
Faculty: CS	0.33±0.00	0.98±0.10	1.40±0.00	0.34±0.01	0.61±0.00	0.51±0.00	0.44±0.00	1.99±0.27	0.93±0.00	0.58±0.00	0.87±0.09	0.33±0.03	**0.32±0.00**
Faculty: History	0.32±0.00	0.57±0.00	2.16±0.80	0.30±0.01	0.57±0.00	0.40±0.00	0.37±0.00	2.13±0.30	0.95±0.00	0.38±0.00	0.84±0.17	**0.28±0.01**	0.30±0.01
Basketball (avg)	0.78±0.00	1.72±0.00	1.98±0.00	0.91±0.03	0.79±0.00	0.88±0.00	0.88±0.00	1.95±0.00	0.99±0.00	0.89±0.00	nan±nan	0.80±0.00	**0.73±0.00**
Basketball finer (avg)	0.81±0.00	1.73±0.01	1.96±0.00	1.39±0.02	0.85±0.00	1.19±0.00	1.15±0.00	1.97±0.00	1.00±0.00	0.90±0.00	nan±nan	0.84±0.00	**0.74±0.00**
Football (avg)	0.91±0.00	1.63±0.12	1.20±0.00	0.94±0.02	0.94±0.00	1.07±0.00	1.08±0.00	1.78±0.03	1.00±0.00	0.90±0.00	1.72±0.09	0.81±0.03	**0.78±0.02**
Football finer (avg)	0.98±0.00	1.68±0.03	1.16±0.00	1.01±0.02	0.93±0.00	1.13±0.00	1.21±0.00	1.91±0.01	1.00±0.00	0.90±0.00	2.06±0.04	0.89±0.06	**0.82±0.01**
HeadToHead	0.25±0.00	0.48±0.00	0.50±0.00	0.28±0.00	0.29±0.00	0.37±0.00	0.34±0.00	0.50±0.01	0.45±0.00	0.36±0.00	nan±nan	0.27±0.00	**0.24±0.00**
Finance	0.41±0.00	0.50±0.00	**0.40±0.00**	0.45±0.00	0.41±0.00	0.44±0.00	0.44±0.00	0.47±0.00	0.41±0.00	0.41±0.00	nan±nan	0.41±0.00	**0.40±0.00**
Animal	0.13±0.00	0.40±0.06	0.58±0.11	0.11±0.00	0.08±0.00	0.14±0.00	0.16±0.00	0.49±0.00	0.26±0.00	0.13±0.00	0.50±0.08	0.10±0.02	**0.06±0.00**
Faculty: Business	0.10±0.00	0.21±0.00	0.30±0.00	0.10±0.00	0.12±0.00	0.12±0.00	0.12±0.00	0.50±0.01	0.17±0.00	0.12±0.00	0.19±0.01	0.10±0.00	**0.09±0.00**
Faculty: CS	**0.08±0.00**	0.24±0.02	0.35±0.00	**0.08±0.00**	0.15±0.00	0.13±0.00	0.11±0.00	0.50±0.07	0.23±0.00	0.15±0.00	0.22±0.02	**0.08±0.01**	**0.08±0.00**
Faculty: History	0.08±0.00	0.14±0.00	0.54±0.20	0.08±0.00	0.15±0.00	0.10±0.00	0.09±0.00	0.53±0.08	0.24±0.00	0.10±0.00	0.21±0.04	**0.07±0.00**	**0.07±0.00**
Basketball (avg)	0.20±0.00	0.43±0.00	0.49±0.00	0.23±0.01	0.20±0.00	0.22±0.00	0.22±0.00	0.49±0.00	0.25±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball finer (avg)	0.20±0.00	0.43±0.00	0.49±0.00	0.35±0.00	0.21±0.00	0.30±0.00	0.29±0.00	0.49±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Football (avg)	0.23±0.00	0.41±0.03	0.30±0.00	0.23±0.01	0.24±0.00	0.27±0.00	0.27±0.00	0.44±0.01	0.25±0.00	0.22±0.00	0.43±0.02	0.22±0.01	**0.21±0.00**
Football finer (avg)	0.25±0.00	0.42±0.01	0.29±0.00	0.25±0.01	0.23±0.00	0.28±0.00	0.30±0.00	0.48±0.00	0.25±0.00	0.22±0.00	0.51±0.01	0.24±0.01	**0.21±0.00**

gradient steps) attains leading performance, while for some other cases it fails. GNNRank-P outperforms across all synthetic models shown here. Full results are in Appendix D.5.2.

We conclude that both non-proximal and proximal methods can achieve leading performance on real-world data sets, while on the synthetic models listed here, the best method in GNNRank-P performs much better than the best method in GNNRank-N. Performance results for each of the variants are provided in Appendix D.5.3; the individual variants also attain comparable and often superior performance compared to the baselines.

We observe across all data sets that our proximal methods: • (1) can improve on existing baseline methods when using them as initial guesses, and never perform significantly worse than the corresponding baseline, hence they can be used to enhance existing methods; • (2) do not rely on baseline methods for an initial guess but can instead use GNNRank-N outcomes, such as "proximal dist" and "proximal innerproduct"; • (3) can outperform SerialRank by unfolding its Fiedler vector calculations with a trainable similarity matrix and proximal gradient steps.

5.4.3 Discussion

Ablation Study. Table 5.4 shows results on varying the current choices: • 1) For GNNRank-N methods: forcing the loss to be the sum $\mathcal{L}_{\text{upset, simple}} + \mathcal{L}_{\text{upset, margin}}$,

Table 5.3: Performance on Kendall Tau based on the lowest $\mathcal{L}_{\text{upset, naive}}$ on ERO models, averaged over 10 runs with one standard deviation. "avg" for time series first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best is marked in **bold red** while the second best is in underline blue . As MVR does not generate results after one week, we leave it out here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	GNNRank-N	GNNRank-P
ERO(p=0.05, style=uniform,η=0.1)	0.75±0.00	0.04±0.00	0.03±0.00	0.70±0.01	0.77±0.00	0.56±0.00	0.58±0.00	0.01±0.05	0.74±0.00	0.77±0.00	0.76±0.01	**0.79±0.01**
ERO(p=0.05, style=gamma,η=0.2)	0.61±0.00	0.01±0.00	-0.01±0.00	0.61±0.00	0.74±0.00	0.52±0.00	0.51±0.00	-0.01±0.01	0.45±0.00	0.64±0.00	0.52±0.01	**0.77±0.00**
ERO(p=0.05, style=uniform,η=0.5)	0.61±0.00	0.05±0.00	0.01±0.00	0.59±0.01	0.68±0.00	0.44±0.00	0.41±0.00	0.05±0.00	0.60±0.00	0.62±0.00	0.62±0.00	**0.70±0.02**
ERO(p=0.05, style=gamma,η=0.4)	0.51±0.00	0.08±0.00	-0.00±0.00	0.52±0.00	0.65±0.00	0.43±0.00	0.43±0.00	0.09±0.01	0.23±0.00	0.44±0.00	0.38±0.08	**0.66±0.01**
ERO(p=1, style=uniform,η=0.5)	0.85±0.00	0.07±0.00	**0.92±0.00**	0.81±0.03	0.91±0.00	0.80±0.00	0.73±0.00	0.24±0.00	0.89±0.00	0.87±0.00	0.90±0.01	**0.92±0.00**
ERO(p=1, style=gamma,η=0.6)	0.72±0.00	0.09±0.00	**0.89±0.00**	0.67±0.01	0.88±0.00	0.65±0.00	0.64±0.00	0.05±0.02	0.74±0.00	0.73±0.00	0.77±0.00	**0.89±0.00**

Table 5.4: $\mathcal{L}_{\text{upset, simple}}$ comparison for different variants on real-world data, averaged over 10 runs, and plus/minus one standard deviation. The best for each group (GNNRank-N or GNNRank-P) is marked in **bold red** while the second best is in underline blue .

Methods	GNNRank-N			GNNRank-P						
Data/Variant	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	no pretrain	$\{\alpha_\gamma\}_{\gamma=1}^{\Gamma}$not trainable	$\Gamma = 3$	$\Gamma = 7$
Animal	0.43±0.06	0.59±0.08	**0.41±0.09**	**0.25±0.00**	**0.25±0.00**	**0.25±0.01**	**0.25±0.00**	**0.25±0.00**	**0.25±0.00**	**0.25±0.00**
Faculty: Business	0.40±0.02	0.49±0.16	**0.38±0.01**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**
Faculty: CS	0.35±0.01	0.36±0.01	**0.33±0.03**	**0.32±0.00**	**0.32±0.00**	**0.32±0.00**	0.33±0.00	**0.32±0.00**	**0.32±0.00**	**0.32±0.00**
Faculty: History	**0.28±0.01**	0.31±0.01	**0.28±0.01**	**0.30±0.01**	**0.30±0.01**	**0.30±0.02**	**0.30±0.01**	**0.30±0.01**	**0.30±0.01**	**0.30±0.01**
Football (avg)	**0.82±0.01**	0.84±0.03	**0.82±0.05**	0.78±0.02	0.78±0.01	0.79±0.01	0.79±0.02	0.79±0.02	**0.77±0.01**	0.78±0.02
Football finer (avg)	**0.90±0.01**	0.97±0.06	0.91±0.07	**0.82±0.01**	0.84±0.01	**0.82±0.01**	0.84±0.02	**0.82±0.00**	**0.82±0.01**	**0.82±0.01**

$\mathcal{L}_{\text{upset, simple}}$ only, or $\mathcal{L}_{\text{upset, margin}}$, respectively; • 2) for GNNRank-P methods: in addition to the variants for GNNRank-N methods, removing pretraining, fixing all α_γ in Algo. 1, and changing the number of Fiedler proximal steps from default 5 to 3 or 7.

It is shown that for GNNRank-N methods, using $\mathcal{L}_{\text{upset, margin}}$ usually harms performance. For GNNRank-P methods, using $\mathcal{L}_{\text{upset, margin}}$ in addition to $\mathcal{L}_{\text{upset, ratio}}$ usually boosts performance. Pretraining generally leads to better performance, so does making $\{\alpha_\gamma\}_{\gamma=1}^{\Gamma}$ trainable. The number of proximal gradient steps does not need to be large probably due to fast convergence, so we use 5 throughout. Full comparison tables are in Appendix D.5.4 including results on $\mathcal{L}_{\text{upset, naive}}$, with similar conclusions.

In addition, note that essentially we could use any neural network method to obtain the node embeddings, yet digraph GNNs are natural to be employed given the input data structure. To validate the benefit of using a digraph GNN, we adopt a two-layer Multilayer perceptron to obtain node embeddings, but obtained on average 2% worse $\mathcal{L}_{upset,simple}$ for both the non-proximal and proximal variants across all real-world data sets, respectively.

Inductive Learning. We observe that our proximal methods, if trained only once and then applied to similar data sets, still perform comparably to multiply trained analogs. This can save training time and validates the inductive learning ability for our framework. To this end, Appendix D.5.5 shows results on the performance of the "IB proximal baseline" variant, trained with "emb baseline" on the *Basketball finer* data set. On average, directly applying gives $\mathcal{L}_{\text{upset, simple}} = 0.75 \pm 0.02$ and $\mathcal{L}_{\text{upset, naive}} = 0.19 \pm 0.01$ while training specifically for the season gives $\mathcal{L}_{\text{upset, simple}} = 0.74 \pm 0.00$ and $\mathcal{L}_{\text{upset, naive}} = 0.19 \pm 0.00$.

Variants and Hyperparameters. The results in Tables 5.2 and 5.3 are selected within either non-proximal or proximal categories, depending on whether they have proximal gradient steps within the architecture. They show the lowest reported evaluation metric (except for Kendall tau, when we select variants based on the lowest $\mathcal{L}_{\text{upset, naive}}$) for all variants within the group. Appendix D.7 gives the variant selected based on minimizing either $\mathcal{L}_{\text{upset, simple}}$ or $\mathcal{L}_{\text{upset, ratio}}$ for non-proximal and proximal groups, respectively. We find that each variant has its scenarios where it shows competitive or even outstanding performance; that "dist" seems to outperform "innerproduct" within non-proximal methods; and that "proximal baseline" is usually the best among proximal methods, with SyncRank output as initial guess, pretrained with a SerialRank similarity matrix. This shows that our proximal method, initialized with a good baseline, e.g. Sync-Rank, and pretrained with information from SerialRank, can boost the corresponding baseline method by using a learnable similarity matrix.

Boosting Baselines. Appendix D.6 shows improvements on $\mathcal{L}_{upset,simple}$ and $\mathcal{L}_{upset,naive}$ by "proximal baseline" when setting a certain baseline as $\mathbf{r}'$. Across all data sets, "proximal baseline" improves the most (by 1.02 and 0.24, respectively) with SyncRank as initial guess, while the average improvement for SpringRank, SerialRank, BTL, Eig.Cent., PageRank and SVD_NRS are 0.07, 0.82, 0.22, 0.19, 0.21, and 0.12, respectively, for $\mathcal{L}_{upset,simple}$, and for 0.00, 0.18, 0.04, 0.03, 0.03, and 0.01, respectively, for $\mathcal{L}_{upset,naive}$.

5.5 Conclusion and Outlook

We have proposed a general framework based on directed graph neural networks to recover global rankings from pairwise comparisons. Future directions include learning a more powerful model to work for different input digraphs, minimizing upsets under some constraints, training with some supervision of ground-truth rankings, and exploring the interplay with low-rank matrix completion. Incorporating side information, in the form of node level covariates, and comparing to the, currently rather limited, existing literature on ranking with covariates, is another interesting direction.

Statement of Authorship for joint/multi-authored papers for PGR thesis

Title of Paper:	*GNNRank: Learning Global Rankings from Pairwise Comparisons via Directed Graph Neural Networks*
Publication Status	▪ Published
Publication Details	*He, Yixuan, Quan Gan, David Wipf, Gesine D. Reinert, Junchi Yan, and Mihai Cucuringu. " GNNRank: Learning Global Rankings from Pairwise Comparisons via Directed Graph Neural Networks." In international conference on machine learning, pp. 8581-8612. PMLR, 2022.*

Student Confirmation

Student name:	Yixuan He
Contribution to the paper	*This is my work, the topic of which was raised by my supervisor Prof. Mihai Cucuringu. The project was continued during a research internship at Amazon (mentor: David Wipf) and a research visit to Shanghai Jiao Tong University (SJTU) invited by Prof. Junchi Yan. The idea of unfolding was from discussions between my Amazon colleague Quan Gan and me. The theoretical results were from me and proofread by my supervisor Prof. Gesine Reinert. I conducted the experiments and wrote the original manuscript.*
Signature: Yixuan He 何奕萱	**Date: 04/27/2024**

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Gesine Reinert	
Supervisor comments	The description is fine.
Signature: Gesine Reinert	**Date:** 29 April 2024

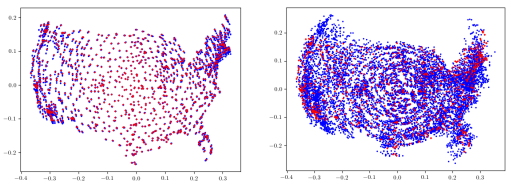
6

Robust Angular Synchronization Using Directed Graph Neural Networks

6.1 Introduction

The group synchronization problem has received considerable attention in recent years, as a key building block of many computational problems. Group synchronization aims to estimate a collection of group elements, given a small subset of potentially noisy measurements of their pairwise ratios $\Upsilon_{i,j} = g_i\, g_j^{-1}$. Some applications are • over the group SO(3) of 3D rotations: rotation-averaging in 3D computer vision [192, 193] and the molecule problem in structural biology [194]; • over the group $\mathbb{Z}_4$ of the integers $\{0, 1, 2, 3\}$ with addition mod 4 as the group operation: solving jigsaw puzzles [195]; • over the group $\mathbb{Z}_n$, resp., SO(2): recovering a global ranking from pairwise comparisons [30, 176], and, • over the Euclidean group of rigid motions $\mathrm{Euc}(2) = \mathbb{Z}_2 \times \mathrm{SO}(2) \times \mathbb{R}^2$: sensor network localization [21].

An important special case is *angular synchronization*, also referred to as *phase synchronization*, which can be viewed as group synchronization over SO(2). The angular synchronization problem aims at obtaining an accurate estimation (up



(a) Low noise. **(b)** High noise.

Figure 6.1: Sensor network localization map.

to a constant additive phase) for a set of unknown angles $\theta_1, \ldots, \theta_n \in [0, 2\pi)$ from m noisy measurements of their pairwise offsets $\theta_i - \theta_j \bmod 2\pi$. This problem has a wide range of applications, such as distributed clock synchronization over wireless networks [16], image reconstruction from pairwise intensity differences [17, 18], phase retrieval [19, 20], and sensor network localization (SNL) [21]. In engineering, the SNL problem seeks to reconstruct the 2D coordinates of a cloud of points from a sparse set of pairwise noisy Euclidean distances; in typical divide-and-conquer approaches that aid with scalability, one first computes a local embedding of nearby points (denoted as *patches*) and is left with the task of stitching the patches together in a globally consistent embedding [21]. Fig. 6.1 is an example of SNL on the U.S. map, where our method recovers city locations (in blue) and aims to match ground-truth locations (in red). Most works in the SNL literature that focus on the methodology development consider only purely synthetic data sets in their experiments; here we consider a real-world data set (actual 2D layout with different levels of densities of cities across the U.S. map), and add synthetic noise to perturb the local patch embeddings for testing the robustness to noise of the angular synchronization component.

An extension of angular synchronization to the heterogeneous setting is *k-synchronization*, introduced in [196], and motivated by real-world graph realization problems (GRP) and ranking. GRP aims to recover coordinates of a cloud of points in $\mathbb{R}^d$, from a sparse subset (edges of a graph) of noisy pairwise Euclidean distances (the case $d = 2$ is the above SNL problem). The motivation for *k-synchronization* arises in structural biology, where the distance measurements between pairs of atoms may correspond to k different configurations of the molecule, in the case of molecules with multiple conformations. In ranking applications, the $k = 2$ sets of disjoint pairwise measurements may correspond to two different judges, whose latent rankings we aim to recover.

A key limitation of existing methods for angular synchronization is their poor performance in the presence of considerable noise. High noise levels are not unusual; measurements in SO(3) can have large outliers in certain biological settings (cryo-EM

and NMR spectroscopy), see for example [194]. Therefore, we need new methods to push the boundary of signal recovery when there is a high level of noise. While neural networks (NNs), in principle, could be trained to address high noise regimes, the angular synchronization problem is not directly amenable to a standard NN architecture due to the directed graph (digraph) structure of the underlying data measurement process and the underlying group structure; hence the need for a customized graph neural network (GNN) architecture and loss function for this task. Here we propose a GNN method called GNNSync for angular synchronization, with a novel cycle loss function, which downweights noisy observations, and explicitly enforces cycle consistency as a quality measure. GNNSync's novelty does not lie in simply applying a data-driven NN to this task, but rather in proposing a framework for handling the pairwise comparisons encoded in a digraph, accounting for the underlying SO(2) group structure, and designing a loss function for increased robustness to noise and outliers, with theoretical support.

Our main contributions are summarized as follows.

• We demonstrate how the angular synchronization problem can be recast as a theoretically-grounded directed graph learning task by first incorporating the inductive biases of classical estimators within the design of a more robust GNN architecture, called GNNSync, and then pairing with a novel training loss that exploits cycle consistency to help infer the unknown angles.

• We perform extensive experiments comparing GNNSync with existing state-of-the-art algorithms from the angular synchronization and k-synchronization literature, across a variety of synthetic outlier models at various density and noise levels, and on a real-world application. GNNSync attains leading performance, especially in high noise regimes, validating its robustness to noise.

6.2 Related work

6.2.1 Angular synchronization

The seminal work of [197] introduced spectral and semidefinite programming (SDP) relaxations for angular synchronization. For the spectral relaxation, the estimated

angles are given by the eigenvector corresponding to the largest eigenvalue of a Hermitian matrix $\mathbf{H}$, whose entries are given by $\mathbf{H}_{i,j} = \exp(\iota \mathbf{A}_{i,j}) \mathbb{1}(\mathbf{A}_{i,j} \neq 0)$, where ι is the imaginary unit, and $\mathbf{A}_{i,j}$ is the observed potentially noisy offset $\theta_i - \theta_j \bmod 2\pi$. [197] also provided an SDP relaxation involving the same matrix $\mathbf{H}$, and empirically demonstrated that the spectral and SDP relaxations yield similar experimental results. A row normalization was introduced to $\mathbf{H}$ prior to the eigenvector computation by [21], which showed improved results. [194] generalized this approach to the 3D setting $\mathrm{Euc}(3) = \mathbb{Z}_2 \times \mathrm{SO}(3) \times \mathbb{R}^3$, and incorporated into the optimization pipeline the ability to operate in a semi-supervised setting, where certain group elements are known a-priori. [196] extended the angular synchronization problem to a heterogeneous setting, to the so-called *k-synchronization* problem, whose goal is to estimate k sets of angles simultaneously, given only the graph union of noisy pairwise offsets, which we also explore in our experiments. The key idea in their work is to estimate the k sets of angles from the top k eigenvectors of the angular embedding matrix $\mathbf{H}$.

[26] modeled the angular (phase) synchronization problem as a least-squares non-convex optimization problem, and proposed a modified version of the power method called the Generalized Power Method (GPM), which is straightforward to implement and free of parameter tuning. GPM often attains leading performance among baselines in our experiments, and the iterative steps in the GPM method motivated the design of the projected gradient steps in our GNNSync architecture. However, GPM is not directly applicable to k-synchronization with $k > 1$ while GNNSync is. For $k = 1$, GNNSync tends to perform significantly better than GPM at high noise levels. [198] studied the tightness of the maximum likelihood semidefinite relaxation for angular synchronization, where the maximum likelihood estimate is the solution to a nonbipartite Grothendieck problem over the complex numbers. A truncated least-squares approach was proposed by [199] that minimizes the discrepancy between the estimated angle differences and the observed differences under some constraints. [200] tackled the angular synchronization problem with a multi-frequency approach. [201] unified various group synchronization problems over subgroups of

the orthogonal group. [202] provided recovery guarantees for eigenvector relaxation and semidefinite convex relaxation methods for weighted angular synchronization. [203] applied a message-passing procedure based on cycle consistency information, to estimate the corruption levels of group ratios and consequently solve the synchronization problem, but the method is focused on the restrictive setting of adversarial or uniform corruption and sufficiently small noise. In addition, [203] requires post-processing based on the estimated corruption levels to obtain the group elements, while GNNSync is trained end-to-end. [204] utilized energy minimization ideas, with a variant converging linearly to the ground truth rotations.

6.2.2 Directed graph neural networks

Digraph node embeddings can be effectively learned via directed graph neural networks [28]. For learning such an embedding, [114] constructed a GNN using higher-order proximity. [6] built a complex Hermitian Laplacian matrix and proposed a spectral digraph GNN. [5] introduced imbalance objectives for digraph clustering. Our GNNSync framework can readily incorporate any existing digraph neural network.

6.2.3 Relationship with other group synchronization methods

Angular synchronization outputs can be used to obtain global rankings by using a one-dimensional ordering. To this end, recovering rankings of n objects from pairwise comparisons can be viewed as group synchronization over $\mathbb{Z}_n$. To recover global rankings from pairwise comparisons, GNNRank [30] adopted an unfolding idea to add an inductive bias from [25] to the NN architecture. Inspired by [30], we adapt their framework to borrow strength from solving a related problem. We adapt their “innerproduct" variant to k-synchronization, remove the 1D ordering at the end of the GNNRank framework, and rescale the estimated quantities to the range $[0, 2\pi)$. We also borrow strength from the projected gradient steps in GPM [26] and add projected gradient steps to our GNNSync architecture. Another

key novelty is that we devise novel objectives, which reflect the angular structure of the data, to serve as our training loss functions. The architectures are also very different: While in GNNRank the proximal gradient steps play a vital role from an unrolling perspective, and the whole architecture could be viewed as an unrolling of the SerialRank algorithm, here, although we borrow strength from the GPM method, the whole architecture is different from merely unrolling GPM. Furthermore, the baselines serve as initial guesses for the "proximal baseline" variant in GNNRank, but serve as input node features in our approach.

Other methods have been introduced for group synchronization, but mostly in the context of SO(3). [205] proposed an efficient algorithm for synchronization over SO(3) under high levels of corruption and noise. [206] provided a novel quadratic programming formulation for estimating the corruption levels, but again its focus is on SO(3). Unrolled algorithms (which are NNs) were introduced for SO(3) in [193]. While an adaptation to SO(2) may be possible in principle, as its objective functions are based on the level of agreement between the estimated angles and ground-truth, its experiments require ground-truth during training, usually not available in practice. In contrast, our GNNSync framework can be trained without any known angles.

6.3 Problem definition

The *angular synchronization* problem aims at obtaining an accurate estimation (up to a constant additive phase) for a set of n unknown angles $\theta_1, \ldots, \theta_n \in [0, 2\pi)$ from m noisy measurements of their offsets $\theta_i - \theta_j \bmod 2\pi$, for $i, j \in \{1, \ldots, n\}$. We encode the noisy measurements in a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each of the n elements of the node set $\mathcal{V}$ has as attribute an angle $\theta_i \in [0, 2\pi)$. The edge set $\mathcal{E}$ represents pairwise measurements of the angular offsets $(\theta_i - \theta_j) \bmod 2\pi$. The weighted directed graph has a corresponding adjacency matrix $\mathbf{A}$ with $\mathbf{A}_{i,j} = (\theta_i - \theta_j) \bmod 2\pi \geq 0$. Estimating the unknown angles from noisy offsets amounts to assigning an estimate $r_i \in [0, 2\pi)$ to each node $i \in \mathcal{V}$. For computational complexity considerations, we randomly keep one of $\mathbf{A}_{i,j}$ and $\mathbf{A}_{j,i}$ as observed quantity and set the other of these

to zero. Thus, at most one of $\mathbf{A}_{i,j}$ and $\mathbf{A}_{j,i}$ can be nonzero by construction; the other original entry can be inferred from $\mathbf{A}_{i,j} + \mathbf{A}_{j,i} = 0 \bmod 2\pi$.

An extension of the above problem to the heterogeneous setting is the *k-synchronization* problem, which is defined as follows. We are given only the graph union of k digraphs $\mathcal{G}_1, \ldots, \mathcal{G}_k$, with the same node set and disjoint edge sets, which encode noisy measurements of k sets $(\theta_{i,l} - \theta_{j,l}) \bmod 2\pi$, for $l \in \{1, \ldots, k\}, i, j \in \{1, \ldots, n\}$, of angle differences modulo 2π. Its adjacency matrix is denoted by $\mathbf{A}$. The problem is to estimate these k sets of n unknown angles $\theta_{i,l} \in [0, 2\pi), \forall l \in \{1, \ldots, k\}, i \in \{1, \ldots, n\}$, simultaneously. Note that we are given only $\mathcal{G} = \mathcal{G}_1 \cup \cdots \cup \mathcal{G}_k$ and the value of k, and each edge in $\mathcal{G}$ belongs to exactly one of $\mathcal{G}_1, \ldots, \mathcal{G}_k$. To unify notations, we view the normal angular synchronization problem as a special case of the more general k-synchronization problem where $k = 1$.

6.4 Loss and evaluation

6.4.1 Loss and evaluation for angular synchronization

For a vector $\mathbf{r} = [r_1, \ldots, r_n]^\top$ with estimated angles as entries, we define $\mathbf{T} = [(\mathbf{r}\mathbf{1}^\top - \mathbf{1}\mathbf{r}^\top) \bmod 2\pi] \in \mathbb{R}^{n \times n}$. Then $\mathbf{T}_{i,j} = (r_i - r_j) \bmod 2\pi$ estimates $\mathbf{A}_{i,j}$. We only compare $\mathbf{T}$ with $\mathbf{A}$ at locations where $\mathbf{A}$ has nonzero entries. We introduce the residual matrix $\mathbf{M}$ with entries

$$\mathbf{M}_{i,j} = \min\left((\mathbf{T}_{i,j} - \mathbf{A}_{i,j}) \bmod 2\pi, (\mathbf{A}_{i,j} - \mathbf{T}_{i,j}) \bmod 2\pi\right)$$

if $\mathbf{A}_{i,j} \neq 0$, and $\mathbf{M}_{i,j} = 0$ if $\mathbf{A}_{i,j} = 0$. Then our upset loss is defined as

$$\mathcal{L}_{\text{upset}} = \|\mathbf{M}\|_F / t, \tag{6.1}$$

where the subscript F means Frobenius norm, and t is the number of nonzero elements in $\mathbf{A}$. Despite the non-differentiablility of the loss function, using the concept of a limiting subdifferential from [207] we can give the following theoretical guarantee on the minimization of eq. (6.1); its proof is in Appendix (App.) E.1.1, where also the case of general k is discussed.

Proposition 2. *Every local minimum of eq.* (6.1) *is a directional stationary point of eq.* (6.1).

For *evaluation*, we employ a Mean Square Error (MSE) function with angle corrections, considered in [208]. As the offset measurements are unchanged if we shift all angles by a constant, denoting the ground-truth angle vector as $\mathbf{R}$, this evaluation function can be written as

$$\mathcal{D}_{\mathrm{MSE}}(\mathbf{r},\mathbf{R}) = \min_{\theta_0\in[0,2\pi)} \sum_{i=1}^{n} [\min(\delta_i \bmod 2\pi, (-\delta_i) \bmod 2\pi)]^2, \tag{6.2}$$

where $\delta_i = r_i + \theta_0 - \theta_i,\ \forall i = 1,\dots,n$. Additional implementation details are provided in App. E.3.4.

6.4.2 Cycle consistency relation

For noiseless observations, every cycle in the angular synchronization problem ($k = 1$) or every cycle whose edges correspond to the same offset graph $\mathcal{G}_l$ ($k > 1$) satisfy the *cycle consistency* relation that the angle sum mod 2π is 0. For 3-cycles (i, j, q), such that $\mathbf{A}_{i,j} \cdot \mathbf{A}_{j,q} \cdot \mathbf{A}_{q,i} > 0$, this leads to

$$(\mathbf{A}_{i,j} + \mathbf{A}_{j,q} + \mathbf{A}_{q,i}) \bmod 2\pi = (\theta_i - \theta_j + \theta_j - \theta_q + \theta_q - \theta_i) \bmod 2\pi = 0,$$

as $(a + b \bmod m) = \{(a \bmod m) + (b \bmod m) \bmod m\}$. Hence we obtain the 3-cycle condition

$$(\mathbf{A}_{i,j} + \mathbf{A}_{j,q} + \mathbf{A}_{q,i}) \bmod 2\pi = 0, \forall (i,j,q) \text{ such that } \mathbf{A}_{i,j} \cdot \mathbf{A}_{j,q} \cdot \mathbf{A}_{q,i} > 0. \tag{6.3}$$

With $\mathbb{T} = \{(i, j, q) : \mathbf{A}_{i,j} \cdot \mathbf{A}_{j,q} \cdot \mathbf{A}_{q,i} > 0\}$, we define the *cycle inconsistency level* $\frac{1}{|\mathbb{T}|} \sum_{(i,j,q)\in\mathbb{T}} [(\mathbf{A}_{i,j} + \mathbf{A}_{j,q} + \mathbf{A}_{q,i}) \bmod 2\pi]$. We devise a loss function to minimize the cycle inconsistency level with reweighted edges.

6.4.3 Loss and evaluation for general k-synchronization

The upset loss for general k is defined similarly as in Sec. 6.4.1. Recall that the observed graph $\mathcal{G}$ has adjacency matrix $\mathbf{A}$. Given k groups of estimated angles $\{r_{1,l}, \dots, r_{n,l}\}$, $l = 1, \dots, k$, we define the matrix $\mathbf{T}^{(l)}$ with entries $\mathbf{T}^{(l)}_{i,j} = (r_{i,l} - r_{j,l}) \bmod 2\pi$, for $i, j \in \{1, \dots, n\}, l \in \{1, \dots, k\}$. We define $\mathbf{M}^{(l)}$ by $\mathbf{M}^{(l)}_{i,j} = \min((\mathbf{T}^{(l)}_{i,j} - \mathbf{A}_{i,j}) \bmod 2\pi, (\mathbf{A}_{i,j} - \mathbf{T}^{(l)}_{i,j}) \bmod 2\pi)$ if $\mathbf{A}_{i,j} \neq 0$, and $\mathbf{M}^{(l)}_{i,j} = 0$

if $\mathbf{A}_{i,j} = 0$. Define $\mathbf{M}$ by $\mathbf{M}_{i,j} = \min_{l\in\{1,\dots,k\}} \mathbf{M}^{(l)}_{i,j}$. The upset loss is as in eq. (6.1), $\mathcal{L}_{\text{upset}} = \|\mathbf{M}\|_F / t$.

In addition to $\mathcal{L}_{\text{upset}}$, we introduce another option as a loss function based on the cycle consistency relation from Sec. 6.4.2, which adds a regularization that helps in guiding the learning process for certain challenging scenarios (e.g., with sparser $\mathcal{G}$ or larger k). Since measurements are typically noisy, we first estimate the corruption level by entries in $\mathbf{M}$, and use them to construct a confidence matrix $\tilde{\mathbf{C}}$ for edges in $\mathcal{G}$. We define the unnormalized confidence matrix $\mathbf{C}$ by $\mathbf{C}_{i,j} = \frac{1}{1+\mathbf{M}_{i,j}} \mathbb{1}(\mathbf{A}_{i,j} \neq 0)$, then normalize the entries by $\tilde{\mathbf{C}}_{i,j} = \mathbf{C}_{i,j} \frac{\sum_{u,v} \mathbf{A}_{u,v}}{\sum_{u,v} \mathbf{A}_{u,v} \cdot \mathbf{C}_{u,v}}$. The normalization is chosen such that $\sum_{i,j} \mathbf{A}_{i,j} \tilde{\mathbf{C}}_{i,j} = \sum_{u,v} \mathbf{A}_{u,v}$. Keeping the sum of edge weights constant is carried out in order to avoid reducing the cycle inconsistency level by only rescaling edge weights but not their relative magnitudes. Based on the confidence matrix $\tilde{\mathbf{C}}$, we reweigh edges in $\mathcal{G}$ to obtain an updated input graph, whose adjacency matrix is the Hadamard product $\mathbf{A} \odot \tilde{\mathbf{C}}$. This graph attaches larger weights to edges $\mathbf{A}_{i,j}$ for which $\mathbf{T}^{(l)}_{i,j}$ is a good estimate when the edge (i, j) belongs to graph $\mathcal{G}_l$. As the graph assignment of an edge (i, j) is not known during training, we estimate it by

$$g(i,j) = \arg \min_{l\in\{1,\dots,k\}} \mathbf{M}^{(l)}_{i,j}, \text{ and set } g(j,i) = g(i,j), \tag{6.4}$$

thus obtaining our estimated graphs $\tilde{\mathcal{G}}_1, \dots, \tilde{\mathcal{G}}_k$, which are also edge disjoint. Next, aiming to minimize 3-cycle inconsistency of the updated input graph given our graph assignment estimates, we introduce a loss function denoted as the *cycle inconsistency loss* $\mathcal{L}_{\text{cycle}}$; for simplicity, we only focus on 3-cycles (triangles). We interpret the matrix $\tilde{\mathbf{A}} = (\mathbf{A}\odot\tilde{\mathbf{C}} - (\mathbf{A}\odot\tilde{\mathbf{C}})^\top) \bmod 2\pi$ as the adjacency matrix of another weighted directed graph $\tilde{\mathcal{G}}$. The entry $\tilde{A}_{i,j}$ of the new adjacency matrix approximates angular differences of a reweighted graph, with noisy observations downweighted. Note that we only reweigh the adjacency matrix in the cycle loss definition, but do not update the input graph. The underlying idea is that this updated denoised graph may display higher cycle consistency than the original graph. From our graph assignment estimates, we obtain estimated adjacency matrices $\tilde{\mathbf{A}}^{(l)}$ for $l \in \{1, \dots, k\}$, where $\tilde{\mathbf{A}}^{(l)}_{i,j} = \mathbb{1}(g(i,j) = l)\tilde{\mathbf{A}}_{i,j}$. Let $\mathbb{T}^{(l)} = \{(i,j,q) : \tilde{A}^{(l)}_{i,j} \cdot \tilde{A}^{(l)}_{j,q} \cdot \tilde{A}^{(l)}_{q,i} > 0\}$ denote the set of all triangles in $\tilde{\mathcal{G}}_l$, and set $S^{(l)}_{i,j,q} = \tilde{\mathbf{A}}^{(l)}_{i,j} + \tilde{\mathbf{A}}^{(l)}_{j,q} + \tilde{\mathbf{A}}^{(l)}_{q,i}$ for $(i,j,q) \in \mathbb{T}^{(l)}$. We define

$$\mathcal{L}_{\text{cycle}}^{(l)} = \frac{1}{|\mathbb{T}^{(l)}|} \sum_{(i,j,q)\in\mathbb{T}^{(l)}} \min(S_{i,j,q}^{(l)} \bmod 2\pi, (-S_{i,j,q}^{(l)}) \bmod 2\pi) \tag{6.5}$$

and set $\mathcal{L}_{\text{cycle}} = \frac{1}{k}\sum_{l=1}^{k} \mathcal{L}_{\text{cycle}}^{(l)}$. The default training loss for $k \geq 2$ is $\mathcal{L}_{\text{cycle}}$ or $\mathcal{L}_{\text{upset}}$ alone; in the experiment section, we also report the performance of a variant based on $\mathcal{L}_{\text{upset}} + \mathcal{L}_{\text{cycle}}$.

For evaluation, we compute $\mathcal{D}_{\text{MSE}}$ with eq. (6.2), for each of the k sets of angles, and consider the average. As the ordering of the k sets can be arbitrary, we consider all permutations of $\{1, \ldots, k\}$, denoted by $perm(k)$. Denoting the ground-truth angle matrix as $\mathbf{R}$, whose (i, l) entry is the ground-truth angle $\theta_{i,l}$, and the l-th entry of the permutation pe by $pe(l)$, the final MSE value is

$$\mathcal{D}_{\text{MSE}}(\mathbf{r}, \mathbf{R}) = \frac{1}{k} \min_{pe\in perm(k)} \sum_{l=1}^{k} \mathcal{D}_{\text{MSE}}(\mathbf{r}_{:,pe(l)}, \mathbf{R}_{:,l}). \tag{6.6}$$

Note that the MSE loss is not used during training as we do not have any ground-truth supervision; the MSE formulation in eq. (6.2) is only used for evaluation. The lack of ground-truth information in the presence of noise is precisely what renders this problem very difficult. If any partial ground-truth information is available, then this can be incorporated into the loss function.

6.5 GNNSync architecture

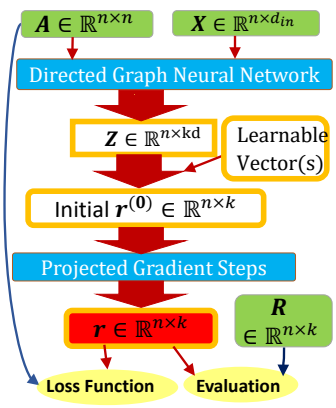


Figure 6.2: GNNSync overview: starting from an adjacency matrix $\mathbf{A}$ encoding (noisy) pairwise offsets and an input feature matrix $\mathbf{X}$, GNNSync first applies a directed GNN to learn node embeddings $\mathbf{Z}$. It then calculates the inner product with a learnable vector (or k learnable vectors for $k > 1$) to produce the initial estimated angles $r_{i,l}^{(0)} \in [0, 2\pi)$ for $l \in \{1, \ldots, k\}$, after rescaling. It then applies several projected gradient steps to the initial angle estimates to obtain the final angle estimates, $r_{i,l} \in [0, 2\pi)$. Let the ground-truth angle matrix be $\mathbf{R} \in \mathbb{R}^{n\times k}$. The loss function is applied to the output angle matrix $\mathbf{r}$, given $\mathbf{A}$, while the final evaluation is based on $\mathbf{R}$ and $\mathbf{r}$. Orange frames indicate trainable vectors/matrices, green squares fixed inputs, the red square the final estimated angles (outputs), and the yellow circles the loss function and evaluation.

6.5.1 Obtaining directed graph embeddings

For obtaining digraph embeddings, any digraph GNN that outputs node embeddings can be applied, e.g. DIMPA by [5], the inception block model by [114], and MagNet by [6]. Here we employ DIMPA; details are in E.1.2. Denoting the final node embedding matrix by $\mathbf{Z} \in \mathbb{R}^{n\times kd}$, the embedding vector $\mathbf{z}_i$ for a node i is $\mathbf{z}_i = (\mathbf{Z})_{(i,:)} \in \mathbb{R}^{kd}$, the i^{th} row of $\mathbf{Z}$.

6.5.2 Obtaining initial estimated angles

To obtain the initial estimated angles for the angular synchronization problem, we introduce a trainable vector $\mathbf{a}$ with dimension equal to the embedding dimension, then calculate the unnormalized estimated angles by the inner product of $\mathbf{z}_i$ with $\mathbf{a}$, plus a trainable bias b, followed by a sigmoid layer to force positive values, and finally rescale the angles to $[0, 2\pi)$; in short: $r_i^{(0)} = 2\pi\,\text{sigmoid}(\mathbf{z}_i \cdot \mathbf{a} + b)$.

For general k-synchronization, we apply independent $\mathbf{a}, b$ values to obtain k different groups of initial angle estimates based on different columns of the node embedding matrix $\mathbf{Z}$. In general, denote $\mathbf{Z}_{i,u:v}$ as the $(v-u+1)$-vector whose entries are from the i-th row and the u-th to v-th columns of the matrix $\mathbf{Z}$. With a trainable vector $\mathbf{a}^{(l)}$ for each $l \in \{1, \ldots, k\}$ with dimension equal to d, we obtain the unnormalized estimated angles by the inner product of $\mathbf{Z}_{i,(l-1)d+1:ld}$ with $\mathbf{a}^{(l)}$, plus a trainable bias b_l, followed by a sigmoid layer to force positive angle values, then rescale the angles to $[0, 2\pi)$; in short: $r_{i,l}^{(0)} = 2\pi\,\text{sigmoid}(\mathbf{z}_{i,(l-1)d+1:ld} \cdot \mathbf{a}^{(l)} + b_l)$.

6.5.3 Projected gradient steps for final angle estimates

Our final angle estimates are obtained after applying several (default: $\Gamma = 5$) projected gradient steps to the initial angle estimates. In brief, projected gradient descent for constrained optimization problems first takes a gradient step while ignoring the constraints, and then projects the result back onto the feasible set to incorporate the constraints. Here the projected gradient steps are inspired by [26]. We construct $\mathbf{H}$ by $\mathbf{H}_{i,j} = \exp(\iota\mathbf{A}_{i,j})\mathbb{1}(\mathbf{A}_{i,j} \neq 0)$, and update the estimated angles using Algo. 3, where $\mathbf{r}_{:,l}$ denotes the l-th column of $\mathbf{r}$. In Algo. 3 the gradient

step is on line 6, while the projection step on line 7 projects the updated matrix to elementwise angles. Fig. 6.2 shows the GNNSync framework.

Algorithm 3: Projected Gradient Steps

Input: Initial angle estimates $\mathbf{r}^{(0)} \in \mathbb{R}^{n\times k}$, Hermitian matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$, number of steps Γ (default: 5).
Parameter: (Initial) parameter set $\{\alpha_\gamma \geq 0\}_{\gamma=1}^{\Gamma}$ that could either be fixed or trainable (default: fixed value 1).
Output: Updated angle estimates $\mathbf{r} \in \mathbb{R}^{n\times k}$.

1: $l \leftarrow 1$;
2: **for** $l \leq k$ **do**
3: $\quad\gamma \leftarrow 1$; $\mathbf{y} \leftarrow \mathbf{r}_{:,l}^{(0)}$;
4: $\quad$**for** $\gamma \leq \Gamma$ **do**
5: $\quad\quad\tilde{\mathbf{y}} \leftarrow \exp(\iota\mathbf{y})$;
6: $\quad\quad\tilde{\mathbf{y}} \leftarrow \alpha_\gamma\tilde{\mathbf{y}} + \mathbf{H}\tilde{\mathbf{y}}$;
7: $\quad\quad\mathbf{y} \leftarrow \text{angle}(\tilde{\mathbf{y}})$ to obtain elementwise angles in radians from complex numbers;
8: $\quad\quad\gamma \leftarrow \gamma + 1$.
9: $\quad$**end for**
10: $\quad\mathbf{r}_{:,l} \leftarrow \mathbf{y}$.
11: $\quad l \leftarrow l + 1$.
12: **end for**
13: **return r**.

If graph assignments can be estimated effectively right after the GNN, one can replace $\mathbf{H}$ with $\mathbf{H}^{(l)}$ for each $l = 1, \ldots, k$ separately, where $\mathbf{H}^{(l)}_{i,j} = \exp(\iota\mathbf{A}^{(l)}_{i,j})\mathbb{1}(\mathbf{A}^{(l)}_{i,j} \neq 0)$, and $\mathbf{A}^{(l)}_{i,j} = \mathbb{1}(g(i,j) = l)\mathbf{A}_{i,j}$ is the estimated adjacency matrix for graph $\mathcal{G}_l$ using network assignments from $g(i,j)$ from eq. (6.4) applied to the initial angle estimates $\mathbf{r}^{(0)}$. Yet, separate $\mathbf{H}^{(l)}$'s may make the architecture sensitive to the accuracy of graph assignments after the GNN, and hence for robustness we simply choose a single $\mathbf{H}$. We also find in Sec. 6.6.4 that the use of $\mathbf{H}$ instead of separate $\mathbf{H}^{(l)}$'s is essential for satisfactory performance. Besides, it is possible to make Algo. 3 parameter-free by further fixing the $\{\alpha_\gamma\}$ values (default: $\alpha_\gamma = 1, \forall\gamma$); we find that using fixed $\{\alpha_\gamma\}$ does not strongly affect performance in our experiments. GNNSync executes the projected gradient descent steps at every training iteration as part of a unified end-to-end training process, but one could also use Algo. 3 to post-process predicted angles without putting the steps in the end-to-end framework. We find that putting Algo. 3 in our end-to-end training framework is usually helpful.

6.5.4 Robustness of GNNSync

Measurement noise that perturbs the edge offsets can significantly impact the performance of group synchronization algorithms. To this end, we demonstrate the robustness of GNNSync to such noise perturbations, with the following theoretical guarantee, proved and further discussed in App. E.1.2

Proposition 3. *For adjacency matrices* $\mathbf{A}, \hat{\mathbf{A}}$*, assume their row-normalized variants* $\mathbf{A}_s, \hat{\mathbf{A}}_s, \mathbf{A}_t, \hat{\mathbf{A}}_t$ *satisfy* $\left\|\mathbf{A}_s - \hat{\mathbf{A}}_s\right\|_F < \epsilon_s$ *and* $\left\|\mathbf{A}_t - \hat{\mathbf{A}}_t\right\|_F < \epsilon_t$*, where subscripts* s, t *denote source and target, resp. Assume further their input feature matrices* $\mathbf{X}, \hat{\mathbf{X}}$ *satisfy* $\left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F < \epsilon_f$*. Then their initial angles* $\mathbf{r}^{(0)}, \hat{\mathbf{r}}^{(0)}$ *from a trained GNNSync using DIMPA satisfy* $\left\|\mathbf{r}^{(0)} - \hat{\mathbf{r}}^{(0)}\right\|_F < B_s\epsilon_s + B_t\epsilon_s + B_f\epsilon_f$*, for values* B_s, B_t, B_f *that can be bounded by imposing constraints on model parameters and input.*

6.6 Experiments

Implementation details are in App. E.3 and extended results in App. E.4.

6.6.1 Data sets and protocol

Previous works in angular synchronization typically only consider synthetic data sets in their experiments, and those applying synchronization to real-world data do not typically publish the data sets. To bridge the gap between synthetic experiments and the real world, we construct synthetic data sets with both correlated and uncorrelated ground-truth rotation angles, using various measurement graphs and noise levels. In addition, we conduct sensor network localization on two data sets.

For synthetic data, we perform experiments on graphs with $n = 360$ nodes for different measurement graphs, with edge density parameter $p \in \{0.05, 0.1, 0.15\}$, noise level $\eta \in \{0, 0.1, \ldots, 0.9\}$ for $k = 1$, and $\eta \in \{0, 0.1, \ldots, 0.7\}$ for $k \in \{2, 3, 4\}$. The graph generation procedure is as follows (with further details in App. E.2.1): • 1) Generate k group(s) of ground-truth angles. One option is to generate each angle from the same Gamma distribution with shape 0.5 and scale 2π. We denote this option with subscript "1". As angles could be highly correlated in practical scenarios, we introduce a more realistic but challenging option "2", with multivariate normal ground-truth angles. The mean of the ground-truth angles is π, with covariance matrix for each $l \in \{1, \ldots, k\}$ defined by $\mathbf{w}\mathbf{w}^\top$, where entries in $\mathbf{w}$ are generated independently from a standard normal distribution. We explore two more options in the SI. We then apply mod 2π to all angles. • 2) Generate a noisy background adjacency matrix $\mathbf{A}_{\text{noise}} \in \mathbb{R}^{n\times n}$. • 3) Construct a complete adjacency matrix where

η portion of the entries are noisy and the rest represent true angular differences. • 4) Generate a measurement graph and sparsify the complete adjacency matrix by only keeping the edges in the measurement graph.

We construct 3 types of measurement graphs from NetworkX [209] and use the following notations, where the subscript $o \in \{1, 2, 3, 4\}$ is the option mentioned in step 1) above:

• Erdős-Rényi (ER) Outlier model: denoted by $\mathrm{ERO}_o(p, k, \eta)$, using as the measurement graph the ER model from NetworkX, where p is the edge density parameter for the ER measurement graph;

• Barabasi Albert (BA) Outlier model: denoted by $\mathrm{BAO}_o(p, k, \eta)$, where the measurement graph is a BA model with the number of edges to attach from a new node to existing nodes equal to $\lceil np/2 \rceil$, using the standard implementation from NetworkX [209]; and

• Random Geometric Graph (RGG) Outlier model: denoted by $\mathrm{RGGO}_o(p, k, \eta)$, with NetworkX parameter "distance threshold value (radius)" $2p$ for the RGG measurement graph. For $k = 1$, we omit the value k and subscript o in the notation, as the two options coincide in this special case.

For real-world data, we conduct sensor network localization on the U.S. map and the PACM point cloud data set [21] with a focus on the SO(2) component, as follows, with data processing details provided in App. E.2.2. • 1) Starting with the ground-truth locations of $n = 1097$ U.S. cities (resp., $n = 426$ points), we construct patches using each city (resp., point) as a central node and add its 50 nearest neighbors to the corresponding patch. • 2) For each patch, we add noise to each node's coordinates independently. • 3) We then rotate the patches using random rotation angles (ground-truth angles generated as in 1) for synthetic models). For each pair of patches that have at least 6 overlapping nodes, we apply Procrustes alignment [210] to estimate the rotation angle based on these overlapping nodes and add an edge to the observed measurement adjacency matrix. • 4) We perform angular synchronization to obtain the initial estimated angles and update the estimated angles by shifting by the average pairwise differences

between the estimated and ground-truth angles, to eliminate the degree of freedom of a global rotation. • 5) Finally, we apply the estimated rotations to the noisy patches and estimate node coordinates by averaging the estimated locations for each node from all patches that contain this node.

6.6.2 Baselines

In our numerical experiments for angular synchronization, we compare against **7 baselines**, where results are averaged over 10 runs: • Spectral Baseline (Spectral) by [197], • Row-Normalized Spectral Baseline (Spectral_RN) by [21], • Generalized Power Method (GPM) by [26], • TranSync by [199], • CEMP_GCW, • CEMP_MST by [203], and • Trimmed Averaging Synchronization (TAS) by [204].

For more general k-synchronization, we compare against two baselines from [196], which are based on the top k eigenvectors of the matrix $\mathbf{H}$ or its row-normalized version. We use names • Spectral and • Spectral_RN to denote them as before. To show that GNNSync (as well as the baselines) deviate from trivial or random solutions, we include an additional baseline denoted "Trivial" for each k, where all angles are predicted equal (with value 1, for simplicity).

6.6.3 Main experimental results

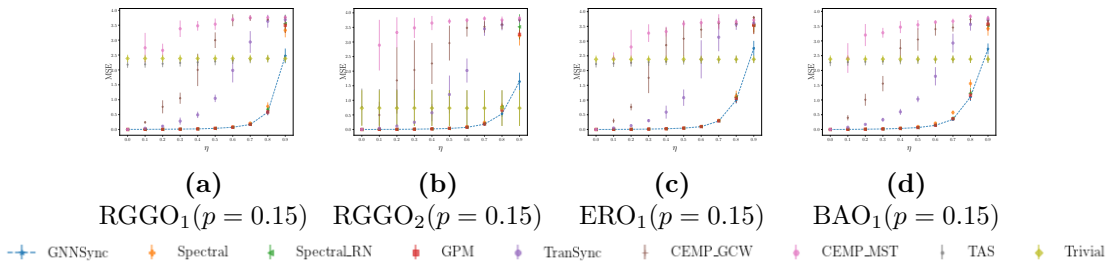


Figure 6.3: MSE performance on angular synchronization ($k = 1$). Error bars indicate one standard deviation. Dashed lines highlight GNNSync variants.

By default, we use the output angles of the baseline "Spectral_RN" as input features for GNNSync, and thus $d_{\text{in}} = k$. The main experimental results are shown in Fig. 6.3 for $k = 1$, and Fig. 6.4 for general $k \in \{2, 3, 4\}$, with additional results reported in App. E.4. For $k > 1$, we use "GNNSync-cycle", "GNNSync-upset" and

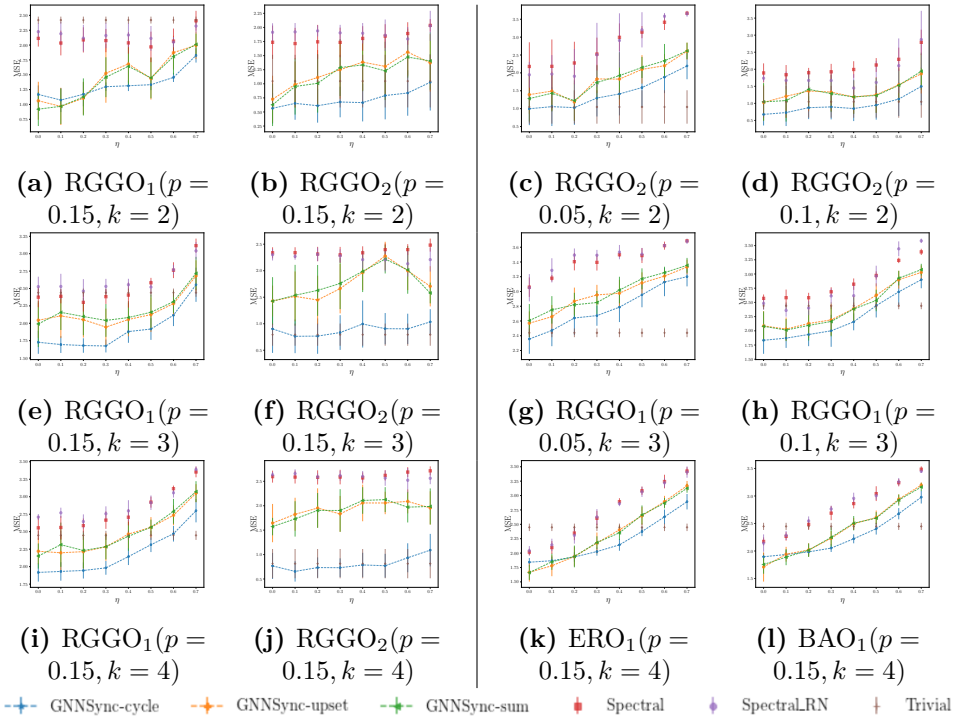


Figure 6.4: MSE performance on k-synchronization for $k \in \{2, 3, 4\}$. p is the network density and η is the noise level. Error bars indicate one standard deviation. Dashed lines highlight GNNSync variants.

"GNNSync-sum" to denote GNNSync variants when considering the training loss function $\mathcal{L}_{\text{cycle}}, \mathcal{L}_{\text{upset}}$, and $\mathcal{L}_{\text{upset}} + \mathcal{L}_{\text{cycle}}$, respectively.

From Fig. 6.3 (with additional figures in App. E.4 Fig. E.2–E.4), we conclude that GNNSync produces generally the best performance compared to baselines, in angular synchronization ($k = 1$). From Fig. 6.4 (see also App. E.4 Fig. E.5–E.13), we again conclude that GNNSync variants attain leading performance for $k > 1$. The first two columns of Fig. 6.4 compare the performance of the two options of ground-truth angles on RGGO models. In columns 3 and 4, we show the effect of varying density parameter p, and different synthetic models under various measurement graphs.

For $k > 1$, GNNSync-upset performs better than both baselines in most cases, with $\mathcal{L}_{\text{upset}}$ simple yet effective to train. GNNSync-cycle generally attains the best performance. As the problems become harder (with increasing η, decreasing p, increasing k, more complex measurement graph RGG), GNNSync-cycle outperforms both baselines and other GNNSync variants by a larger margin. The performance of GNNSync-sum lies between that of GNNSync-upset and GNNSync-cycle, but

is closer to that of GNNSync-upset, see App. E.4 for more discussions on linear combinations of the two losses. We conclude that while GNNSync-upset generally attains satisfactory performance, GNNSync-cycle is more robust to harder problems than other GNNSync variants and the baselines. Accounting for the performance of trivial guesses, we observe that GNNSync variants are more robust to noise, and attain satisfactory performance even when the competitive baselines are outperformed by trivial guesses. We highlight that there is a clear advantage of using cycle consistency in the pipeline, especially when the problem is harder, thus reflecting the angular nature of the problem. For 3-cycle consistency and the cycle loss $\mathcal{L}_{\text{cycle}}$, gradient descent in principle drives down the (non-negative) values S of the sum of three predicted angular differences. To minimize the S values, we encourage a reweighing process of the initial edge weights so that cycle consistency roughly holds. Unlike $\mathcal{L}_{\text{upset}}$ which explicitly encourages small $\mathbf{M}_{i,j}$ values for all edges, $\mathcal{L}_{\text{cycle}}$ only implicitly encourages small $\mathbf{M}_{i,j}$ values via the confidence matrix reweighing process for edges with relatively small noise. In an ideal case, we only have large $\mathbf{M}_{i,j}$ values on noisy edges. In this case, the reweighing process would downweight these noisy edges, which results in a smaller value of the cycle loss function. This is also the underlying reason why $\mathcal{L}_{\text{cycle}}$ is more robust to noise than $\mathcal{L}_{\text{upset}}$. For $k > 1$, we hence recommend using the more intricate $\mathcal{L}_{\text{cycle}}$ function as the training loss function, and we will focus on GNNSync-cycle in the ablation study.

From Fig. 6.1 (see also App. E.4 Tab. E.1–E.4, Fig. E.14–E.23), we observe that GNNSync is able to align patches and recover coordinates effectively, and is more robust to noise than baselines. GNNSync attains competitive MSE values and Average Normalized Error (ANE) results, where ANE (defined explicitly in App. E.4.1) measures the discrepancy between the predicted locations and the actual locations.

6.6.4 Ablation study and discussion

In this subsection, we justify several model choices for all k: • the use of the projected gradient steps; • an end-to-end framework instead of training first

without the projected gradient steps and then applying Algo. 3 as a post-processing procedure; • fixed instead of trainable $\{\alpha_\gamma\}$ values. For $k > 1$, we also justify the use of the $\mathbf{H}$ matrix in Algo. 3 instead of separate $\mathbf{H}^{(l)}$'s based on estimated graph assignments of the edges. To validate the ability of GNNSync to borrow strength from baselines, we set the input feature matrix $\mathbf{X}$ as a set of angles that is estimated by one of the baselines (or k sets of angles estimated by one of the baselines for $k > 1$) and report the performance.

Due to space considerations, results for the ablation study are reported in App. E.4. For $k = 1$, Fig. 22–24 report the MSE performance for different GNNSync variants. Improvements over all possible baselines when taking their output as input features for $k = 1$ are reported in Fig. 34–36. For $k > 1$, we report the results when using $\mathcal{L}_{\text{cycle}}$ as the training loss function in Fig. 25–33. We conclude that Algo. 3 is indeed helpful in guiding GNNSync to attain lower loss values (we omit loss results for space considerations) and better MSE performance, and that end-to-end training usually attains comparable or better performance than using Algo. 3 for post-processing, even when there is no trainable parameter in Algo. 3. Moreover, the baselines are still outperformed by GNNSync if we apply the same number of projected gradient steps as in GNNSyc as fine-tuning post-processing to the baselines, as illustrated in Fig. E.30 and E.31. We observe across all data sets, that GNNSync usually improves on existing baselines when employing their outputs as input features, and never performs significantly worse than the corresponding baseline; hence, GNNSync can be used to enhance existing methods. Further, setting $\{\alpha_\gamma\}$ values to be trainable does not seem to boost performance much, and hence we stick to fixed $\{\alpha_\gamma\}$ values. For $k > 1$, using separate $\mathbf{H}^{(l)}$'s instead of the whole $\mathbf{H}$ in Algo. 3 harms performance, which can be explained by the fact that learning graph assignments effectively via GNN outputs is challenging.

6.7 Conclusion and outlook

This chapter proposed a general NN framework for angular synchronization and a heterogeneous extension. As the current framework is limited to SO(2), we believe

that extending our GNN-based framework to the setting of other more general groups is an exciting research direction to pursue, and constitutes ongoing work (for instance, for doing synchronization over the full Euclidean group $\mathrm{Euc}(2) = \mathbb{Z}_2 \times \mathrm{SO}(2) \times \mathbb{R}^2$). We also plan to optimize the loss functions under constraints, train our framework with supervision of ground-truth angles (anchor information), and explore the interplay with low-rank matrix completion. Another interesting direction is to extend our SNL example to explore the graph realization problem, of recovering point clouds from a sparse noisy set of pairwise Euclidean distances.

Statement of Authorship for joint/multi-authored papers for PGR thesis

Title of Paper:	*Robust Angular Synchronization via Directed Graph Neural Networks*
Publication Status	▪ Published
Publication Details	*Yixuan He, Gesine Reinert, David Wipf, and Mihai Cucuringu. Robust Angular Synchronization via Directed Graph Neural Networks. In The Twelfth International Conference on Learning Representations, 2024.*

Student Confirmation

Student name:	Yixuan He
Contribution to the paper	*This is my work, the topic of which was raised by my supervisor Prof. Mihai Cucuringu. The project was continued during a research internship at Amazon (mentor: David Wipf). The architecture came from discussions among all authors. The idea of the loss functions came from discussions between my supervisors (Prof. Gesine Reinert and Prof. Mihai Cucuringu) and me. The theoretical results were from my supervisor Prof. Gesine Reinert and me. I conducted the experiments and wrote the original manuscript.*
Signature: Yixuan He 何奕萱	**Date: 04/27/2024**

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Gesine Reinert	
Supervisor comments	The description is fine.
Signature: Gesine Reinert	**Date:** 29 April 2024

7 Other Works on GNNs

Here is a collection of abstracts from papers that I contributed to. They are also related to my dissertation overall.

7.1 PyTorch Geometric Signed Directed: A Software Package on Graph Neural Networks for Signed and Directed Graphs

Networks are ubiquitous in many real-world applications (e.g., social networks encoding trust/distrust relationships, correlation networks arising from time series data). While many networks are signed or directed, or both, there is a lack of unified software packages on graph neural networks (GNNs) specially designed for signed and directed networks. In [28], we present PyTorch Geometric Signed Directed (PyGSD), a software package which fills this gap. Along the way, we also provide a brief review surveying typical tasks, loss functions and evaluation metrics in the analysis of signed and directed networks, discuss data used in related experiments, provide an overview of methods proposed, and evaluate the implemented methods with experiments. The deep learning framework consists of easy-to-use GNN models, synthetic and real-world data, as well as task-specific evaluation metrics and loss functions for signed and directed networks. As an extension library for PyG, our proposed

software is maintained with open-source releases, detailed documentation, continuous integration, unit tests and code coverage checks. Our code is publicly available at `https://github.com/SherylHYX/pytorch_geometric_signed_directed`.

This work is under review by a top journal in artificial intelligence.

7.2 MagNet: A Neural Network on Directed Graphs

The prevalence of graph-based data has spurred the rapid development of graph neural networks (GNNs) and related machine learning algorithms. Yet, despite the many data sets naturally modeled as directed graphs, including citation, website, and traffic networks, the vast majority of this research focuses on undirected graphs. In [6], we propose MagNet, a spectral GNN for directed graphs based on a complex Hermitian matrix known as the magnetic Laplacian. This matrix encodes undirected geometric structure in the magnitude of its entries and directional information in their phase. A "charge" parameter controls the extent to which we emphasize directional information. We apply our network to a variety of directed graph node classification and link prediction tasks showing that MagNet performs well on all tasks and that its performance exceeds all other methods on a majority of such tasks. The underlying principles of MagNet are such that it can be adapted to other spectral GNN architectures.

This work has been accepted by NeurIPS 2021.

7.3 PyTorch Geometric Temporal: Spatial-temporal Signal Processing with Neural Machine Learning Models

In [7], we present PyTorch Geometric Temporal, a deep learning framework combining state-of-the-art machine learning algorithms for neural spatiotemporal signal processing. The main goal of the library is to make temporal geometric deep learning available for researchers and machine learning practitioners in a

unified easy-to-use framework. PyTorch Geometric Temporal was created with foundations on existing libraries in the PyTorch ecosystem, streamlined neural network layer definitions, temporal snapshot generators for batching, and integrated benchmark data sets. These features are illustrated with a tutorial-like case study. Experiments demonstrate the predictive performance of the models implemented in the library on real world problems such as epidemiological forecasting, ridehail demand prediction and web-traffic management. Our sensitivity analysis of runtime shows that the framework can potentially operate on web-scale data sets with rich temporal features and spatial structure.

This work has been accepted by CIKM 2021 and won the best paper award.

7.4 Difformer: Scalable (graph) transformers induced by energy constrained diffusion

Real-world data generation often involves complex inter-dependencies among instances, violating the IID-data hypothesis of standard learning paradigms and posing a challenge for uncovering the geometric structures for learning desired instance representations. To this end, we introduce in [32] an energy constrained diffusion model which encodes a batch of instances from a data set into evolutionary states that progressively incorporate other instances' information by their interactions. The diffusion process is constrained by descent criteria w.r.t. a principled energy function that characterizes the global consistency of instance representations over latent structures. We provide rigorous theory that implies closed-form optimal estimates for the pairwise diffusion strength among arbitrary instance pairs, which gives rise to a new class of neural encoders, dubbed as DIFFormer (diffusion-based Transformers), with two instantiations: a simple version with linear complexity for prohibitive instance numbers, and an advanced version for learning complex structures. Experiments highlight the wide applicability of our model as a general-purpose encoder backbone with superior performance in various tasks, such as node classification on large graphs, semi-supervised image/text classification, and spatial-temporal dynamics prediction.

This work has been accepted as a spotlight paper (notable-top-25%) at ICLR 2023.

7.5 CEP3: Community Event Prediction with Neural Point Process on Graph

Many real-world applications can be formulated as event forecasting on Continuous Time Dynamic Graphs (CTDGs) where the occurrence of a timed event between two entities is represented as an edge along with its occurrence timestamp. However, many previous works handle the problem in compromised settings, either formulating it as a link prediction task on the graph given the event time, or a time prediction problem for which event will happen next. In [29], we propose a novel model combining Graph Neural Networks and Marked Temporal Point Process (MTPP) that jointly forecasts multiple link events and their timestamps on communities over a CTDG. Moreover, to scale our model to large graphs, we factorize the joint event prediction problem into three easier conditional probability modeling problems. To evaluate the effectiveness of our model and the rationale behind such a decomposition, we establish a set of benchmarks and evaluation metrics. The experimental results demonstrate the superiority of our model in terms of both accuracy and training efficiency. All the source codes and data sets are available in a GitHub repository.

This work has been accepted by LoG 2022.

7.6 Pyramid Graph Neural Network: a Graph Sampling and Filtering Approach for Multi-scale Disentangled Representations

Despite their success, existing GNNs are shown prone to extracting low-frequency information which may not always be pertinent to the task at hand. Though efforts have been made to cover wider frequency profiles for graph filtering, it remains open that how to disentangle the mixed multi-frequency information. Instead of directly performing message passing on the whole graph, in this work (paper not yet public),

we devise a framework called Pyramid Graph Neural Network (PyGNN) based on an assumption that has been well studied from the graph signal processing literature: bandlimited space of graph signals is able to recover the full signal from the subset of vertices, thus we are able to build the connection between the spectral domain and the vertex domain. Specifically, we develop an $\omega-$bandlimited method for cascading downsampling of the input graphs into a series of subgraphs that support signals at various frequency spectra. These subgraphs are then fed into their own respective GNN backbones for frequency profile-specific representation learning, whereby a localized lightweight message passing scheme is performed that gradually filters the respective frequency spectra in a pyramid manner. The spectra information is disentangled in the final node representation with expressiveness. Results on node classification on network data sets show its superiority over state-of-the-art methods.

This work has been accepted by KDD 2023.

7.7 Inferring Metabolic States from Single Cell Transcriptomic Data via Geometric Deep Learning

The ability to measure gene expression at single-cell resolution has elevated our understanding of how biological features emerge from complex and interdependent networks at molecular, cellular, and tissue scales. As technologies have evolved that complement scRNAseq measurements with things like single-cell proteomic, epigenomic, and genomic information, it becomes increasingly apparent how much biology exists as a product of multimodal regulation. Biological processes such as transcription, translation, and post-translational or epigenetic modification impose both energetic and specific molecular demands on a cell and are therefore implicitly constrained by the metabolic state of the cell. While metabolomics is crucial for defining a holistic model of any biological process, the chemical heterogeneity of the metabolome makes it particularly difficult to measure, and technologies capable of doing this at single-cell resolution are far behind other multiomics modalities. To address these challenges, in [34], we present GEFMAP (Gene Expression-based Flux

Mapping and Metabolic Pathway Prediction), a method based on geometric deep learning for predicting flux through reactions in a global metabolic network using transcriptomics data, which we ultimately apply to scRNAseq. GEFMAP leverages the natural graph structure of metabolic networks to learn both a biological objective for each cell and estimate a mass-balanced relative flux rate for each reaction in each cell using novel deep learning models.

This work has been accepted by RECOMB 2024.

7.8 Generalization Error of Graph Neural Networks in the Mean-field Regime

[35] provides a theoretical framework for assessing the generalization error of graph neural networks in the over-parameterized regime, where the number of parameters surpasses the quantity of data points. We explore two widely utilized types of graph neural networks: graph convolutional neural networks and message passing graph neural networks. Prior to this study, existing bounds on the generalization error in the over-parametrized regime were uninformative, limiting our understanding of over-parameterized network performance. Our novel approach involves deriving upper bounds within the mean-field regime for evaluating the generalization error of these graph neural networks. We establish upper bounds with a convergence rate of $O(1/n)$, where n is the number of graph samples. These upper bounds offer a theoretical assurance of the networks' performance on unseen data in the challenging over-parameterized regime and overall contribute to our understanding of their performance.

This work has been accepted by ICML 2024.

8 Conclusion and Future Work

8.1 Conclusion

Graph data are ubiquitous and have many applications. In this dissertation, the main focus is on signed networks and directed networks, with application to signal recovery from sparse noisy pairwise comparisons. In particular, we apply GNNs to signal recovery tasks on recovering global rankings from pairwise comparisons, and angular synchronization. With deep learning techniques and exploiting network features, GNNs are powerful in the analysis of networks. SSSNET and DIGRAC tackle the problem of node clustering, which aims at grouping together nodes that are similar. The definition of similarity is heavily dependent on the downstream task at hand. Our proposed methods achieve leading performance in signed and directed clustering tasks, respectively. They complement existing non-GNN methods by the possibility to include exogenous information such as node-level features and cluster labels. At the same time, they can borrow strength from such spectral methods by treating the embeddings from these methods as input node features. Compared with existing GNN methods, SSSNET is the first GNN method for signed clustering that is not based on social balance theory, while DIGRAC is the first GNN method for directed clustering that is purely driven by directionality instead of edge density, MSGNN is one of the first spectral GNN methods for signed directed networks, GNNRank

is the first neural network method for recovering global rankings from pairwise comparisons, and GNNSync is the first neural network for angular synchronization.

One central task during my doctoral studies is node clustering. [211] argues that the major obstacle in comparing the quality of different clustering algorithms is the difficulty in evaluating without considering the context, and we explicitly quantify several concepts of similarity while proposing our clustering approaches. For SSSNET, the clusters are defined such that node within clusters should be mostly positively connected while nodes across clusters should be mostly negatively connected. For DIGRAC, the clusters are detected based on network flows.

We have proposed a number of GNN architectures based on the task at hand. For SSSNET, we propose a novel signed mixed-path aggregation scheme and construct a novel GNN architecture for semi-supervised node clustering for signed graphs. For DIGRAC, we devise a directed mixed-path aggregation scheme. For MSGNN, we adopt the MagNet architecture but employ our novel Laplacian matrix. In GNNRank, we borrow strength from an existing ranking baseline called SerialRank [25] and unfold the Fiedler eigenvector computation steps. We also adopt ideas from GPM [26] to refine our initial angle guesses in GNNSync. In particular, for the GNN architecture, similar pipelines may be useful in different aspects. For example, the basic mixed-path aggregation scheme is used in both SSSNET and DIGRAC for node clustering. For both GNNRank and GNNSync, we first apply a directed graph neural network architecture to obtain the node embeddings, and then refine the embeddings by inductive biases based on the task at hand.

For a successful GNN model, the choice of the loss function is of great importance, as it could affect the performance of downstream tasks. Differentiability of the loss also requires careful consideration and design. Indeed, the design of the loss function is important for all GNNs; in particular, it represents the core contribution of our DIGRAC method and is essential for GNNRank and GNNSync.

Theoretical guarantees can guide users in better understanding the proposed methods, and assessing their strengths and weaknesses in various settings. In

DIGRAC, we derived hypothesis testing methods to determine clusters for consideration. In MSGNN, we proved that our proposed magnetic signed Laplacian matrix enjoys certain desirable properties. In GNNRank, we validate the convergence of our proximal gradient steps. In GNNSync, the behavior of the loss functions is discussed.

8.2 Future Work

There are several limitations in our current contributions that naturally lead to future directions. In addition, my doctoral research centers around methodological design, but theoretical aspects and applications are also very important. On the theoretical side, I plan to explore the role of graph convolution in graph-related tasks such as node classification. Besides, I plan to design algorithms driven by applications, for example, exploring chimpanzee activities. Specifically, I have the following future research directions.

8.2.1 Future Directions Based on the Limitations of Our Current Contributions

A current limitation in our work is scalability, and applying our GNNs to large-scale networks will be of interest. Adaptations using mini-batch or sampling could be helpful [73]. It might also be beneficial to borrow ideas from spectral sparsification, to reduce the number of edges while retaining structural properties of the graphs, such as preserving cuts [212].

GNNs are also not always preferred over simple spectral methods, as the training process often takes more time than simply solving matrix problems, especially when the networks are dense. In addition, compared to traditional approaches like spectral methods, GNNs relying solely on local information are not capable of computing several important graph properties such as shortest/longest cycle, diameter, or certain motifs, which limits their expressiveness. This is because such GNNs, including popular ones like GCN [71] and GAT [139], cannot distinguish some simple non-isomorphic graphs. [213] Our methods in this dissertation take into account information beyond simply edgewise unordered local information,

such as by considering signed paths (Chap. 2) and cycle information (Chap. 6). Based on our current contributions, another future direction is to incorporate more global information into the framework, ideally in the architecture itself rather than simply in the loss function.

Besides, the data representations using graphs are also not always ideal, as the real world is often more complicated. For example, a collaboration involving several authors cannot be completely represented via several separate edges but rather better modeled with a hyperedge. This opens up research topics concerning hypergraphs [8, 214, 215], with relationships more complex than just pairwise.

Furthermore, one aspect that is currently not well explored in our work is the stability of the methods. Evaluating the stability in terms of performance is not sufficient. A more detailed comparison of the outputs from the model under different initial parameters or different random seeds would be helpful to tell which part of the model prediction is the most sensitive. The changes in the output include, for example, the change in the meta-graph structure or cluster assignments (pairwise ARI values between predicted clustering results) for node clustering, the change in individual rankings for GNNRank, and the change of misclassified nodes/edges. It might also be beneficial to explore the parts of the output which are the most stable.

Moreover, in terms of benchmarks, it will be interesting to see whether the network's radius influences the performance significantly. To understand the limit of the proposed framework, it might be useful to check whether the improvement in the architecture (such as by improving the expressive power) can lead to the improvement in the performance, or whether it is the data quality that has caused imperfect performance. While designing synthetic data sets, we could explore the possibility of designing a method that could potentially match the data generation process and to obtain the possibly optimal solution.

Finally, as many networks are evolving, we aim to extend our work to temporal settings, as another future direction, building upon our current contributions such as [7] and [29].

8.2.2 When and How can Graph Convolution Boost Node Classification Accuracy

This project is inspired by [216] and the findings from our empirical analysis for previous GNN projects. From our previous experiments, clustering performance does not always increase as we increase the number of hops of neighbors to consider, see Section A.3.4 and Section B.3.4. Indeed, the use of graph convolution could help with node classification performance sometimes but not always, and it also increases computational costs. Here, we want to explore when graph convolution helps with node classification and how to use graph convolution effectively.

Here are some details. Suppose we have two classes: $\mathcal{C}_0$ and $\mathcal{C}_1$, with $n_0 = |\mathcal{C}_0|$ and $n_1 = |\mathcal{C}_1|$ nodes. Assume that there are associated node features which are encoded in the feature matrix $\mathbf{X}$. We further assume for each class, node features follow a Gaussian distribution: for the i-th row, $\mathbf{X}_i \sim \mathcal{N}(\mu, d_0 I)$ if $i \in \mathcal{C}_0$ and $\mathbf{X}_i \sim \mathcal{N}(\nu, d_1 I)$ if $i \in \mathcal{C}_1$. The adjacency matrix $\mathbf{A} = (a_{ij})$ is assumed to have independent Bernoulli entries, so that $a_{ij} \sim Ber(p_0)$ if $i, j \in \mathcal{C}_0$, $a_{ij} \sim Ber(p_1)$ if $i, j \in \mathcal{C}_1$ and $a_{ij} \sim Ber(q)$ if i, j are in distinct classes.

In more compact notation, let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a random matrix, whose rows are conditional i.i.d. Gaussian from two (or in general, more) high-dimensional normal distributions. Let $\mathbf{Y} = \mathbf{D}^{-1}(\mathbf{A} + \tau \mathbf{I}) \in \mathbb{R}^{n \times n}$ be a (normalized) graph convolution matrix, where $\mathbf{D}$ is the degree matrix for $\mathbf{A} + \tau \mathbf{I}$, and $\mathbf{A}$ is the graph adjacency matrix sampled from a conditional stochastic block model, $\tau \geq 0$ is a regularizing weight for added self-loops. We are interested in the distribution of $\mathbf{YX}$, and how the distribution would affect node classification accuracy. Based on the distribution of $\mathbf{X}$ and $\mathbf{YX}$, we want to obtain the accuracy of the best-performing linear classifier before and after graph convolution, respectively. Note that $\mathbf{YX}$ corresponds to applying graph convolution once, but we are also interested in applying graph convolution multiple times. Further, we aim for theoretical insights which give guidance on the power of the convolution matrix which should perform best for our classification tasks.

8.2.3 Chimpanzee Activities and Community Structures

For most of my doctoral projects, we first create a GNN and then look for applications. The chimpanzee project is motivated the other way around. In this subsection, we explore the applicability of GNNs to specific aspects of our application. We will determine the suitability of GNNs, explore potential methodologies for their implementation, and discuss how these can be integrated with results obtained from other network analysis methods.

This work, which is in progress, involves a chimpanzee data set. This data set concerns wild chimpanzees in Uganda, made available by Aaron Sandel, an Assistant Professor from the Department of Anthropology at the University of Texas at Austin. The focus of this research is on social relationships.

The main question is how best to model networks across time. The chimpanzees here have been studied for almost 30 years. Around 2015 the chimpanzees became polarized, and by 2018 they were nearly completely separate groups. The research question is how to best document this split, for example through clustering or community detection, and determine when the schism started, and ideally, incorporating covariates so as to assess possible correlates of the split.

We have constructed several types of networks based on their interactions and activities, either monthly or yearly. Based on the graphs constructed, we conduct exploratory data analysis using tools from network analysis. For example, we record the betweenness centrality values for the individuals and aim to locate important connectors, and we consider community structures for each time step by maximizing modularity. To compare the communities across time, we analyze their pairwise Adjusted Rand Index values, plot Sankey diagrams, and align the communities based on a modified version of the Hungarian algorithm. We report information such as the switching proportions of communities across time, the change in the observed number of individuals, the change in modularity values, the change in the small world index, and the eigenvalues of the Laplacian matrix. To incorporate community dynamics, we consider two notions of similarities between the nodes: the counts of shared communities across time, and the longest shared path of community identities.

However, the current edge weights are determined based on heuristics without optimization. One research question that may involve GNNs is to construct better edge weights. Based on the different types of relationships in the data (different proximity ranges, for example), we have constructed additional networks based on single relationships, where every edge is given a unit edge weight for a single day, and multiple occurrences for the same date are given a smaller addition each time to the edge weight. After that, our plan is as follows. We will assign increasing (with heuristically correct order) edge weights to combine the networks into one single weighted network. We will then conduct temporal prediction tasks on these networks and optimize the combination weights. One option is to employ the TGCN [217] model as the temporal GNN model and conduct experiments for the period 1998 to 2013 (last prediction year 2013 and first start year 1998, before the split). Another option is to investigate graph contrastive learning so that we are able to optimize the graph weights using unsupervised learning [218, 219]. We will also incorporate ideas from graph learning [220, 221]. The learned weights would be employed.

8.3 Summary and Outlook

To sum up, this dissertation centers around graph neural networks for network analysis. The dissertation provides novel GNN algorithms for solving problems in complex networks such as signed and directed graphs, with applications of directed graphs to signal recovery problems. In the future, we will deepen our contributions to the theoretical foundations of algorithm design and application-driven methodologies.

References

[1] Zonghan Wu et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[2] Jie Zhou et al. “Graph Neural Networks: A Review of Methods and Applications”. In: *AI Open* 1 (2020), pp. 57–81.

[3] Yixuan He et al. “SSSNET: Semi-Supervised Signed Network Clustering”. In: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM. 2022, pp. 244–252.

[4] Alexandre Bovet and Peter Grindrod. *The Activity of the Far Right on Telegram*. `https://www.researchgate.net/publication/346968575_The_Activity_of_the_Far_Right_on_Telegram_v21`. 2020.

[5] Yixuan He, Gesine Reinert, and Mihai Cucuringu. “DIGRAC: Digraph Clustering Based on Flow Imbalance”. In: *Learning on Graphs Conference*. PMLR. 2022, pp. 21–1.

[6] Xitong Zhang et al. “MagNet: A Neural Network for Directed Graphs”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 27003–27015.

[7] Benedek Rozemberczki et al. “PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models”. In: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. CIKM ’21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 4564–4573. URL: `https://doi.org/10.1145/3459637.3482014`.

[8] Ismail Bustany et al. “K-Specpart: Supervised Embedding Algorithms and Cut Overlay for Improved Hypergraph Partitioning”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2023).

[9] Jingnan Zhang, Xin He, and Junhui Wang. “Directed Community Detection with Network Embedding”. In: *Journal of the American Statistical Association* (2021), pp. 1–11.

[10] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. “From Louvain to Leiden: Guaranteeing Well-Connected Communities”. In: *Scientific Reports* 9.1 (2019), pp. 1–12.

[11] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to Information Retrieval*. Vol. 39. Cambridge University Press Cambridge, 2008.

[12] Tie-Yan Liu. *Learning to Rank for Information Retrieval*. Springer Science & Business Media, 2011.

[13] Vikas C Raykar and Shipeng Yu. "Ranking Annotators for Crowdsourced Labeling Tasks". In: *Advances in Neural Information Processing Systems*. 2011, pp. 1809–1817.

[14] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. "Performance of Recommender Algorithms on Top-N Recommendation Tasks". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. Association for Computing Machinery. 2010, pp. 39–46.

[15] Alkeos Tsokos et al. "Modeling Outcomes of Soccer Matches". In: *Machine Learning* 108.1 (2019), pp. 77–95.

[16] Arvind Giridhar and Praveen R Kumar. "Distributed Clock Synchronization over Wireless Networks: Algorithms and Analysis". In: *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE. 2006, pp. 4915–4920.

[17] Stella X. Yu. "Angular Embedding: From Jarring Intensity Differences to Perceived Luminance". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 2302–2309.

[18] Stella X. Yu. "Angular Embedding: A Robust Quadratic Criterion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.1 (2011), pp. 158–173.

[19] Anton Forstner et al. "Well-Conditioned Ptychographic Imaging via Lost Subspace Completion". In: *Inverse Problems* 36.10 (2020), p. 105009.

[20] Mark A Iwen et al. "Phase Retrieval from Local Measurements: Improved Robustness via Eigenvector-Based Angular Synchronization". In: *Applied and Computational Harmonic Analysis* 48.1 (2020), pp. 415–444.

[21] Mihai Cucuringu, Yaron Lipman, and Amit Singer. "Sensor Network Localization by Eigenvector Synchronization over the Euclidean Group". In: *ACM Transactions on Sensor Networks (TOSN)* 8.3 (2012), pp. 1–42.

[22] Yu Tian et al. "Rethinking Kernel Methods for Node Representation Learning on Graphs". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 11681–11692.

[23] Lawrence Hubert and Phipps Arabie. "Comparing Partitions". In: *Journal of Classification* 2.1 (1985), pp. 193–218.

[24] Franz J Brandenburg, Andreas Gleißner, and Andreas Hofmeier. "Comparing and Aggregating Partial Orders with Kendall tau distances". In: *Discrete Mathematics, Algorithms and Applications* (2013).

[25] Fajwel Fogel, Alexandre d'Aspremont, and Milan Vojnovic. "SerialRank: Spectral Ranking Using Seriation". In: *Advances in Neural Information Processing Systems* 27 (2014), pp. 900–908.

[26] Nicolas Boumal. "Nonconvex Phase Synchronization". In: *SIAM Journal on Optimization* 26.4 (2016), pp. 2355–2377.

[27] Yixuan He et al. "MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian". In: *Learning on Graphs Conference*. PMLR. 2022, pp. 40–1.

[28] Yixuan He et al. "Pytorch Geometric Signed Directed: A Software Package on Graph Neural Networks for Signed and Directed Graphs". In: *Learning on Graphs Conference*. PMLR. 2024, pp. 12–1.

[29] Xuhong Wang et al. "CEP3: Community Event Prediction with Neural Point Process on Graph". In: *Learning on Graphs Conference*. PMLR. 2022, pp. 39–1.

[30] Yixuan He et al. "GNNRank: Learning Global Rankings from Pairwise Comparisons via Directed Graph Neural Networks". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 8581–8612.

[31] Yixuan He et al. "Robust Angular Synchronization via Directed Graph Neural Networks". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=5sjxMwWmk8.

[32] Qitian Wu et al. "DIFFormer: Scalable (Graph) Transformers Induced by Energy Constrained Diffusion". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=j6zUzrapY3L.

[33] Haoyu Geng et al. "Pyramid Graph Neural Network: A Graph Sampling and Filtering Approach for Multi-Scale Disentangled Representations". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, pp. 518–530.

[34] Holly R Steach et al. "Inferring Metabolic States from Single Cell Transcriptomic Data via Geometric Deep Learning". In: *bioRxiv* (2023), pp. 2023–12.

[35] Gholamali Aminian et al. "Generalization Error of Graph Neural Networks in the Mean-Field Regime". In: *arXiv preprint arXiv:2402.07025* (2024).

[36] Marc J Perry. *State-to-State Migration Flows, 1995 to 2000*. US Department of Commerce, Economics and Statistics Administration, US Census Bureau, 2003.

[37] Samuel Franklin Sampson. *A Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships*. Ithaca, NY 14850, USA: Cornell University, 1968.

[38] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. "Signed Networks in Social Media". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 1361–1370.

[39] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-Series Clustering–a Decade Review". In: *Information Systems* 53 (2015), pp. 16–38.

[40] Jiliang Tang, Charu Aggarwal, and Huan Liu. "Recommendations in Signed Social Networks". In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 31–40.

[41] Han Xiao, Bruno Ordozgoiti, and Aristides Gionis. "Searching for Polarization in Signed Graphs: A Local Spectral Approach". eng. In: *Proceedings of The Web Conference 2020*. WWW '20 (2020), pp. 362–372.

[42] Amin Javari et al. "ROSE: Role-Based Signed Network Embedding". eng. In: *Proceedings of The Web Conference 2020*. WWW '20. Taipei: Association for Computing Machinery, 2020, pp. 2782–2788.

[43] Yiqi Chen et al. “"Bridge": Enhanced Signed Directed Network Embedding”. eng. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. CIKM ’18. Lingotto, Turin, Italy: Association for Computing Machinery, 2018, pp. 773–782.

[44] Xu Pinghua et al. “Social Trust Network Embedding”. English. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*. Vol. 2019-. Beijing, China: Institute of Electrical and Electronics Engineers Inc., 2019, pp. 678–687.

[45] Yeon-Chang Lee et al. “ASiNE: Adversarial Signed Network Embedding”. eng. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 609–618.

[46] Xu Chen et al. “Decoupled Variational Embedding for Signed Directed Networks”. In: *ACM Transactions on the Web (TWEB)* 15.1 (2020), pp. 1–31.

[47] Dengcheng Yan et al. “MUSE: Multi-Faceted Attention for Signed Network Embedding”. In: *Neurocomputing* 519 (2023), pp. 36–43.

[48] Tyler Derr, Yao Ma, and Jiliang Tang. “Signed Graph Convolutional Networks”. In: *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE. Singapore: IEEE, 2018, pp. 929–934.

[49] Suhang Wang et al. “Attributed Signed Network Embedding”. eng. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. Vol. 131841. CIKM ’17. Singapore, Singapore: Association for Computing Machinery, 2017, pp. 137–146.

[50] Jérôme Kunegis et al. “Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization”. In: *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM. Sydney, Australia: IEEE, 2010, pp. 559–570.

[51] Kai-Yang Chiang, Joyce Whang, and Inderjit Dhillon. “Scalable Clustering of Signed Networks Using Balance Normalized Cut”. eng. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM ’12. New York, United States: Association for Computing Machinery, 2012, pp. 615–624.

[52] Mihai Cucuringu et al. “SPONGE: A Generalized Eigenproblem for Clustering Signed Networks”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1088–1098.

[53] Yiqi Chen et al. “BASSI: Balance and Status Combined Signed Network Embedding”. English. In: *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10827. Gold Coast, Australia: Springer Verlag, 2018, pp. 55–63.

[54] Yu Li et al. “Learning Signed Network Embedding via Graph Attention”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 4772–4779.

[55] Junjie Huang et al. “Signed Graph Attention Networks”. English. In: *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11731. Munich, Germany: Springer Verlag, 2019, pp. 566–577.

[56] Frank Harary. "On the Notion of Balance of a Signed Graph". In: *Michigan Mathematical Journal* 2.2 (1953), pp. 143–146.

[57] Kartik Sharma et al. "Balance Maximization in Signed Networks via Edge Deletions". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 752–760.

[58] Ramanthan Guha et al. "Propagation of Trust and Distrust". In: *Proceedings of the 13th International Conference on World Wide Web*. 2004, pp. 403–412.

[59] Jiliang Tang, Xia Hu, and Huan Liu. "Is Distrust the Negation of Trust? The Value of Distrust in Social Media". In: *Proceedings of the 25th ACM Conference on Hypertext and Social Media*. 2014, pp. 148–157.

[60] Kenneth E Read. "Cultures of the Central Highlands, New Guinea". In: *Southwestern Journal of Anthropology* 10.1 (1954), pp. 1–43.

[61] Hamilton William L., Ying Rex, and Leskovec Jure. "Inductive Representation Learning on Large Graphs". English. In: *Advances in Neural Information Processing Systems*. Vol. 2017-. Long Beach, California, USA: Neural Information Processing Systems Foundation, 2017, pp. 1025–1035.

[62] Pedro Mercado, Francesco Tudisco, and Matthias Hein. "Clustering Signed Networks with the Geometric Mean of Laplacians". In: *Advances in Neural Information Processing System* 29 (2016).

[63] Quan Zheng and David B Skillicorn. "Spectral Embedding of Signed Networks". In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM. 2015, pp. 55–63.

[64] Xueyan Liu et al. "Semi-Supervised Stochastic Blockmodel for Structure Analysis of Signed Networks". In: *Knowledge-Based Systems* 195 (2020), p. 105714.

[65] "An MBO Scheme for Clustering and Semi-Supervised Clustering of Signed Networks". In: *Communications in Mathematical Sciences* 19.1 (2021), pp. 73–109.

[66] Kumar Ayan Bhowmick, Koushik Meneni, and Bivas Mitra. "On the Network Embedding in Sparse Signed Networks". In: *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11441. Macau, China: Springer Verlag, 2019, pp. 94–106.

[67] Suhang Wang et al. "Signed Network Embedding in Social Media". In: *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM. 2017, pp. 327–335.

[68] Ruo-Chun Tzeng, Bruno Ordozgoiti, and Aristides Gionis. "Discovering Conflicting Groups in Signed Networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 10974–10985.

[69] Francesco Bonchi et al. "Discovering Polarized Communities in Signed Networks". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. Beijing, China: Association for Computing Machinery, 2019, pp. 961–970.

[70] Mongi Arfaoui and Aymen Ben Rejeb. "Oil, Gold, US Dollar and Stock Market Interdependencies: A Global Analytical Insight". In: *European Journal of Management and Business Economics* 26 (2017), pp. 278–293.

[71] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[72] Gero Greiner and Riko Jacob. "The I/O Complexity of Sparse Matrix Dense Matrix Multiplication". In: *LATIN 2010: Theoretical Informatics: 9th Latin American Symposium, Oaxaca, Mexico, April 19-23, 2010. Proceedings 9*. Springer. 2010, pp. 143–156.

[73] Matthias Fey et al. "GNNAutoScale: Scalable and Expressive Graph Neural Networks via Historical Embeddings". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3294–3304.

[74] Simone Romano et al. "Standardized Mutual Information for Clustering Comparisons: One Step Further in Adjustment for Chance". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1143–1151.

[75] Alessia Amelio and Clara Pizzuti. "Is Normalized Mutual Information a Fair Measure for Comparing Community Detection Methods?" In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 2015, pp. 1584–1585.

[76] Michael Bertolacci et al. "Climate Inference on Daily Rainfall Across the Australian Continent, 1876–2015". In: *Annals of Applied Statistics* 13.2 (2019), pp. 683–712.

[77] yahoo! finance. *S&P 1500 Data Set Link*. https://finance.yahoo.com/. [Online; accessed 19-January-2021]. 2021.

[78] Arunachalam Vinayagam et al. "Integrating Protein-Protein Interaction Networks with Phenotypes Reveals Signs of Interactions". In: *Nature methods* 11.1 (2014), pp. 94–99.

[79] Robert West et al. "Exploiting Social Network Structure for Person-to-Person Sentiment Analysis". In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 297–310.

[80] Arti Patidar, Vinti Agarwal, and Kamal Kant Bharadwaj. "Predicting Friends and Foes in Signed Networks Using Inductive Inference and Social Balance Theory". In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE. 2012, pp. 384–388.

[81] Yaoping Hou, Jiongsheng Li, and Yongliang Pan. "On the Laplacian Eigenvalues of Signed Graphs". In: *Linear and Multilinear Algebra* 51.1 (2003), pp. 21–30.

[82] Maria A Riolo et al. "Efficient Method for Estimating the Number of Communities in a Network". In: *Physical review e* 96.3 (2017), p. 032310.

[83] Kamalika Chaudhuri, Fan Chung, and Alexander Tsiatas. "Spectral Clustering of Graphs with General Degrees in the Extended Planted Partition Model". In: *25th Annual Conference on Learning Theory*. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: JMLR Workshop and Conference Proceedings, 2012, pp. 35.1–35.23.

[84] Arash A. Amini et al. "Pseudo-Likelihood Methods for Community Detection in Large Sparse Networks". In: *The Annals of Statistics* 41.4 (2013), pp. 2097–2122.

[85] Mihai Cucuringu et al. “Regularized Spectral Methods for Clustering Signed Networks”. In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 12057–12135.

[86] Abhay Jha et al. “Clustering to Forecast Sparse Time-Series Data”. In: *2015 IEEE 31st International Conference on Data Engineering*. IEEE. Seoul, South Korea: IEEE, 2015, pp. 1388–1399.

[87] Mihai Cucuringu et al. “Scalable Constrained Clustering: A Generalized Spectral Method”. In: *Artificial Intelligence and Statistics Conference (AISTATS)* (2016).

[88] Fragkiskos D Malliaros and Michalis Vazirgiannis. “Clustering and Community Detection in Directed Networks: A Survey”. In: *Physics Reports* 533.4 (2013), pp. 95–142.

[89] William R. Palmer and Tian Zheng. “Spectral Clustering for Directed Networks”. In: *Complex Networks & Their Applications IX*. Ed. by Rosa M. Benito et al. Cham: Springer International Publishing, 2021, pp. 87–99.

[90] Andrew Elliott et al. “Core–Periphery Structure in Directed Networks”. In: *Proceedings of the Royal Society A* 476.2241 (2020), p. 20190783.

[91] Michelle Girvan and Mark EJ Newman. “Community Structure in Social and Biological Networks”. In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826.

[92] Mark EJ Newman. “Modularity and Community Structure in Networks”. In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582.

[93] Jure Leskovec et al. “Statistical Properties of Community Structure in Large Social and Information Networks”. In: *Proceedings of the 17th International Conference on World Wide Web*. 2008, pp. 695–704.

[94] Elizabeth A Leicht and Mark EJ Newman. “Community Structure in Directed Networks”. In: *Physical Review Letters* 100.11 (2008), p. 118703.

[95] Yilin Chen and Jack W Baker. “Community Detection in Spatial Correlation Graphs: Application to Non-Stationary Ground Motion Modeling”. In: *Computers and Geosciences* 154 (2021), p. 104779.

[96] Caiyan Jia et al. “Node Attribute-Enhanced Community Detection in Complex Networks”. In: *Scientific Reports* 7.1 (2017), pp. 1–15.

[97] Mihai Cucuringu et al. “Hermitian Matrices for Clustering Directed Graphs: Insights and Applications”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 983–992.

[98] Steinar Laenen and He Sun. “Higher-Order Spectral Clustering of Directed Graphs”. In: *Advances in Neural Information Processing Systems* (2020).

[99] Martin Rosvall and Carl T Bergstrom. “Maps of Random Walks on Complex Networks Reveal Community Structure”. In: *Proceedings of the national academy of sciences* 105.4 (2008), pp. 1118–1123.

[100] Kun Deng, Prashant G Mehta, and Sean P Meyn. “Optimal Kullback-Leibler Aggregation via Spectral Theory of Markov Chains”. In: *IEEE Transactions on Automatic Control* 56.12 (2011), pp. 2793–2808.

[101] Bernhard C Geiger et al. "Optimal Kullback–Leibler Aggregation via Information Bottleneck". In: *IEEE Transactions on Automatic Control* 60.4 (2014), pp. 1010–1022.

[102] Ali Shojaie and Emily B Fox. "Granger Causality: A Review and Recent Advances". In: *Annual Review of Statistics and Its Application* 9 (2022), pp. 289–319.

[103] Mukeshwar Dhamala, Govindan Rangarajan, and Mingzhou Ding. "Analyzing Information Flow in Brain Networks with Nonparametric Granger Causality". In: *NeuroImage* 41.2 (2008), pp. 354–362. URL: `https://www.sciencedirect.com/science/article/pii/S1053811908001328`.

[104] Jakob Runge et al. "Detecting and Quantifying Causal Associations in Large Nonlinear Time Series Datasets". In: *Science Advances* 5.11 (2019).

[105] Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. "Detection and Clustering of Lead-Lag Networks for Multivariate Time Series with an Application to Financial Markets". In: *7th SIGKDD Workshop on Mining and Learning from Time Series (MiLeTS)* (2021).

[106] Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. "Lead–Lag Detection and Network Clustering for Multivariate Time Series with an Application to the US Equity Market". In: *Machine Learning* (2022), pp. 1–42.

[107] A Harzallah and R Sadourny. "Observed Lead-Lag Relationships between Indian Summer Monsoon and Some Meteorological Variables". In: *Climate Dynamics* 13.9 (1997), pp. 635–648.

[108] Andrew Elliott et al. "Anomaly Detection in Networks with Application to Financial Transaction Networks". In: *arXiv preprint arXiv:1901.00402* (2019).

[109] Meng-Chieh Lee et al. "GAWD: Graph Anomaly Detection in Weighted Directed Graph Databases". In: *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2021, pp. 143–150.

[110] Lingxiao Zhao et al. "Graph Anomaly Detection with Unsupervised GNNs". In: *arXiv preprint arXiv:2210.09535* (2022).

[111] Venu Satuluri and Srinivasan Parthasarathy. "Symmetrizations for Clustering Directed Graphs". In: *Proceedings of the 14th International Conference on Extending Database Technology*. 2011, pp. 343–354.

[112] Karl Rohe, Tai Qin, and Bin Yu. "Co-Clustering Directed Graphs to Discover Asymmetries and Directional Communities". In: *Proceedings of the National Academy of Sciences* 113.45 (2016), pp. 12679–12684.

[113] Zekun Tong et al. "Directed Graph Convolutional Network". In: *arXiv preprint arXiv:2004.13970* (2020).

[114] Zekun Tong et al. "Digraph Inception Convolutional Networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17907–17918.

[115] Bojan Mohar. "A New Kind of Hermitian Matrices for Digraphs". In: *Linear Algebra and its Applications* 584 (2020), pp. 343–352.

[116] Zekun Tong et al. "Directed Graph Contrastive Learning". In: *Advances in Neural Information Processing Systems* 34 (2021).

[117] Yi Ma et al. “Spectral-Based Graph Convolutional Network for Directed Graphs”. In: *arXiv preprint arXiv:1907.08990* (2019).

[118] Federico Monti, Karl Otness, and Michael M. Bronstein. “MotifNet: A Motif-Based Graph Convolutional Network for Directed Graphs”. In: (2018).

[119] Yuan Yao et al. “Flow-Based Clustering on Directed Graphs: A Structural Entropy Minimization Approach”. In: *IEEE Access* 8 (2019), pp. 152579–152591.

[120] Andrea Lancichinetti et al. “Finding Statistically Significant Communities in Networks”. In: *PloS one* 6.4 (2011), e18961.

[121] Nicolas Dugué and Anthony Perez. “Directed Louvain: Maximizing Modularity in Directed Networks”. PhD thesis. Université d’Orléans, 2015.

[122] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. “Spectral Clustering with Graph Neural Networks for Graph Pooling”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 874–883.

[123] Hongteng Xu, Dixin Luo, and Lawrence Carin. “Scalable Gromov-Wasserstein Learning for Graph Partitioning and Matching”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 3052–3062.

[124] Hongteng Xu et al. “Gromov-Wasserstein Learning for Graph Matching and Node Embedding”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6932–6941.

[125] Samir Chowdhury and Tom Needham. “Generalized Spectral Clustering via Gromov-Wasserstein Learning”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 712–720.

[126] Hongteng Xu. “Gromov-Wasserstein Factorization Models for Graph Clustering”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 6478–6485.

[127] Xin Liu, Hui-Min Cheng, and Zhong-Yuan Zhang. “Evaluation of Community Detection Methods”. In: *IEEE Transactions on Knowledge and Data Engineering* 32.9 (2019), pp. 1736–1746.

[128] Andrew Elliott et al. “Anomaly Detection in Networks Using Spectral Methods and Network Comparison Approaches”. In: *arXiv preprint arXiv:1901.00402* (2019).

[129] Lada A Adamic and Natalie Glance. “The Political Blogosphere and the 2004 US Election: Divided They Blog”. In: *Proceedings of the 3rd International Workshop on Link Discovery*. 2005, pp. 36–43.

[130] Sugato Basu, Ian Davidson, and Kiri Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. CRC Press, 2008.

[131] Mihai Cucuringu et al. “Scalable Constrained Clustering: A Generalized Spectral Method”. In: *Artificial Intelligence and Statistics Conference (AISTATS) 2016* (2016).

[132] Srijan Kumar et al. “Edge Weight Prediction in Weighted Signed Networks”. In: *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE. Barcelona, Spain: IEEE, 2016, pp. 221–230.

[133] Junjie Huang et al. "SDGNN: Learning Node Representation for Signed Directed Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 1. 2021, pp. 196–203.

[134] Quan Zheng, David B Skillicorn, and Olivier Walther. "Signed Directed Social Network Analysis Applied to Group Conflict". In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE. 2015, pp. 1007–1014.

[135] Jiangping Hu and Wei Xing Zheng. "Bipartite Consensus for Multi-Agent Systems on Directed Signed Networks". In: *52nd IEEE Conference on Decision and Control*. IEEE. 2013, pp. 3451–3456.

[136] Weijia Ju et al. "A New Algorithm for Positive Influence Maximization in Signed Networks". In: *Information Sciences* 512 (2020), pp. 1571–1591.

[137] Junjie Huang et al. "Signed Graph Attention Networks". In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 566–577.

[138] Stefano Fiorini et al. "SigMaNet: One Laplacian to Rule Them All". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6. 2023, pp. 7568–7576.

[139] Petar Veličković et al. "Graph Attention Networks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[140] Rahul Singh and Yongxin Chen. "Signed Graph Neural Networks: A Frequency Perspective". In: *Transactions on Machine Learning Research* (2023). URL: https://openreview.net/forum?id=RZveYHgZbu.

[141] Michael Schlichtkrull et al. "Modeling Relational Data with Graph Convolutional Networks". In: *European Semantic Web Conference*. Springer. 2018, pp. 593–607.

[142] Shikhar Vashishth et al. "Composition-Based Multi-Relational Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=BylA_C4tPr.

[143] Zhao Zhang et al. "Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 9612–9619.

[144] Michaël Fanuel, Carlos M Alaiz, and Johan AK Suykens. "Magnetic Eigenmaps for Community Detection in Directed Networks". In: *Physical Review E* 95.2 (2017), p. 022302.

[145] Michaël Fanuel et al. "Magnetic Eigenmaps for the Visualization of Directed Networks". In: *Applied and Computational Harmonic Analysis* 44 (2018), pp. 189–199.

[146] Bruno Messias F. de Resende and Luciano da F. Costa. "Characterization and Comparison of Large Directed Networks Through the Spectra of the Magnetic Laplacian". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.7 (2020), p. 073141.

[147] Fatihcan M. Atay and Hande Tunçel. "On the Spectrum of the Normalized Laplacian for Signed Graphs: Interlacing, Contraction, and Replication". In: *Linear Algebra and its Applications* 442 (2014). Special Issue on Spectral Graph Theory on the occasion of the Latin Ibero-American Spectral Graph Theory Workshop (Rio de Janeiro, 27-28 September 2012), pp. 165–177. URL: https://www.sciencedirect.com/science/article/pii/S0024379513005211.

[148] M. Defferrard, X. Bresson, and P. Vandergheynst. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *Advances in Neural Information Processing Systems*. Neural Information Processing Systems Foundation, 2016, pp. 3844–3852.

[149] Alexander Cloninger. "A Note on Markov Normalized Magnetic Eigenmaps". In: *Applied and Computational Harmonic Analysis* 43.2 (2017), pp. 370–380.

[150] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. "Wavelets on Graphs via Spectral Graph Theory". In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[151] Joan Bruna et al. "Spectral Networks and Deep Locally Connected Networks on Graphs". In: *International Conference on Learning Representations (ICLR)*. 2014.

[152] Ron Levie et al. "Transferability of Spectral Graph Convolutional Neural Networks". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 12462–12520.

[153] Yixuan He. "GNNs for Node Clustering in Signed and Directed Networks". In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 2022, pp. 1547–1548.

[154] Bruno Ordozgoiti, Antonis Matakos, and Aristides Gionis. "Finding Large Balanced Subgraphs in Signed Networks". In: *Proceedings of The Web Conference 2020*. WWW '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 1378–1388. URL: https://doi.org/10.1145/3366423.3380212.

[155] Paolo Massa and Paolo Avesani. "Controversial Users Demand Local Trust Metrics: An Experimental Study on Epinions.com Community". In: *AAAI*. 2005, pp. 121–126.

[156] Clive WJ Granger. "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods". In: *Econometrica: Journal of the Econometric Society* (1969), pp. 424–438.

[157] Lionel Barnett and Anil K Seth. "Granger Causality for State-Space Models". In: *Physical Review E* 91.4 (2015), p. 040101.

[158] Quang-Vinh Dang and Claudia-Lavinia Ignat. "Link-Sign Prediction in Dynamic Signed Directed Networks". In: *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE. 2018, pp. 36–45.

[159] M. G. Kendall and B. B. Smith. "On the Method of Paired Comparisons". In: *Biometrika* 31.3-4 (1940), pp. 324–345.

[160] J. Atkins, E. Boman, and B. Hendrickson. "A Spectral Algorithm for Seriation and the Consecutive Ones Problem". In: *SIAM Journal on Computing* (1998).

[161] Miroslav Fiedler. "Algebraic Connectivity of Graphs". In: *Czechoslovak Mathematical Journal* 23.2 (1973), pp. 298–305.

[162] Yannan Chen, Liqun Qi, and Xiaoyan Zhang. "The Fiedler vector of a Laplacian tensor for hypergraph partitioning". In: *SIAM Journal on Scientific Computing* 39.6 (2017), A2508–A2537.

[163] Jingjing Wang et al. "Top-N Personalized Recommendation with Graph Neural Networks in Moocs". In: *Computers and Education: Artificial Intelligence* 2 (2021), p. 100010.

[164] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. "Graph Neural Networks for Fast Node Ranking Approximation". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15.5 (2021), pp. 1–32.

[165] Alexandre d'Aspremont, Mihai Cucuringu, and Hemant Tyagi. "Ranking and Synchronization from Pairwise Measurements via SVD." In: *Journal of Machine Learning Research* 22 (2021), pp. 19–1.

[166] Iqbal Ali, Wade D. Cook, and Moshe Kress. "On the Minimum Violations Ranking of a Tournament". In: *Management Science* 32.6 (1986), pp. 660–672. URL: `http://www.jstor.org/stable/2631621`.

[167] Ralph Allan Bradley and Milton E Terry. "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[168] R Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Courier Corporation, 1959.

[169] David Firth. "Bradley-Terry models in R". In: *Journal of Statistical Software* 12.1 (2005), pp. 1–12.

[170] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, USA, 2010.

[171] L. Page et al. "The Pagerank Citation Ranking: Bringing Order to the Web". In: *Proceedings of the 7th International World Wide Web Conference*. 1998, pp. 161–172.

[172] Phillip Bonacich. "Power and Centrality: A Family of Measures". In: *American Journal of Sociology* 92.5 (1987), pp. 1170–1182.

[173] Sahand Negahban, Sewoong Oh, and Devavrat Shah. "Rank Centrality: Ranking from Pairwise Comparisons". In: *Operations Research* (2017).

[174] Herbert A David. "Ranking from Unbalanced Paired-Comparison Data". In: *Biometrika* 74.2 (1987), pp. 432–436.

[175] Mangesh Gupte et al. "Finding Hierarchy in Directed Online Social Networks". In: *Proceedings of the 20th International Conference on the World Wide Web*. 2011, pp. 557–566.

[176] Mihai Cucuringu. "Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and SDP Synchronization". In: *IEEE Transactions on Network Science and Engineering* 3.1 (2016), pp. 58–79.

[177] A. Singer. "Angular Synchronization by Eigenvectors and Semidefinite Programming". In: *Applied and Computational Harmonic Analysis* 30.1 (2011), pp. 20–36.

[178] Caterina De Bacco, Daniel B Larremore, and Cristopher Moore. "A Physical Model for Efficient Ranking in Networks". In: *Science advances* 4.7 (2018), eaar8260.

[179] Franco Scarselli et al. "Graph Neural Networks for Ranking Web Pages". In: *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*. IEEE. 2005.

[180] Clemens Damke and Eyke Hüllermeier. "Ranking Structured Objects with Graph Neural Networks". In: *Discovery Science: 24th International Conference, DS 2021, Halifax, NS, Canada, October 11–13, 2021, Proceedings 24*. Springer. 2021, pp. 166–180.

[181] Leonardo Rigutini et al. "SortNet: Learning to Rank by A Neural Preference Function". In: *IEEE Transactions on Neural Networks* 22.9 (2011), pp. 1368–1380.

[182] Marius Köppel et al. "Pairwise Learning to Rank by Neural Networks Revisited: Reconstruction, Theoretical Analysis and Practical Performance". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part III*. Springer. 2020, pp. 237–252.

[183] Chris Burges et al. "Learning to Rank Using Gradient Descent". In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005, pp. 89–96.

[184] Karol Gregor and Yann LeCun. "Learning Fast Approximations of Sparse Coding". In: *Proceedings of the 27th International Conference on Machine Learning*. 2010, pp. 399–406.

[185] Ziwei Liu et al. "Deep Learning Markov Random Field for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.8 (2017), pp. 1814–1828.

[186] Arindam Chowdhury et al. "Unfolding WMMSE Using Graph Neural Networks for Efficient Power Allocation". In: *IEEE Transactions on Wireless Communications* (2021).

[187] Yongyi Yang et al. "Graph Neural Networks Inspired by Classical Iterative Algorithms". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 11773–11783.

[188] El-Ghazali Talbi. "A Taxonomy of Metaheuristics for Bi-Level Optimization". In: *Metaheuristics for Bi-Level Optimization*. Springer, 2013, pp. 1–39.

[189] Ulrike Von Luxburg. "A Tutorial on Spectral Clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[190] Elizabeth A Hobson and Simon DeDeo. "Social Feedback and the Emergence of Rank in Animal Society". In: *PLoS Computational Biology* 11.9 (2015), e1004411.

[191] Aaron Clauset, Samuel Arbesman, and Daniel B Larremore. "Systematic Inequality and Hierarchy in Faculty Hiring Networks". In: *Science Advances* 1.1 (2015), e1400005.

[192] Federica Arrigoni and Andrea Fusiello. "Synchronization Problems in Computer Vision with Closed-Form Solutions". In: *International Journal of Computer Vision* 128.1 (2020), pp. 26–52.

[193] Noam Janco and Tamir Bendory. "Unrolled Algorithms for Group Synchronization". In: *IEEE Open Journal of Signal Processing* (2023).

[194] Mihai Cucuringu, Amit Singer, and David Cowburn. "Eigenvector Synchronization, Graph Rigidity and the Molecule Problem". In: *Information and Inference: A Journal of the IMA* 1.1 (2012), pp. 21–67.

[195] Vahan Huroyan, Gilad Lerman, and Hau-Tieng Wu. "Solving Jigsaw Puzzles by the Graph Connection Laplacian". In: *SIAM Journal on Imaging Sciences* 13.4 (2020), pp. 1717–1753.

[196] Mihai Cucuringu and Hemant Tyagi. "An Extension of the Angular Synchronization Problem to the Heterogeneous Setting". In: *Foundations of Data Science* 4.1 (2022), pp. 71–122. URL: /article/id/61ea042f2d80b7489437f000.

[197] Amit Singer. "Angular Synchronization by Eigenvectors and Semidefinite Programming". In: *Applied and Computational Harmonic Analysis* 30.1 (2011), pp. 20–36.

[198] Afonso S Bandeira, Nicolas Boumal, and Amit Singer. "Tightness of the Maximum Likelihood Semidefinite Relaxation for Angular Synchronization". In: *Mathematical Programming* 163.1 (2017), pp. 145–167.

[199] Xiangru Huang et al. "Translation Synchronization via Truncated Least Squares". In: *Advances in Neural Information Processing Systems* 30 (2017).

[200] Tingran Gao and Zhizhen Zhao. "Multi-Frequency Phase Synchronization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2132–2141.

[201] Huikang Liu, Man-Chung Yue, and Anthony Man-Cho So. "A Unified Approach to Synchronization Problems over Subgroups of the Orthogonal Group". In: *Applied and Computational Harmonic Analysis* (2023).

[202] Frank Filbir, Felix Krahmer, and Oleh Melnyk. "On Recovery Guarantees for Angular Synchronization". In: *Journal of Fourier Analysis and Applications* 27.2 (2021), pp. 1–26.

[203] Gilad Lerman and Yunpeng Shi. "Robust Group Synchronization via Cycle-Edge Message Passing". In: *Foundations of Computational Mathematics* 22.6 (2022), pp. 1665–1741.

[204] Tyler Maunu and Gilad Lerman. "Depth Descent Synchronization in SO (d)". In: *International journal of computer vision* (2023), pp. 1–19.

[205] Yunpeng Shi and Gilad Lerman. "Message Passing Least Squares Framework and Its Application to Rotation Synchronization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8796–8806.

[206] Yunpeng Shi, Cole M Wyeth, and Gilad Lerman. "Robust Group Synchronization via Quadratic Programming". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20095–20105.

[207] Jiajin Li, Anthony Man-Cho So, and Wing-Kin Ma. "Understanding Notions of Stationarity in Nonsmooth Optimization: A Guided Tour of Various Constructions of Subdifferential for Nonsmooth Functions". In: *IEEE Signal Processing Magazine* 37.5 (2020), pp. 18–31.

[208] Amit Singer and Yoel Shkolnisky. "Three-Dimensional Structure Determination from Common Lines in Cryo-Em by Eigenvectors and Semidefinite Programming". In: *SIAM journal on imaging sciences* 4.2 (2011), pp. 543–572.

[209] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring Network Structure, Dynamics, and Function Using Networkx*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[210] John C Gower. "Generalized Procrustes Analysis". In: *Psychometrika* 40 (1975), pp. 33–51.

[211] Ulrike Von Luxburg, Robert C Williamson, and Isabelle Guyon. "Clustering: Science or Art?" In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 65–79.

[212] Daniele Calandriello et al. "Improved Large-Scale Graph Learning Through Ridge Spectral Sparsification". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 688–697.

[213] Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. "Generalization and Representational Limits of Graph Neural Networks". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3419–3430.

[214] Ismail Bustany et al. "SpecPart: A Supervised Spectral Framework for Hypergraph Partitioning Solution Improvement". In: *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*. 2022, pp. 1–9.

[215] Zexi Liu et al. "Hypergraph Transformer for Semi-Supervised Classification". In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 7515–7519.

[216] Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. "Graph Convolution for Semi-Supervised Classification: Improved Linear Separability and Out-of-Distribution Generalization". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 684–693. URL: http://proceedings.mlr.press/v139/baranwal21a.html.

[217] Ling Zhao et al. "T-Gcn: A Temporal Graph Convolutional Network for Traffic Prediction". In: *IEEE transactions on intelligent transportation systems* 21.9 (2019), pp. 3848–3858.

[218] Petar Veličković et al. "Deep Graph Infomax". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rklz9iAcKQ.

[219] Yizhen Zheng et al. "Rethinking and Scaling Up Graph Contrastive Learning: An Extremely Efficient Approach with Group Discrimination". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 10809–10820.

[220] Xiaowen Dong et al. "Learning Graphs from Data: A Signal Representation Perspective". In: *IEEE Signal Processing Magazine* 36.3 (2019), pp. 44–63.

[221] Xingyue Pu et al. "Learning to learn graph topologies". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4249–4262.

[222] Adam P Harrison and Dileepan Joseph. "High Performance Rearrangement and Multiplication Routines for Sparse Tensor Arithmetic". In: *SIAM Journal on Scientific Computing* 40.2 (2018), pp. C258–C281.

[223] Elan Sopher Markowitz et al. "Graph Traversal with Tensor Functionals: A Meta-Algorithm for Scalable Learning". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=6DOZ8XNNfGN.

[224] Louis HY Chen, Larry Goldstein, and Qi-Man Shao. *Normal Approximation by Stein's Method*. Springer Science & Business Media, 2010.

[225] Ryan L Phillips and Rita Ormsby. "Industry Classification Schemes: An Analysis and Review". In: *Journal of Business & Finance Librarianship* 21.1 (2016), pp. 1–25.

[226] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *Proceedings of the 3rd International Conference on Learning Representations*. 2015.

[227] Pan Zhang. "Evaluating Accuracy of Community Detection Using the Relative Normalized Mutual Information". In: *Journal of Statistical Mechanics: Theory and Experiment* 2015.11 (2015), P11006.

[228] Mihai Cucuringu, Vincent Blondel, and Paul Van Dooren. "Extracting Spatial Information from Networks with Low-Order Eigenvectors". In: *Phys. Rev. E* 87 (3 Mar. 2013), p. 032803. URL: http://link.aps.org/doi/10.1103/PhysRevE.87.032803.

[229] Stijn Van Dongen. "Graph Clustering via a Discrete Uncoupling Process". In: *SIAM Journal on Matrix Analysis and Applications* 30.1 (2008), pp. 121–141.

[230] Trung Vu, Raviv Raich, and Xiao Fu. "On Convergence of Projected Gradient Descent for Minimizing a Large-Scale Quadratic over the Unit Sphere". In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2019, pp. 1–6.

[231] R Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Vol. 317. Springer Science & Business Media, 2009.

[232] Chandler Davis and William Morton Kahan. "The Rotation of Eigenvectors by a Perturbation. III". In: *SIAM Journal on Numerical Analysis* 7.1 (1970), pp. 1–46.

[233] Ren-Cang Li. "Relative Perturbation Theory: II. Eigenspace and Singular Subspace Variations". In: *SIAM Journal on Matrix Analysis and Applications* 20.2 (1998), pp. 471–492.

[234] Yi Yu, Tengyao Wang, and Richard J Samworth. "A Useful Variant of the Davis–Kahan Theorem for Statisticians". In: *Biometrika* 102.2 (2015), pp. 315–323.

[235] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. "Graph Neural Networks: Architectures, Stability, and Transferability". In: *Proceedings of the IEEE* 109.5 (2021), pp. 660–682.

[236] Kurt Plarre and PR Kumar. "Object Tracking by Scattered Directional Sensors". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 3123–3128.

[237] Alan Mainwaring et al. “Wireless Sensor Networks for Habitat Monitoring”. In: *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*. 2002, pp. 88–97.

Appendices

Contents (Appendix)

A

SSSNET: Semi-Supervised Signed Network Clustering Supplementary Information

A.1 Additional Results

A.1.1 Additional Results on Synthetic Data

Figure A.1 provides further results on synthetic data. In addition to the ARI scores for two more synthetic settings, SSBM($n = 1000, K = 20, p = 0.01, \rho = 1.5$) and SSBM($n = 1000, K = 2, p = 0.1, \rho = 2$), we also report the NMI scores, Balanced Normalized Cut values $\mathcal{L}_{\text{BNC}}$, and unhappy ratios, on some synthetic data used in Figure 2.5 in the main text. From Figure A.1, we remark that SSSNET gives comparable balanced normalized cut values and unhappy ratios in these regimes, and leading performance in terms of both ARI and NMI.

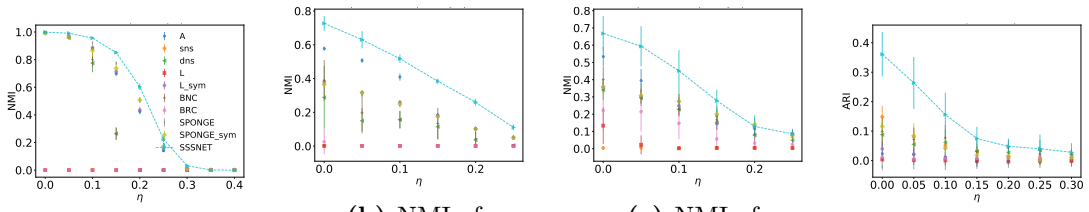
A.1.2 Additional Results on Real-World Data

Discussion on Attributes for *Sampson*

On this data set, SSSNET with the 'Cloisterville' attribute achieves highest ARI. When ignoring this attribute and instead using the identity matrix with 25 rows as input feature matrix for *Sampson*, we achieve a test ARI 0.37±0.19, which is much lower than SSSNET's test ARI with 1-dimensional attributes, but still higher than the other methods.

Extended Result Table for *Fin-YNet*

Table A.1 gives an extended comparison of different methods on the financial correlation data set *Fin-YNet*. In the first panel of table the difference (± 1 s.e. when applicable) to the best-performing method is given; hence each row will have at least one zero entry. We conclude that SSSNET attains the best performance in terms of both ARI and NMI, with regards to both test nodes and all nodes, in each of the 21 years.

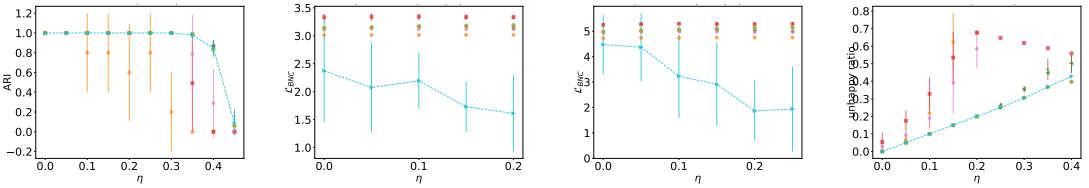


(a) NMI of SSBM($n = 30000, K = 5, p = 0.001, \rho = 1.5$)

(b) NMI of POL-SSBM($n = 10000, n_c = 2, p = 0.01, \rho = 1.5$)

(c) NMI of POL-SSBM($n = 1050, n_c = 2, p = 0.1, \rho = 1$)

(d) ARI of SSBM($n = 1000, K = 20, p = 0.01, \rho = 1.5$)



(e) ARI of SSBM($n = 1000, K = 2, p = 0.1, \rho = 2$)

(f) $\mathcal{L}_{\text{BNC}}$ of POL-SSBM($n = 5000, n_c = 3, p = 0.1, \rho = 1.5$)

(g) $\mathcal{L}_{\text{BNC}}$ of POL-SSBM($n = 5000, n_c = 5, p = 0.1, \rho = 1.5$)

(h) Unhappy ratio of SSBM($n = 10000, K = 5, p = 0.01, \rho = 1.5$)

Figure A.1: Extended node clustering result comparison on synthetic data. Dashed lines are added to SSSNET's performance to highlight our result. Each setting is averaged over ten runs. Error bars are given by standard errors. NMI and ARI results are on test nodes only (the higher, the better), while $\mathcal{L}_{\text{BNC}}$ and unhappy ratio results are on all nodes in the signed network (the lower, the better).

Table A.1: Clustering performance comparison on *Fin-YNet*; the first panel shows distance to the best performance and the second panel shows absolute performance. The best is in **bold red**, and second best in underline blue. Standard deviations are not shown due to space constraint.

Metric	A	sns	dns	L	L_sym	BNC	BRC	SPONGE	SPONGE_sym	SSSNET
test ARI dist.	0.22	0.37	0.32	0.33	0.22	0.32	0.33	0.20	0.16	**0.00**
all ARI dist.	0.27	0.43	0.37	0.38	0.27	0.37	0.38	0.24	0.2	**0.00**
test NMI dist.	0.11	0.53	0.39	0.39	0.14	0.39	0.40	0.12	0.09	**0.00**
all NMI dist.	0.17	0.44	0.35	0.36	0.19	0.35	0.35	0.12	0.11	**0.00**
test ARI	0.18	0.03	0.08	0.07	0.17	0.08	0.07	0.19	0.24	**0.40**
all ARI	0.19	0.03	0.09	0.08	0.19	0.09	0.08	0.22	0.26	**0.46**
test NMI	0.54	0.12	0.26	0.26	0.51	0.26	0.25	0.53	0.56	**0.65**
all NMI	0.38	0.11	0.20	0.19	0.36	0.20	0.19	0.42	0.44	**0.55**

GICS Alignments Plots on S&P1500

We remark that SSSNET (Figure A.2) uncovers several very cohesive clusters, such as IT, Discretionary, Utility, and Financials. These recovered clusters are visually more cohesive than those reported by SPONGE.

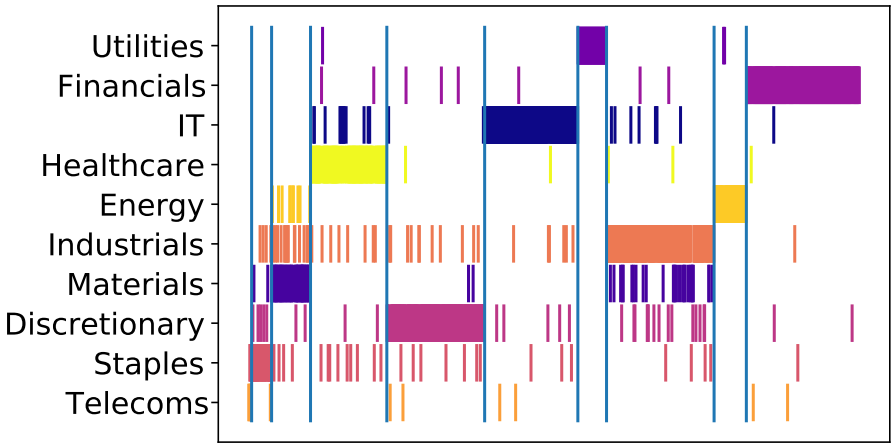


Figure A.2: Alignment of SSSNET clusters with GICS sectors in S&P 1500; ARI=0.71. Colors denote distinct sectors of the US economy, indexing the rows; the total area of a color denotes the size of a GICS sector. Columns index the recovered SSSNET clusters, with the widths proportional to cluster sizes.

A.2 Extended Data Description

A.2.1 SSBM Construction Details

A Signed Stochastic Block Model (SSBM) for a network on n nodes with K blocks (clusters), is constructed similar to [52] but with a more general cluster size definition. • (1) Assign block sizes $n_0 \leq n_1 \leq \cdots \leq n_{K-1}$ with size ratio $\rho \geq 1$, as follows. If $\rho = 1$, then the first $K-1$ blocks have the same size $\lfloor n/K \rfloor$, and the last block has size $n - (K-1)\lfloor n/K \rfloor$. If $\rho > 1$, we set $\rho_0 = \rho^{\frac{1}{K-1}}$. Solving $\sum_{i=0}^{K-1} \rho_0^i n_0 = n$ and taking integer value gives $n_0 = \left\lfloor n(1-\rho_0)/(1-\rho_0^K) \right\rfloor$. Further, set $n_i = \lfloor \rho_0 n_{i-1} \rfloor$, for $i = 1, \cdots, K-2$ if $K \geq 3$, and $n_{K-1} = n - \sum_{i=0}^{K-2} n_i$. Then, the ratio of the size of the largest to the smallest block is approximately $\rho_0^{K-1} = \rho$. • (2) Assign each node to one of K blocks, so that each block has the allocated size. • (3) For each pair of nodes in the same block, with probability $p_{\text{in}} = p$, create an edge with $+1$ as weight between them, independently of the other potential edges. • (4) For each pair of nodes in different blocks, with probability $p_{\text{out}} = p$, create an edge with -1 as weight between them, independently of the other potential edges. • (5) Flip the sign of the across-cluster edges from the previous stage with sign flip probability $\eta_{\text{in}} = \eta$, and $\eta_{\text{out}} = \eta$ for edges within and across clusters, respectively.

As our framework can be applied to different connected components separately, after generating the initial SSBM, we concentrate on the largest connected component. To further avoid numerical issues, we modify the synthetic network by adding randomly wired edges to nodes of degree 1 or 2.

The actual implementation of the above algorithm is given in `https://github.com/SherylHYX/SSSNET_Signed_Clustering/blob/main/src/utils.py`, modified from `https://github.com/alan-turing-institute/SigNet/blob/master/signet/block_models.py`.

A.2.2 Discussion on POL-SSBM

Our generalizations are as follows: • (1) Our model allows for different sizes of communities and blocks, governed by the parameter ρ, which is more realistic. • (2) Instead of using a single parameter for the edge probability and sign flips, we use two parameters p and η. We also assume equal edge sampling probability

throughout the entire graph, as we want to avoid being able to trivially solve the problem by considering the absolute value of the edge weights, and thus falling back onto the standard community detection setting in unsigned graphs. • (3) We consider more than two polarized communities, while also allowing for the existence of ambient nodes in the graph, in the spirit of [41].

A.2.3 Real-World Data Description

We perform experiments on six real-world signed network data sets (*Sampson* [37], *Rainfall* [76], *Fin-YNet*, *S&P 1500* [77], *PPI* [78], and *Wiki-Rfa* [79]). Table 2.1 in the main text gives some summary statistics; here is a brief description of each data set.

• The Sampson monastery data [37] were collected by Sampson while resident at the monastery; the study spans 12 months. This data set contains relationships (esteem, liking, influence, praise, as well as disesteem, negative influence, and blame) between 25 novices in total, who were preparing to join a New England monastery. Each novice was asked to rank their top three choices for each of these relationships. Some novices gave ties for some of the choices, and nominated four instead of three other novices. the positive attributes have values 1, 2 and 3 in increasing order of affection, whereas the negative attributes take values -1, -2, and -3, in increasing order of dislike. The social relations were measured at five points in time. Some novices had left the monastery during this process; at time point 4, 18 novices are present. These novices possess as feature whether or not they attended the minor seminary of 'Cloisterville' before coming to the monastery. For the other 7 novices this information is not available. We combine these relationships into a network of 25 nodes by adding the weights for each relationship across all time points. Missing observations were set to 0. Based on his observations and analyses, Sampson divided the novices into four groups: Young Turks, Loyal Opposition, Outcasts, and an interstitial group; this division is taken as ground truth. We use as node (novice) attribute whether or not they attended 'Cloisterville' before coming to the monastery.

• *Rainfall* [76] contains 64,408 pairwise correlations between $n = 306$ locations in Australia, at which historical rainfalls have been measured.

• *Fin-YNet* consists of yearly correlation matrices for $n = 451$ stocks for 2000-2020 (21 distinct networks), using *market excess returns*. That is, we compute each correlation matrix from overnight (previous close to open) and intraday (open-to-close) price daily returns, from which we subtract the market return of the S&P500 index for the same time interval. In other words, within a given year, for each stock, we consider the time series of 500 market excess returns (there are 250 trading days within a year, and each day contributes with two returns, an overnight one and in intraday one). Each correlation network is built from the empirical correlation matrix of the entire set of stocks. For this data set, we report the results averaged over the 21 networks.

• *S&P1500* [77] considers daily prices for $n = 1,193$ stocks in the S&P 1500 Index, between 2003 and 2015, and builds correlation matrices from market excess returns (ie, from the return price of each financial instrument, the return of the market S&P500 is subtracted). Since we do not threshold, the result is thus a fully-connected weighted network, with stocks as nodes and correlations as edge weights.

• *PPI* [78] is a signed protein-protein interaction network between $n = 3,058$ proteins.

• *Wiki-Rfa* [79] is a signed network describing voting information for electing Wikipedia managers. Positive edges represent supporting votes, while negative edges represent opposing votes. We extract the largest connected component and remove nodes with degree at most one, resulting in $n = 7,634$ nodes for experiments.

A.3 Implementation Details

A.3.1 Efficient Algorithm for SIMPA

An efficient implementation of SIMPA is given in Algorithm 4. We omit the subscript $\mathcal{V}$ for ease of notation. The matrix operations described in Eq. equation 2.1 in the main text appear to be computationally expensive and space unfriendly. However, SSSNET resolves these concerns via an efficient sparsity-aware implementation without explicitly calculating the sets of powers, such as $\mathcal{A}^{s+,h}$. The algorithm also takes sparse matrices as input, and sparsity is maintained throughout. Therefore, for input feature dimension d_{in} and hidden dimension d, if $d' = \max(d_{\text{in}}, d) \ll n$, time and space complexity of SIMPA, and implicitly SSSNET, is $\mathcal{O}(|\mathcal{E}|d'h^2 + 4nd'K)$ and $\mathcal{O}(4|\mathcal{E}| + 10nd' + nK)$, respectively [72, 222]. When the network is large, SIMPA is amendable to a minibatch version using neighborhood sampling, similar to the minibatch forward propagation algorithm in [61, 223]. SIMPA is also amenable to an auto-scale version with theoretical guarantees, following [73].

Algorithm 4: Signed Mixed-Path Aggregation (SIMPA) algorithm for signed directed networks

Input : (Sparse) row-normalized adjacency matrices $\overline{\mathbf{A}}^{s+}, \overline{\mathbf{A}}^{s-}, \overline{\mathbf{A}}^{t+}, \overline{\mathbf{A}}^{t-}$; initial hidden representations $\mathbf{H}^{s+}, \mathbf{H}^{s-}, \mathbf{H}^{t+}, \mathbf{H}^{t-}$; hop h; lists of scalar weights $\Omega^{s+} = (\omega^{s+}_{\mathbf{M}}, \mathbf{M} \in \mathcal{A}^{s+,h}), \Omega^{s-} = (\omega^{s-}_{\mathbf{M}}, \mathbf{M} \in \mathcal{A}^{s-,h})$, $\Omega^{t+} = (\omega^{t+}_{\mathbf{M}}, \mathbf{M} \in \mathcal{A}^{t+,h}), \Omega^{t-} = (\omega^{t-}_{\mathbf{M}}, \mathbf{M} \in \mathcal{A}^{t-,h})$.

Output : Vector representations $\mathbf{z}_i$ for all $v_i \in \mathcal{V}$ given by $\mathbf{Z}$.

$\mathbf{Z}^{s+} \leftarrow \Omega^{s+}[0] \cdot \mathbf{H}^{s+}$; $\mathbf{Z}^{t+} \leftarrow \Omega^{t+}[0] \cdot \mathbf{H}^{t+}$; $\mathbf{Z}^{s-}, \mathbf{Z}^{t-} \leftarrow \mathbf{0}$;

$\tilde{\mathbf{X}}^{s+} \leftarrow \mathbf{H}^{s+}, \bar{\mathbf{X}}^{s-} \leftarrow \mathbf{H}^{s-}, \tilde{\mathbf{X}}^{t+} \leftarrow \mathbf{H}^{t+}, \bar{\mathbf{X}}^{t-} \leftarrow \mathbf{H}^{t-}$; $j \leftarrow 0$;

for $i \leftarrow 0$ **to** h **do**

  if $i > 0$ **then**

    $\tilde{\mathbf{X}}^{s+} \leftarrow \overline{\mathbf{A}}^{s+}\tilde{\mathbf{X}}^{s+}$; $\tilde{\mathbf{X}}^{t+} \leftarrow \overline{\mathbf{A}}^{t+}\tilde{\mathbf{X}}^{t+}$;

    $\mathbf{Z}^{s+} \leftarrow \mathbf{Z}^{s+} + \Omega^{s+}[i] \cdot \tilde{\mathbf{X}}^{s+}$; $\bar{\mathbf{X}}^{s-} \leftarrow \overline{\mathbf{A}}^{s+}\bar{\mathbf{X}}^{s-}$;

    $\mathbf{Z}^{t+} \leftarrow \mathbf{Z}^{t+} + \Omega^{t+}[i] \cdot \tilde{\mathbf{X}}^{t+}$; $\bar{\mathbf{X}}^{t-} \leftarrow \overline{\mathbf{A}}^{t+}\bar{\mathbf{X}}^{t-}$;

  end

  if $i \neq h$ **then**

    $\tilde{\mathbf{X}}^{s-} \leftarrow \overline{\mathbf{A}}^{s-}\bar{\mathbf{X}}^{s-}$; $\tilde{\mathbf{X}}^{t-} \leftarrow \overline{\mathbf{A}}^{t-}\tilde{\mathbf{X}}^{t-}$;

    $\mathbf{Z}^{s-} \leftarrow \mathbf{Z}^{s-} + \Omega^{s-}[j] \cdot \tilde{\mathbf{X}}^{s-}$;

    $\mathbf{Z}^{t-} \leftarrow \mathbf{Z}^{t-} + \Omega^{t-}[j] \cdot \tilde{\mathbf{X}}^{t-}$; $j \leftarrow j + 1$;

    for $k \leftarrow 0$ **to** $h - i - 2$ **do**

      $\tilde{\mathbf{X}}^{s-} \leftarrow \overline{\mathbf{A}}^{s+}\bar{\mathbf{X}}^{s-}$; $\tilde{\mathbf{X}}^{t-} \leftarrow \overline{\mathbf{A}}^{t+}\tilde{\mathbf{X}}^{t-}$;

      $\mathbf{Z}^{s-} \leftarrow \mathbf{Z}^{s-} + \Omega^{s-}[j] \cdot \tilde{\mathbf{X}}^{s-}$;

      $\mathbf{Z}^{t-} \leftarrow \mathbf{Z}^{t-} + \Omega^{t-}[j] \cdot \tilde{\mathbf{X}}^{t-}$; $j \leftarrow j + 1$;

    end

  end

end

$\mathbf{Z} = \text{CONCAT}\,(\mathbf{Z}^{s+}, \mathbf{Z}^{s-}, \mathbf{Z}^{t+}, \mathbf{Z}^{t-})$;

A.3.2 Machines

Experiments were conducted on a compute node with 4 Nvidia RTX 8000, 48 Intel Xeon Silver 4116 CPUs and 1000GB RAM, a compute node with 3 NVIDIA GeForce RTX 2080, 32 Intel Xeon E5-2690 v3 CPUs and 64GB RAM, a compute node with 2 NVIDIA Tesla K80, 16 Intel Xeon E5-2690 CPUs and 252GB RAM, and an Intel 2.90GHz i7-10700 processor with 8 cores and 16 threads. With the above, most experiments can be completed within a day.

A.3.3 Data Splits and Input

For each setting of synthetic data and real-world data, we first generate five different networks, each with two different data splits, then conduct experiments on them and report average performance over these 10 runs.

For synthetic data, 10% of all nodes are selected as test nodes for each cluster (the actual number is the ceiling of the total number of nodes times 0.1, so we would not fall below 10% of test nodes), 10% are selected as validation nodes (for model selection and early-stopping; again, we take the ceiling for the actual number), while the remaining roughly 80% are selected as training nodes (the actual number is bounded above by 80% since we take ceiling). For most real-world data sets, we extract the largest weak connected component for experiments. For Wiki-Rfa, we further rule out nodes that have degree less than two.

As for input features, we weigh the unit-length eigenvectors of the Signed Laplacian or regularized adjacency matrix by their eigenvalues introduced in [81]. For the Signed Laplacian features, we divide each eigenvector by its corresponding eigenvalue, since smaller eigenvectors are more likely to be informative. For regularized adjacency matrix features, we multiply eigenvalues by eigenvectors, since larger eigenvectors are more likely to be informative. After this scaling, there are no further standardization steps before inputting the features to our model. When features are available (in the case of *Sampson* data set), we standardize the one-dimensional binary input feature, so that the whole vector has mean zero and variance one.

For the sns and dns methods defined in Sec. 2.4.2 in the main text, we stack the eigenvectors associated with the smallest K eigenvalues of the corresponding Laplacians [63] to construct the feature matrix, then apply K-means to obtain the cluster assignments. For the other implementations, we also take the first K eigenvectors, either smallest or largest, following `https://github.com/alan-turing-institute/SigNet/blob/master/signet/cluster.py`.

A.3.4 Hyperparameters

We conduct hyperparmeter selection via a greedy search manner. To explain the details, consider for example the following synthetic data setting: polarized SSBMs with 1050 nodes, $n_c = 2$ SSBM communities, $\rho = 1.5, N = 200, p = 0.1$.

Recall that the the objective function minimizes

$$\mathcal{L} = \mathcal{L}_{\text{PBNC}} + \gamma_s(\mathcal{L}_{\text{CE}} + \gamma_t \mathcal{L}_{\text{triplet}}), \tag{A.1}$$

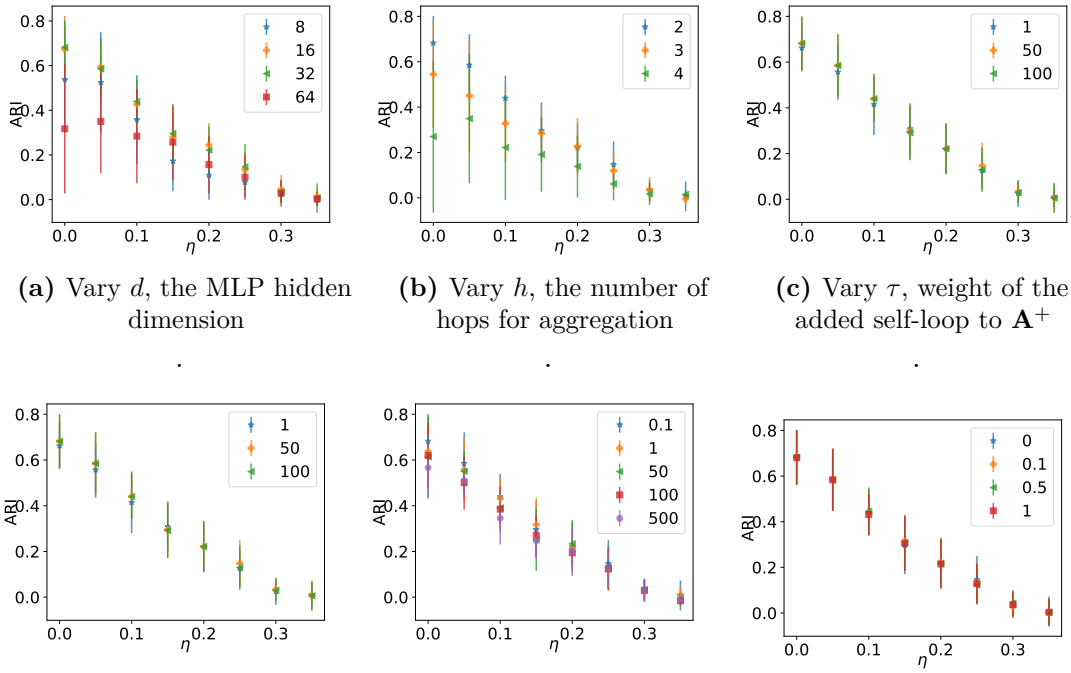
where $\gamma_s, \gamma_t > 0$ are weights for the supervised part of the loss and triplet loss within the supervised part, respectively.

Note that the cosine similarity (used in triplet loss $\mathcal{L}_{\text{triplet}}$) between two randomly picked vectors in d dimensions is bounded by $\sqrt{\ln(d)/d}$ with high probability. In our experiments $d = 32$, and $\sqrt{\ln(2d)/(2d)} \approx 0.25, \sqrt{\ln(4d)/(4d)} \approx 0.19$. In contrast, for fairly uniform clustering, the cross-entropy loss grows like $\log n$,

which in our experiments ranges between 3 and 17. Thus some balancing of the contribution is required.

Instead of a grid search, we tune hyperparameters according to what performs the best in the current default setting. If two settings give similar results, we pick the simpler setting, for example, the smaller hop size or the lower number of seed nodes.

When we reach a local optimum, we stop searching. Indeed, just a few iterations (less than five) were required for us to find the current setting, as SSSNET tends to be robust to most hyperparameters.



(a) Vary d, the MLP hidden dimension

(b) Vary h, the number of hops for aggregation

(c) Vary τ, weight of the added self-loop to $\mathbf{A}^+$

(d) Vary γ_s in Eq. equation 2.4 in main.

(e) Vary γ_t in Eq. equation 2.4 in main.

(f) Vary α in $\mathcal{L}_{\text{triplet}}$.

Figure A.3: Hyperpameter analysis on polarized SSBMs with $n = 1050$ nodes, $n_c = 2$ communites, $\rho = 1.5$, default community size $N = 200$, and $p = 0.1$.

Figure A.3 compares the performance of SSSNET on polarized SSBMs with 1050 nodes, $n_c = 2$ SSBM communities, $\rho = 1.5, N = 200, p = 0.1$, under different hyperparameter settings. By default, we use the loss function Eq. equation 2.4 in the main text with $\gamma_t = 0.1, \gamma_s = 50$, and $d = 32, l = 2, \tau = 0.5, h = 2, \alpha = 0$. We use the default seed ratio as 0.1 (the ratio of the number of seed nodes to the number of training nodes). We remark from (a) that as we increase the MLP hidden dimension d, performance first improves then decreases, with 32 a desirable value. As we increase the number of hops to consider, performance drops in (b). Therefore, we would use the simplest, yet best choice, hop 2. The decrease in both cases might be explained by too much noise introduced. For (c), the self-loop weight τ added to the positive part of the adjacency matrix does not seem to affect performance much, and hence we use 0.5 throughout. We conclude from (d) that it is recommended to have $\gamma_s > 1$ so as to take advantage of labeling information. The best triplet loss ratio in our candidates is $\gamma_t = 0.1$, based on (e). From (f), the influence of α is not evident, and hence we use $\alpha = 0$ throughout.

A.3.5 Training

For all synthetic data, we train SSSNET with a maximum of 300 epochs, and stop training when no gain in validation performance is achieved for 100 epochs (early-stopping).

For real-world data, when "ground-truth" labels are available, we still have separate test nodes. For S&P1500 data set, we do not have validation nodes, and 90% of all nodes are training nodes. For data sets with no "ground-truth" labels available, we train SSSNET in a self-supervised setting, using all nodes to train. We stop training when the training loss does not decrease for 100 epochs or when we reach the maximum number of epochs, 300.

For the two-layer MLP, we do not have a bias term for each layer, and we use a Rectified Linear Unit (ReLU), followed by a dropout layer with 0.5 dropout probability between the two layers, following [22]. We use Adam as the optimizer, and ℓ_2 regularization with weight decay $5 \cdot 10^{-4}$ to avoid overfitting.

B

DIGRAC: Digraph Clustering Based on Flow Imbalance Supplementary Information

B.1 Directed Mixed Path Aggregation (DIMPA)

To instantiate DIGRAC, we can employ any digraph aggregator that could generate the probability matrix $\mathbf{P}$. In this paper, we devise a simple yet effective directed mixed path aggregation scheme, to obtain the probability assignment matrix $\mathbf{P}$ and feed it to the loss function, as a special case of the successful *SSSNET* method introduced by [3]. Thus, in order to build node embeddings, we capture local network information by taking a weighted average of information from neighbors within h hops. To this end, we row-normalize the adjacency matrix, $\mathbf{A}$, to obtain $\overline{\mathbf{A}}^s$. Similar to the regularization discussed in [71], we add a weighted self-loop to each node and normalize by setting $\overline{\mathbf{A}}^s = (\tilde{\mathbf{D}}^s)^{-1}\tilde{\mathbf{A}}^s$, where $\tilde{\mathbf{A}}^s = \mathbf{A} + \tau\mathbf{I}$, with $\tilde{\mathbf{D}}^s$ the diagonal matrix with entries $\tilde{\mathbf{D}}^s(i,i) = \sum_j \tilde{\mathbf{A}}^s(i,j)$, and τ is a small value; we take $\tau = 0.5$; see Section B.3.4 for details.

The h-hop **source** matrix is given by $(\overline{\mathbf{A}}^s)^h$. We denote the set of *up-to-h-hop* source neighborhood matrices as $\mathcal{A}^{s,h} = \{\mathbf{I}, \overline{\mathbf{A}}^s, \ldots, (\overline{\mathbf{A}}^s)^h\}$. Similarly, for aggregating information when each node is viewed as a **target** node of a link, we carry out the same procedure for $\mathbf{A}^T$ which is the transpose of $\mathbf{A}$. We denote the set of up-to-h-hop target neighborhood matrices as $\mathcal{A}^{t,h} = \{\mathbf{I}, \overline{\mathbf{A}}^t, \ldots, (\overline{\mathbf{A}}^t)^h\}$, where $\overline{\mathbf{A}}^t$ is the row-normalized target adjacency matrix calculated from $\mathbf{A}^T$. As convention, the superscript s stands for *source* and the superscript t stands for *target*.

Next, we define two feature mapping functions for source and target embeddings, respectively. Assume that for each node in $\mathcal{V}$, a vector of features is available, and summarize these features in the input feature matrix $\mathbf{X}$. The source embedding is given by

$$\mathbf{Z}^s = \left(\sum_{\mathbf{M} \in \mathcal{A}^{s,h}} \omega^s_{\mathbf{M}} \cdot \mathbf{M} \right) \cdot \mathbf{H}^s \in \mathbb{R}^{n \times d}, \tag{B.1}$$

where for each $\mathbf{M}$, $\omega^s_{\mathbf{M}}$ is a learnable scalar, d is the dimension of this embedding, and $\mathbf{H}^s = \mathbf{MLP}^{(s,l)}(\mathbf{X})$. Here, the hyperparameter l controls the number of layers in the multilayer perceptron (MLP) with ReLU activation; we fix $l = 2$ throughout. Each layer of the MLP has the same number d of hidden units. The target embedding $\mathbf{Z}^t$ is defined similarly, with s replaced by t in Eq. (B.1). Different parameters for the MLPs for different embeddings are possible. After these two decoupled aggregations, we concatenate the embeddings to obtain the final node embedding as a $n \times (2d)$ matrix $\mathbf{Z} = \text{CONCAT}\left(\mathbf{Z}^s, \mathbf{Z}^t\right)$. The embedding vector $\mathbf{z}_i$ for a node v_i is the i^{th} row of $\mathbf{Z}$, $\mathbf{z}_i := (\mathbf{Z})_{(i,:)} \in \mathbb{R}^{2d}$.

After obtaining the embedding matrix $\mathbf{Z}$, we apply a linear layer (an affine transformation) to $\mathbf{Z}$, so that the resulting matrix has K columns. Next, we apply the unit *softmax* function to the rows and obtain the assignment probability matrix $\mathbf{P}$. Fig. B.1 gives an overview of this implementation.

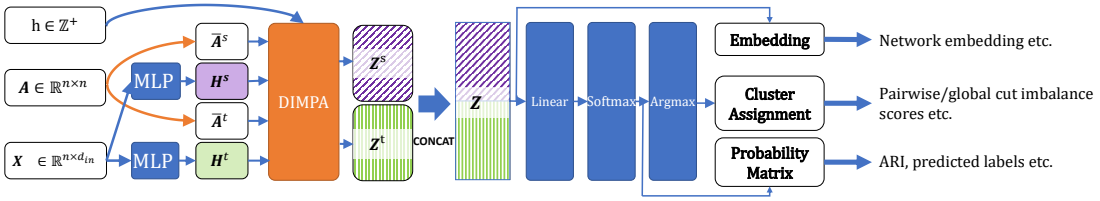


Figure B.1: DIGRAC with DIMPA as aggregator overview: from feature matrix $\mathbf{X}$ and adjacency matrix $\mathbf{A}$, we first compute the row-normalized adjacency matrices $\overline{\mathbf{A}}^s$ and $\overline{\mathbf{A}}^t$. Then, we apply two separate MLPs on $\mathbf{X}$, to obtain hidden representations $\mathbf{H}^s$ and $\mathbf{H}^t$. Next, we compute their decoupled embeddings using Eq. (B.1), and its equivalent for target embeddings. The concatenated decoupled embeddings are the final embeddings. For node clustering tasks, we add a linear layer followed by a unit *softmax* to obtain the probability matrix $\mathbf{P}$. Applying *argmax* on each row of $\mathbf{P}$ yields node cluster assignments.

To avoid computationally expensive and space unfriendly matrix operations, as described in Eq. B.1, DIGRAC uses an efficient sparsity-aware implementation, described in Algorithm 5, without explicitly calculating the sets of powers $\mathcal{A}^{s,h}$ and $\mathcal{A}^{t,h}$. We omit the subscript $\mathcal{V}$ for ease of notation. The algorithm is efficient in the sense that it takes sparse matrices as input, and never explicitly computes a multiplication of two $n \times n$ matrices. Therefore, for input feature dimension d_{in} and hidden dimension d, if $d' = \max(d_{\text{in}}, d) \ll n$, time and space complexity of DIMPA, and implicitly DIGRAC, is $\mathcal{O}(|\mathcal{E}|d'h^2 + 2nd'K)$ and $\mathcal{O}(2|\mathcal{E}| + 4nd' + nK)$, respectively [72, 222].

While it is a current shortcoming of DIGRAC that it does not scale well to very large networks, this limitation is shared by all the GNN competitors compared against in the paper, and some of the spectral methods. DIGRAC scales well in the sense that when the underlying network is sparse, the sparsity is preserved throughout the pipeline. In contrast, Bi_sym and DD_sym [111] construct derived dense matrices for manipulation, rendering the methods no longer scalable. These methods resulted in N/A values in Table 3.1 in the main text. For large-scale networks, DIMPA is amenable to a minibatch version using neighborhood sampling, similar to the minibatch forward propagation algorithm in [61, 223]. We are also aware of a framework [73] for scaling up graph neural networks automatically, where theoretical guarantees are provided, and ideas there will be exploited in future. We expect that the theoretical guarantees could be adapted to our situation.

Algorithm 5: Weighted Multi-Hop Neighbor Aggregation (DIMPA).

Input : (Sparse) row-normalized adjacency matrices $\overline{\mathbf{A}}^s, \overline{\mathbf{A}}^t$; initial hidden representations $\mathbf{H}^s, \mathbf{H}^t$; hop $h (h \geq 2)$; lists of scalar weights $\Omega^s = (\omega^s_{\mathbf{M}}, \mathbf{M} \in \mathcal{A}^{s,h}), \Omega^t = (\omega^t_{\mathbf{M}}, \mathbf{M} \in \mathcal{A}^{t,h})$.

Output : Vector representations $\mathbf{z}_i$ for all $v_i \in \mathcal{V}$ given by $\mathbf{Z}$.

$\tilde{\mathbf{X}}^s \leftarrow \overline{\mathbf{A}}^s \mathbf{H}^s$; $\tilde{\mathbf{X}}^t \leftarrow \overline{\mathbf{A}}^t \mathbf{H}^t$;

$\mathbf{Z}^s \leftarrow \Omega^s[0] \cdot \mathbf{H}^s + \Omega^s[1] \cdot \tilde{\mathbf{X}}^s$; $\mathbf{Z}^t \leftarrow \Omega^t[0] \cdot \mathbf{H}^t + \Omega^t[1] \cdot \tilde{\mathbf{X}}^t$;

for $i \leftarrow 2$ **to** h **do**

  $\tilde{\mathbf{X}}^s \leftarrow \overline{\mathbf{A}}^s \tilde{\mathbf{X}}^s$; $\tilde{\mathbf{X}}^t \leftarrow \overline{\mathbf{A}}^t \tilde{\mathbf{X}}^t$;

  $\mathbf{Z}^s \leftarrow \mathbf{Z}^s + \Omega^s[i] \cdot \tilde{\mathbf{X}}^s$; $\mathbf{Z}^t \leftarrow \mathbf{Z}^t + \Omega^t[i] \cdot \tilde{\mathbf{X}}^t$;

end

$\mathbf{Z} = \text{CONCAT}(\mathbf{Z}^s, \mathbf{Z}^t)$;

B.2 Loss and Objectives

B.2.1 Proof of Proposition 1

Moreover, we clarify that we make an assumption on the limiting behavior of the weights, namely that

$$\frac{\max_e |w_e|}{\sqrt{\sum_e w_e^2}} = o(m(k,l))$$

where $m(k,l)$ is the number of edges. This is a natural assumption: In the case that all weights are equal in absolute value, this assumption is satisfied as then $\frac{\max_e |w_e|}{\sqrt{\sum_e w_e^2}} = \frac{1}{\sqrt{m(k,l)}}$. The assumption is generally satisfied when there is not too much variability in the weights. If for example all but one weight pair was equal to 0, then the assumption would be violated, and also a normal approximation would not hold as there would only be two non-zero observations.

Proposition 4. *Suppose that $\mathcal{C}_k$ and $\mathcal{C}_l$ are two clusters of n_k and n_l nodes, respectively, with $m(k,l)$ edges between them, with symmetric edge weights $w_{ij} = w_{ji} \in [0,1]$ and with edge direction drawn independently at random with equal probability $\frac{1}{2}$ for each direction. We assume that the edge weights satisfy $\frac{\max_e |w_e|}{\sqrt{\sum_e w_e^2}} = o(m(k,l))$. Then $W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)$ is approximately normally distributed with mean 0 and variance $||w||^2$ as $m(k,l) \to \infty$.*

Proof. For each edge between the two clusters $\mathcal{C}_k$ and $\mathcal{C}_l$, the edge direction is random, i.e. the edge is from $\mathcal{C}_k$ to $\mathcal{C}_l$ with probability 0.5, and $\mathcal{C}_l$ to $\mathcal{C}_k$ with probability 0.5 also. Let $\mathcal{E}^{k,l}$ denote the set of $m(k,l) > 0$ edges between $\mathcal{C}_k$ and $\mathcal{C}_l$. For every edge $e \in \mathcal{E}^{k,l}$, the edge direction is encoded by a Rademacher random variable X_e with $X_e = 1$ if the edge is from $\mathcal{C}_k$ to $\mathcal{C}_l$, and $X_e = -1$ otherwise. Then $(X_e + 1)/2 \sim Ber(0.5)$ is a Bernoulli(0.5) random variable with mean $2 \times 0.5 - 1 = 0$ and variance $2^2 \times 0.5 \times (1 - 0.5) = 1$. We have the representation

$$W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k) = \sum_{e \in \mathcal{E}^{k,l}} X_e w_e$$

as the sum of $m(k,l)$ independent bounded random variables with finite third moments. Moreover, $W(\mathcal{C}_k,\mathcal{C}_l) - W(\mathcal{C}_l,\mathcal{C}_k)$ has mean 0 and variance $||w||^2$. The assertion now follows from a version of the Central Limit Theorem, Theorem 3.4 in [224]; we repeat the relevant part here:

Theorem 4 (Extract from Theorem 3.4 in [224])**.** *Let $\xi_1,\ldots,\xi_n$ be independent random variables with zero means satisfying $\sum_{i=1}^n Var(\xi_i) = 1$ and assume that there is a $\delta > 0$ such that $|\xi_i| \leq \delta$ for $1 \leq i \leq n$. Let Φ denote the cumulative distribution function of the standard normal distribution. Then*

$$\sup_{zin\mathbb{R}} \left| \mathbb{P}\left(\sum_{i=1}^n \xi_i \leq z) - \Phi(z) \right) \right| \leq 3.3\delta.$$

We apply this theorem with n replaced by $m(k,l)$, the number of edges, and take $\xi_e = \frac{X_e w_e}{\sqrt{\sum_e w_e^2}}$. Then ξ_e has mean zero and, using an enumeration of the edges, $\sum_{e=1}^{m(k,l)} \mathrm{Var}(\xi_e) = 1$. Moreover, $|\xi_e| \leq \frac{\max_e |w_e|}{\sqrt{\sum_e w_e^2}} =: \delta$ holds for all $e \in \{1,\ldots,m(k,l)\}$ and hence the theorem applies for the limit $m(k,l) \to \infty$. The stated result follows from using that if Z/σ has the standard normal distribution then Z has the mean zero normal distribution with variance σ^2.

□

B.2.2 Additional Details on Probabilistic Cut and Volume

Recall that the **probabilistic cut** from cluster $\mathcal{C}_k$ to $\mathcal{C}_l$ is defined as

$$W(\mathcal{C}_k,\mathcal{C}_l) = \sum_{i,j\in\{1,\ldots,n\}} \mathbf{A}_{i,j}\cdot\mathbf{P}_{i,k}\cdot\mathbf{P}_{j,l} = (\mathbf{P}_{(:,k)})^T\mathbf{A}\mathbf{P}_{(:,l)},$$

where $\mathbf{P}_{(:,k)}, \mathbf{P}_{(:,l)}$ denote the k^{th} and l^{th} columns of the assignment probability matrix $\mathbf{P}$, respectively. The **imbalance flow** between clusters $\mathcal{C}_k$ and $\mathcal{C}_l$ is defined as

$$|W(\mathcal{C}_k,\mathcal{C}_l) - W(\mathcal{C}_l,\mathcal{C}_k)|,$$

for $k,l \in \{0,\ldots,K-1\}$. The loss functions proposed in the main paper can be understood in terms of a probabilistic notion of degrees, as follows. We define the probabilistic out-degree of node v_i with respect to cluster k by $\tilde{d}_{i,k}^{(\text{out})} = \sum_{j=1}^n \mathbf{A}_{i,j}\cdot\mathbf{P}_{j,k} = (\mathbf{A}\mathbf{P}_{(:,k)})_i$, where subscript i refers to the i^{th} entry of the vector $\mathbf{A}\mathbf{P}_{(:,k)}$. Similarly, we define the probabilistic in-degree of node v_i with respect to cluster k by $\tilde{d}_{i,k}^{(\text{in})} = (\mathbf{A}^T\mathbf{P}_{(:,k)})_i$, where $\mathbf{A}^T$ is the transpose of $\mathbf{A}$. The **probabilistic degree** of node v_i with respect to cluster k is $\tilde{d}_{i,k} = \tilde{d}_{i,k}^{(\text{in})} + \tilde{d}_{i,k}^{(\text{out})} = ((\mathbf{A}^T+\mathbf{A})\mathbf{P}_{(:,k)})_i = \sum_{j=1}^n (\mathbf{A}_{i,j}+\mathbf{A}_{j,i})\cdot\mathbf{P}_{j,k}$.

For comparisons and ease of interpretation, it is advantageous to normalize the imbalance flow between clusters; for this purpose, we introduce the probabilistic volume of a cluster, as follows. The *probabilistic out-volume* for cluster $\mathcal{C}_k$ is defined as $VOL^{(\text{out})}(\mathcal{C}_k) = \sum_{i,j}\mathbf{A}_{j,i}\cdot\mathbf{P}_{j,k}$, and the *probabilistic in-volume* for cluster $\mathcal{C}_k$ is defined as $VOL^{(\text{in})}(\mathcal{C}_k)(\mathbf{A}^T\mathbf{P}_{(:,k)})_i$, where $\mathbf{A}^T$ is the transpose of $\mathbf{A}$. These volumes can be viewed as sum of probabilistic out-degrees and in-degrees, respectively; for example, $VOL^{(\text{in})}(\mathcal{C}_k) = \sum_{i=1}^n \tilde{d}_{i,k}^{(\text{in})}$. Then, it holds true that

$$VOL^{(\text{out})}(\mathcal{C}_k) = \sum_{i,j}\mathbf{A}_{i,j}\cdot\mathbf{P}_{i,k} \geq \sum_{i,j}\mathbf{A}_{i,j}\cdot\mathbf{P}_{i,k}\cdot\mathbf{P}_{j,l} = W(\mathcal{C}_k,\mathcal{C}_l), \tag{B.2}$$

since entries in $\mathbf{P}$ are probabilities, which are in $[0,1]$, and all entries of $\mathbf{A}$ are nonnegative. Similarly, $VOL^{(\text{in})}(\mathcal{C}_k) \geq W(\mathcal{C}_l, \mathcal{C}_k)$.

The **probabilistic volume** for cluster $\mathcal{C}_k$ is defined as

$$VOL(\mathcal{C}_k) = VOL^{(\text{out})}(\mathcal{C}_k) + VOL^{(\text{in})}(\mathcal{C}_k) = \sum_{i,j} (\mathbf{A}_{i,j} + \mathbf{A}_{j,i}) \cdot \mathbf{P}_{j,k}.$$

Then, it holds true that $VOL(\mathcal{C}_k) \geq W(\mathcal{C}_k, \mathcal{C}_l)$ for all $l \in \{0, \ldots, K-1\}$ and

$$\min(VOL(\mathcal{C}_k), VOL(\mathcal{C}_l)) \geq \max(W(\mathcal{C}_k, \mathcal{C}_l), W(\mathcal{C}_l, \mathcal{C}_k)) \geq |W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|. \tag{B.3}$$

When there exists a strong imbalance, then $|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)| \approx \max(W(\mathcal{C}_k, \mathcal{C}_l), W(\mathcal{C}_l, \mathcal{C}_k))$. As an extreme case, if $\mathbf{P}_{j,l} = 1$ for all nonnegative terms in the summations in Eq. (B.2), and $VOL^{(\text{in})}(\mathcal{C}_k) = 0$, then $|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)| = VOL(\mathcal{C}_k)$.

B.2.3 Variants of Normalization

Recall that the imbalance term involved in most of our experiments, named $\text{CI}^{\text{vol\_sum}}$, is defined as

$$\text{CI}^{\text{vol\_sum}}(k,l) = 2\frac{|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|}{VOL(\mathcal{C}_k) + VOL(\mathcal{C}_l)} \in [0,1]. \tag{B.4}$$

An alternative, which does not take volumes into account, is given by

$$\text{CI}^{\text{plain}}(k,l) = \left|\frac{W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)}{W(\mathcal{C}_k, \mathcal{C}_l) + W(\mathcal{C}_l, \mathcal{C}_k)}\right| = 2\left|\frac{W(\mathcal{C}_k, \mathcal{C}_l)}{W(\mathcal{C}_k, \mathcal{C}_l) + W(\mathcal{C}_l, \mathcal{C}_k)} - \frac{1}{2}\right| \in [0,1]. \tag{B.5}$$

We call this cut flow imbalance CI^{plain} as it does not penalize extremely unbalanced cluster sizes.

To achieve balanced cluster sizes and still constrain each imbalance term to be in $[0,1]$, one solution is to multiply the imbalance flow value by the minimum of $VOL(\mathcal{C}_k)$ and $VOL(\mathcal{C}_l)$, and then divide by $\max_{(k',l')\in\mathcal{T}}(\min(VOL(\mathcal{C}_{k'}), VOL(\mathcal{C}_{l'})))$, where $\mathcal{T} = \{(\mathcal{C}_k, \mathcal{C}_l) : 0 \leq k < l \leq K-1, k, l \in \mathbb{Z}\}$. The reason for using $\mathcal{T}$ is that $\text{CI}^{\text{plain}}(k,l)$ is symmetric with respect to k and l, and $\text{CI}^{\text{plain}}(k,l) = 0$ whenever $k = l$. Note that the maximum of the minimum here equals the second largest volume among clusters. We then obtain $\text{CI}^{\text{vol\_min}}$ as

$$\text{CI}^{\text{vol\_min}}(k,l) = \text{CI}^{\text{plain}}(k,l) \times \frac{\min(VOL(\mathcal{C}_k), VOL(\mathcal{C}_l))}{\max_{(k',l')\in\mathcal{T}}(\min(VOL(\mathcal{C}_{k'}), VOL(\mathcal{C}_{l'})))}. \tag{B.6}$$

Another potential choice, denoted $\text{CI}^{\text{vol\_max}}$, whose normalization follows from the same reasoning as $\text{CI}^{\text{vol\_sum}}$, is given by

$$\text{CI}^{\text{vol\_max}}(k,l) = \frac{|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|}{\max(VOL(\mathcal{C}_k), VOL(\mathcal{C}_l))} \in [0,1]. \tag{B.7}$$

Note that the current $\text{CI}^{\text{vol\_sum}}(k,l)$ term can be reformulated as

$$\text{CI}^{\text{vol\_sum}}(k,l) = 2\frac{|W(\mathcal{C}_k, \mathcal{C}_l) - W(\mathcal{C}_l, \mathcal{C}_k)|}{VOL(\mathcal{C}_k) + VOL(\mathcal{C}_l)} = 2\frac{W(\mathcal{C}_k, \mathcal{C}_l) + W(\mathcal{C}_l, \mathcal{C}_k)}{VOL(\mathcal{C}_k) + VOL(\mathcal{C}_l)} \times \text{CI}^{\text{plain}}(k,l), \tag{B.8}$$

with the first term in the decomposition corresponding to the relative ratio of inter- and intra-cluster edge density. For our synthetic data, this term is constant as we have constant edge density across the graph. However, for certain real-world data sets, one could also maximize this first term by increasing the inter-cluster density while decreasing the intra-cluster density, which seems to be a side effect. However, in our experiments, we also evaluate our results with different metrics, including objectives without any normalization, and conclude that this side effect does not create any issues in our data sets.

B.2.4 Selection of the Loss Function

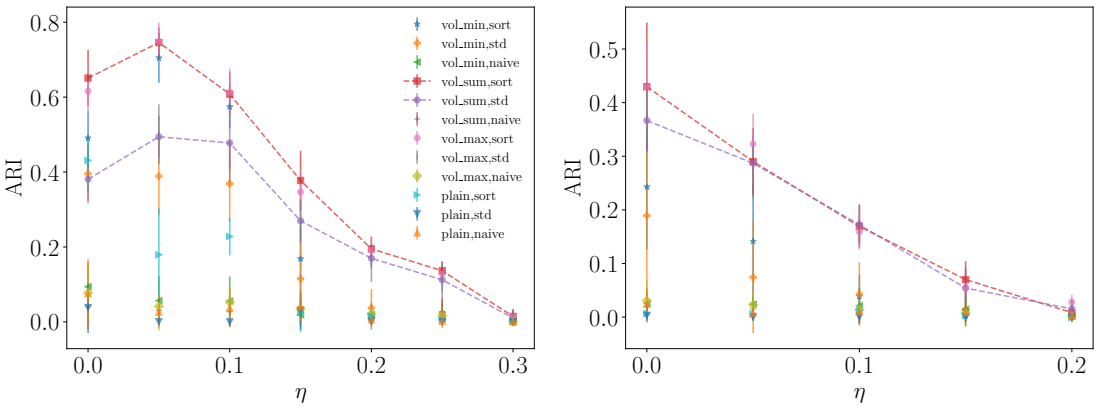
Table B.1: Naming conventions for objectives and loss functions

Selection variant / CI	$CI^{\text{vol\_sum}}$	$CI^{\text{vol\_min}}$	$CI^{\text{vol\_max}}$	CI^{plain}
sort	$\mathcal{O}^{\text{sort}}_{\text{vol\_sum}}, \mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$	$\mathcal{O}^{\text{sort}}_{\text{vol\_min}}, \mathcal{L}^{\text{sort}}_{\text{vol\_min}}$	$\mathcal{O}^{\text{sort}}_{\text{vol\_max}}, \mathcal{L}^{\text{sort}}_{\text{vol\_max}}$	$\mathcal{O}^{\text{sort}}_{\text{plain}}, \mathcal{L}^{\text{sort}}_{\text{plain}}$
std	$\mathcal{O}^{\text{std}}_{\text{vol\_sum}}, \mathcal{L}^{\text{std}}_{\text{vol\_sum}}$	$\mathcal{O}^{\text{std}}_{\text{vol\_min}}, \mathcal{L}^{\text{std}}_{\text{vol\_min}}$	$\mathcal{O}^{\text{std}}_{\text{vol\_max}}, \mathcal{L}^{\text{std}}_{\text{vol\_max}}$	$\mathcal{O}^{\text{std}}_{\text{plain}}, \mathcal{L}^{\text{std}}_{\text{plain}}$
naive	$\mathcal{O}^{\text{naive}}_{\text{vol\_sum}}, \mathcal{L}^{\text{naive}}_{\text{vol\_sum}}$	$\mathcal{O}^{\text{naive}}_{\text{vol\_min}}, \mathcal{L}^{\text{naive}}_{\text{vol\_min}}$	$\mathcal{O}^{\text{naive}}_{\text{vol\_max}}, \mathcal{L}^{\text{naive}}_{\text{vol\_max}}$	$\mathcal{O}^{\text{naive}}_{\text{plain}}, \mathcal{L}^{\text{naive}}_{\text{plain}}$

Table B.1 provides naming conventions of all the twelve pairs of variants of objectives and loss functions used in this paper. We select the loss functions for DIGRAC based on two representative models, and compare the performance of different loss functions. We use DIMPA (introduced in B.1) as an instantiation of DIGRAC's aggregator, for which $d = 32$, hidden units, $h = 2$ hops, and no seed nodes. Figures B.2(a) and B.3 compare twelve choices of loss combinations on a DSBM with $n = 1000$ nodes, $K = 5$ blocks, $\rho = 1, p = 0.02$ without ambient nodes, with a complete meta-graph structure. The subscript indicates the choice of pairwise imbalance, and the superscript indicates the variant for selecting pairs. Figures B.2(b) and B.4 are based on a DSBM with $n = 1000$ nodes, $K = 5$ blocks, $\rho = 1, p = 0.02$ without ambient nodes, with a cycle meta-graph structure. For these figures, dash lines highlight the "*sort*" variant as well as the "*std*" variant based on $CI^{\text{vol\_sum}}$, which have been introduced in the main text.

We also plot the imbalance evolution curves for the above two synthetic models when $\eta = 0.05$, for all the loss variants, in Figure B.5.

These figures indicate that the "*sort*" variant generally provides the best test ARI performance and the best overall global imbalance scores, among which using normalizations $\text{CI}^{\text{vol\_sum}}$ and $\text{CI}^{\text{vol\_max}}$ perform the best. The "*std*" variant is comparable with the "*sort*" variant in many instances, but is less stable in performance. We observe, however, from Figure B.5, that the "std" variants normally converge much faster. Taking the above into account, if we have prior knowledge of the network structure, or when we could conduct some prior analysis on the value β to take, the "*sort*" variant should be the variant of choice. Further, from Figure B.5, we observe that normalization in the loss function helps avoid the degenerate situation that the loss does not decrease. Such degeneracy can occur in the "plain" variants, raising issues about the practical usefulness of these variants. We observe that $\mathcal{L}^{\text{sort}}_{\text{vol\_min}}$ appears to behave worse than $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$ and $\mathcal{L}^{\text{sort}}_{\text{vol\_max}}$, even when using the "sort" variant to select pairwise imbalance scores. One possible explanation is that $\mathcal{L}^{\text{sort}}_{\text{vol\_min}}$ does not penalize extreme volume sizes, and that it takes minimum as well as maximum which, as functions of the data, are not as smooth as taking a summation. Throughout our experiments in the main text, we hence use the loss function $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$.



(a) DSBM("complete", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)

(b) DSBM("cycle", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)

Figure B.2: ARI comparison of loss functions on DSBM with 1000 nodes, 5 blocks, $\rho = 1, p = 0.02$ without ambient nodes, of cycle (left) and complete (right) meta-graph structures, respectively. The first component of the legend is the choice of pairwise imbalance, and the second component is the variant of selecting pairs. The naming conventions for the abbreviations in the legend are provided in Table B.1.

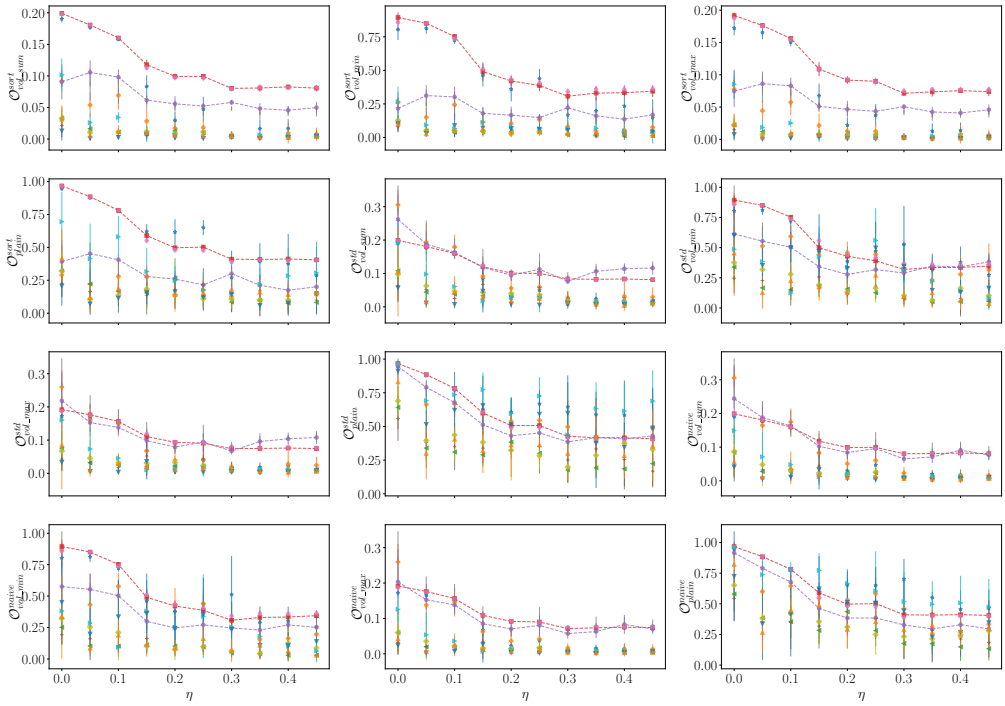


Figure B.3: Imbalance scores comparison of loss functions on DSBM with 1000 nodes, 5 blocks, $\rho = 1, p = 0.02$ without ambient nodes, of the **complete meta-graph** structure. The legend is the same as Fig. B.2(a).

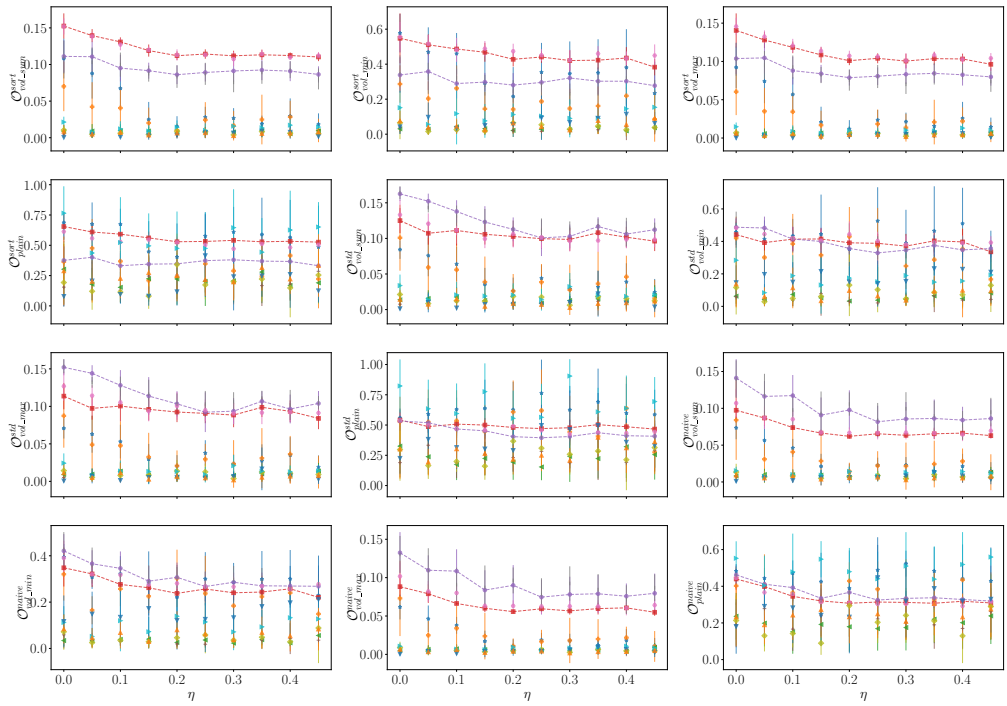


Figure B.4: Imbalance scores comparison of loss functions on DSBM with 1000 nodes, 5 blocks, $\rho = 1, p = 0.02$ without ambient nodes, of the **cyclic meta-graph** structure. The legend is the same as Fig. B.2(a).

B.3 Implementation Details

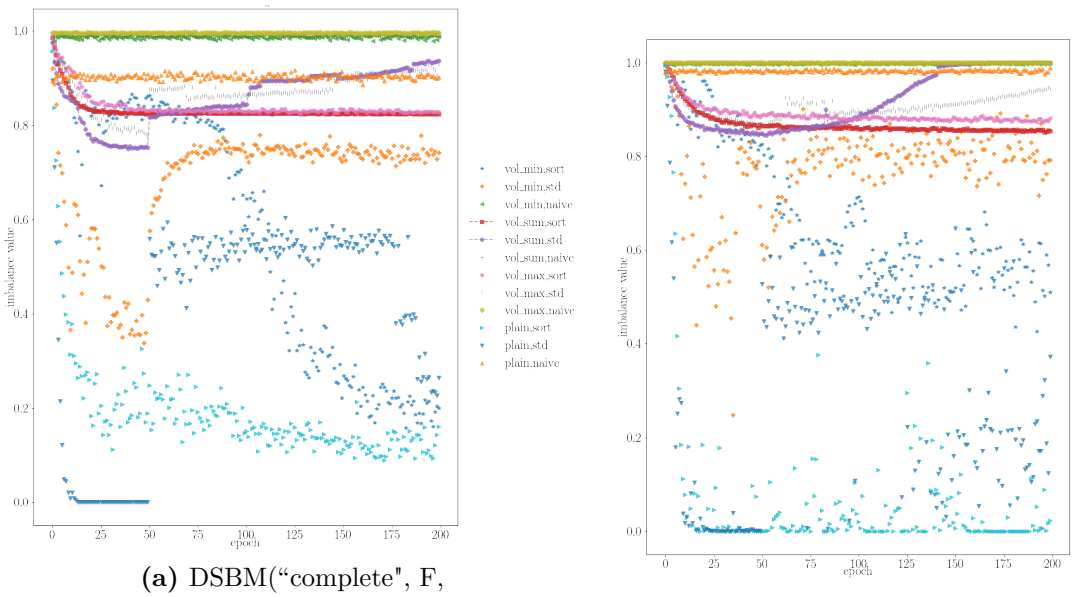
B.3.1 Code

To fully reproduce our results, code and preprocessed data are available at `https://github.com/SherylHYX/DIGRAC_Directed_Clustering`.

B.3.2 Hardware

Experiments were conducted on a compute node with 8 Nvidia RTX 8000, 48 Intel Xeon Silver 4116 CPUs and 1000GB RAM, a compute node with 4 NVIDIA GeForce RTX 2080, 32 Intel Xeon E5-2690 v3 CPUs and 64GB RAM, a compute node with 2 NVIDIA Tesla K80, 16 Intel Xeon E5-2690 CPUs and 252GB RAM, and an Intel 2.90GHz i7-10700 processor with 8 cores and 16 threads.

With this setup, all experiments for spectral methods, MagNet, DiGCL, and DIGRAC can be completed within two days, including repeated experiments, to obtain averages over multiple runs. DGCN, DiGCN, and MagNet have much longer run time (especially DGCN, which is space-consuming, and we cannot run many experiments in parallel), with a total of three days for them to finish. The slow speed stems from the competitor methods; some of the other GNN methods take a long time to run. Table 3.1 in the main text shows N/A values for Bi_sym and for DD_sym exactly for this reason. Empirically, DIGRAC is among the fastest among all GNN methods to which it is compared. In detail, Table B.2 reports the average runtime for all GNN methods on a variety of DSBM models, and illustrates that DIGRAC indeed takes the least or second least computational time per epoch. The results are averaged over 10 runs for the first 200 epochs. DiGCL is also efficient



(a) DSBM("complete", F, $n = 1000, K = 5, p = 0.02, \rho = 1, \eta = 0.05$)

(b) DSBM("cycle", F, $n = 1000, K = 5, p = 0.02, \rho = 1, \eta = 0.05$)

Figure B.5: Imbalance loss evolution comparison of loss functions on DSBM with 1000 nodes, 5 blocks, $\rho = 1, p = 0.02, \eta = 0.05$ without ambient nodes, of cycle (left) and complete (right) meta-graph structures, respectively. The first component of the legend is the choice of pairwise imbalance, and the second component is the variant of selecting pairs. The naming conventions for the abbreviations in the legend are provided in Table B.1.

in running time, but with worse performance than DIGRAC even as a supervised method, see the enlarged synthetic results in Sec. B.3.8 (Figure B.9). The total number of epochs required until the validation loss does not decrease for 200 epochs (or the maximum number of 1000 epochs is reached) varies for different data sets.

Table B.2: GNN average runtime (seconds per epoch) comparison. The results are averaged over 10 runs for the first 200 epochs. The fastest is highlighted in **bold red** while the second fastest is marked with underline blue .

Runtime (second per epoch on average)/GNN method	DiGCL	DGCN	DiGCN	MagNet	DIGRAC
DSBM("complete", T, $n = 1000, K = 5, p = 0.1, \rho = 1.5, \eta = 0.1$)	**0.107**	0.606	0.469	0.369	0.308
DSBM("path", F, $n = 1000, K = 5, p = 0.02, \rho = 1, \eta = 0.15$)	**0.061**	0.227	0.212	0.238	0.201
DSBM("star", F, $n = 1000, K = 5, p = 0.02, \rho = 1, \eta = 0.3$)	**0.095**	0.305	0.294	0.324	0.292
DSBM("star", F, $n = 5000, K = 5, p = 0.02, \rho = 1, \eta = 0.4$)	0.222	0.966	0.276	0.116	**0.101**
DSBM("cycle", F, $n = 5000, K = 5, p = 0.01, \rho = 1.5, \eta = 0$)	0.177	0.330	0.099	0.095	**0.089**
DSBM("cycle", F, $n = 30000, K = 5, p = 0.001, \rho = 1, \eta = 0$)	**0.070**	0.868	0.208	0.183	0.156

B.3.3 Data

Data Splits and Preprocessing

The results comparing DIGRAC with other methods on synthetic data are averaged over 50 runs, five synthetic networks under the same setting, each with 10 different

data splits. For synthetic data, 10% of all nodes are selected as test nodes for each cluster (the actual number is the ceiling of the total number of nodes times 0.1, to avoid falling below 10% of test nodes), 10% are selected as validation nodes (for model selection and early-stopping; again, we consider the ceiling for the actual number), while the remaining roughly 80% are selected as training nodes (the actual number can never be higher than 80% due to using the ceiling for both the test and validation splits). To further clarify the training setup, we use 0% of the labels in training. As DIGRAC is a self-supervised method, in principle, we could use all nodes for training. However, for a fair comparison with other GNN methods, we use only 80% of the nodes for training. For supervised methods our split of 80% - 10% - 10% is a standard split. For the non-GNN methods, all nodes are used for training.

For both synthetic and real-world data sets, we extract the largest weakly connected component for experiments, as our framework could be applied to different weakly connected components, if the digraph is disconnected. Isolated nodes do not include any imbalance information. As customary in community detection, they are often omitted in real networks. When "ground-truth" is given, test results are averaged over 10 different data splits on one network. When no labels are available, results are averaged over 10 different data splits.

Averaged results are reported with error bars representing one standard deviation in the figures, and plus/minus one standard deviation in the tables.

Synthetic Data

Our synthetic data, DSBM, which we denote by DSBM $(\mathcal{M}, \mathbb{1}(\text{ambient}), n, K, p, \rho, \eta)$, is built similarly to [97] but with possibly unequal cluster sizes: • (1) Assign cluster sizes $n_0 \leq n_1 \leq \cdots \leq n_{K-1}$ with size ratio $\rho \geq 1$, as follows. If $\rho = 1$ then the first $K-1$ clusters have the same size $\lfloor n/K \rfloor$ and the last cluster has size $n - (K-1)\lfloor n/K \rfloor$. If $\rho > 1$, we set $\rho_0 = \rho^{\frac{1}{K-1}}$. Solving $\sum_{i=0}^{K-1} \rho_0^i n_0 = n$ and taking integer value gives $n_0 = \left\lfloor n(1-\rho_0)/(1-\rho_0^K) \right\rfloor$. Further, set $n_i = \lfloor \rho_0 n_{i-1} \rfloor$, for $i = 1, \cdots, K-2$ if $K \geq 3$, and $n_{K-1} = n - \sum_{i=0}^{K-2} n_i$. Then the ratio of the size of the largest to the smallest cluster is approximately $\rho_0^{K-1} = \rho$. • (2) Assign each node randomly to one of K clusters, so that each cluster has the allocated size. • (3) For node $v_i, v_j \in \mathcal{C}_k$, independently sample an edge from node v_i to node v_j with probability $p \cdot \tilde{\mathbf{F}}_{k,k}$. • (4) For each pair of different clusters $\mathcal{C}_k, \mathcal{C}_l$ with $k \neq l$, for each node $v_i \in \mathcal{C}_k$, and each node $v_j \in \mathcal{C}_l$, independently sample an edge from node v_i to node v_j with probability $p \cdot \tilde{\mathbf{F}}_{k,l}$.

Real-World Data

For real-world data sets, we choose the number K of clusters in the meta-graph and the number β of edges between clusters in the meta-graph as follows. As they are needed as input for DIGRAC, we resort to Herm_rw [97] as an initial view of the network clustering. When a suitable meta-graph is suggested in a previous publication, then we use that choice. Otherwise, the number K of clusters is determined using the clustering from Herm_rw. First, we pick a range of K, and for each K, we calculate the global imbalance scores and plot the predicted meta-graph flow matrix $\mathbf{F}'$ based on the clustering from Herm_rw. Its entries are defined as

$$\mathbf{F}'(k,l) = \mathbb{1}(W(\mathcal{C}_k, \mathcal{C}_l) + W(\mathcal{C}_l, \mathcal{C}_k) > 0) \times \frac{W(\mathcal{C}_k, \mathcal{C}_l)}{W(\mathcal{C}_k, \mathcal{C}_l) + W(\mathcal{C}_l, \mathcal{C}_k)}. \tag{B.9}$$

These entries can be viewed as predicted probabilities of edge directions. Then, we choose K from this range so that the predicted meta-graph flow matrix has the high-

est imbalance scores and strong imbalance in the predicted meta-graph flow matrix.

The choice of β, which we assume should be equal to the number of edges in the meta-graph, is as follows. We plot the ranked pairs of $\mathrm{CI}^{\mathrm{plain}}$ values from Herm_rw and select the β which is at least as large as $K-2$, to allow the meta-graph to be connected, and which corresponds to a large drop in the plot.

Here we provide a brief description for each of the data sets; Table B.3 gives the number, n, of nodes, the number, $|\mathcal{E}|$, of directed edges, the number $|\mathcal{E}^r|$, of reciprocal edges (self-loops are counted once and for $u \neq v$, a reciprocal edge $u \to v, v \to u$ is counted twice) as well as their percentage among all edges, for the real-world networks, illustrating the variability in network size and density (defined as $|\mathcal{E}|/[n(n-1)]$).

• *Telegram* [4] is a pairwise influence network between $n = 245$ Telegram channels with $|\mathcal{E}| = 8,912$ directed edges. It is found in [4] that this network reveals a core-periphery structure in the sense of [90]. A directed core-periphery structure arises when there is a densely connected group of edges – a core – and sparsely connected groups of peripheral nodes with edges leading into the core, as well as sparsely connected groups of peripheral nodes with edges coming out of the core. Following [4] we assume $K = 4$ clusters, and the core-periphery structures gives $\beta = 5$.

• *Blog* [129] records $|\mathcal{E}| = 19,024$ directed edges between $n = 1,212$ political blogs from the 2004 US presidential election. In [129] it is found that there is an underlying structure with $K = 2$ clusters corresponding to the Republican and Democratic parties. Hence we choose $K = 2$ and $\beta = 1$.

• *Migration* [36] reports the number of people that migrated between pairs of counties in the US during 1995-2000. It involves $n = 3,075$ countries and $|\mathcal{E}| = 721,432$ directed edges after obtaining the largest weakly connected component. We choose $K = 10$ and $\beta = 9$, following [97]. Since the original digraph has extremely large entries, to cope with these outliers, we preprocess the input network by

$$\mathbf{A}_{i,j} = \frac{\mathbf{A}_{i,j}}{\mathbf{A}_{i,j} + \mathbf{A}_{j,i}} \mathbb{1}(\mathbf{A}_{i,j} > 0), \forall i, j \in \{1, \cdots, n\}, \tag{B.10}$$

which follows the preprocessing of [97]. The results for not doing this preprocessing is provided in Table B.11.

• *WikiTalk* [38] contains all users and discussion from the inception of Wikipedia until Jan. 2008. The $n = 2,388,953$ nodes in the network represent Wikipedia users and a directed edge from node v_i to node v_j denotes that user i edited at least once a talk page of user j. There are $|\mathcal{E}| = 5,018,445$ edges. We choose $K = 10$ clusters among candidates $\{2, 3, 5, 6, 8, 10\}$, and $\beta = 10$.

• *Lead-Lag* [105] contains yearly lead-lag matrices from 269 stocks from 2001 to 2019. We choose $K = 10$ clusters based on the GICS industry sectors [225], and choose $\beta = 3$ to emphasize the top three pairs of imbalance values. The lead-lag matrices are built from time series of daily price log returns, as detailed in [105]. The lead-lag metric for entry (i, j) in the network encodes a measure of the extent to which stock i leads stock j, and is obtained by applying a functional that computes the signed normalized area under the curve (auc) of the standard cross-correlation function (ccf). The resulting matrix is skew-symmetric, and entry (i, j) quantifies the extent to which stock i leads or lags stocks j, thus leading to a directed network interpretation. Starting from the skew-symmetric matrix, we further convert negative entries to zero, so that the resulting digraph can be directly fed into other methods; note that this step does not throw away any information, and is pursued only to render the representation of the digraph consistent with the format expected by all methods compared, including DIGRAC. Note that the statistics given in Table B.3 are averaged over the 19 years.

Table B.3: Summary statistics for the real-world networks.

data set	n	$\lvert\mathcal{E}\rvert$	density	weighted	$\lvert\mathcal{E}^r\rvert$	$\frac{\lvert\mathcal{E}^r\rvert}{\lvert\mathcal{E}\rvert}(\%)$
Telegram	245	8,912	$1.28 \cdot 10^{-2}$	True	1,572	17.64
Blog	1,222	19,024	$1.49 \cdot 10^{-1}$	True	4,617	24.27
Migration	3,075	721,432	$7.63 \cdot 10^{-2}$	True	351,100	48.67
WikiTalk	2,388,953	5,018,445	$8.79 \cdot 10^{-7}$	False	723,526	14.42
Lead-Lag	269	29,159	$4.04 \cdot 10^{-1}$	True	0.00	0.00

As input features, after obtaining eigenvectors from Hermitian matrices constructed as in [97], we standardize each column vector so that it has mean zero and variance one. We use these features for all GNN methods except MagNet, since MagNet has its own way of generating random features of dimension one.

B.3.4 Hyperparameter Selection for DIMPA

We conduct hyperparmeter selection via a greedy search, for DIGRAC implemented with DIMPA as its aggregator. To explain the details, consider for example the following synthetic data setting: DSBM with 1000 nodes, 5 clusters, $\rho = 1$, and $p = 0.02$, without ambient nodes under different hyperparameter settings. By default, we use the loss function $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$, $d = 32$ hidden units, hop $h = 2$, and no seed nodes. Instead of a grid search, we tune hyperparameters according to what performs the best in the default setting of the respective GNN method. The procedure starts with a random setting. For the next iteration, the hyperparameters are set to the current best setting (based on the last iteration), independently. For example, if we start with $a = 1, b = 2, c = 3$, and we find that under this default setting, the best a (when fixing $b = 2, c = 3$) is 2 and the best b (when fixing $a = 1, c = 3$) is 3, and the best c is 3 (when fixing $a = 1, b = 2$), then for the next iteration, we set $a = 2, b = 3, c = 3$. If two settings give similar results, we choose the simpler setting, for example, the smaller hop size. When we reach a local optimum, we stop searching. Indeed, just a few iterations (less than five) were required for us to find the current setting, as DIGRAC tends to be robust to most hyperparameters.

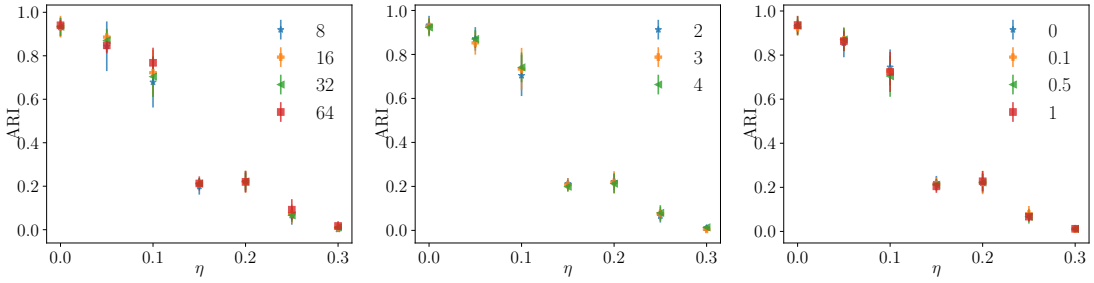
Fig. B.6, B.7 and B.8 are plots corresponding to the same setting but for three different meta-graph structures, namely the complete meta-graph structure, the cycle structure but with ambient nodes, and the complete structure with ambient nodes, respectively.

In theory, more hidden units give better expressive power. To reduce complexity, we use 32 hidden units throughout, which seems to have desirable performance. We observe that for low-noise regimes, more hidden units actually hurt performance. We can draw a similar conclusion about the hyperparameter selection. In terms of τ, DIGRAC seems to be robust to different choices. Therefore, we use $\tau = 0.5$ throughout.

B.3.5 Use of Seed Nodes in a Semi-Supervised Manner

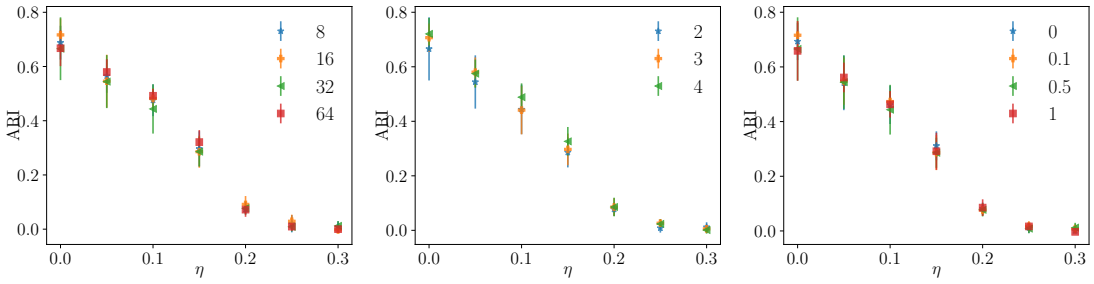
Supervised Loss

For seed nodes in $\mathcal{V}^{\text{seed}}$, similar to the loss function in [3], we use as a supervised loss function the sum of a cross-entropy loss and a triplet loss. The cross-entropy



(a) Vary d. **(b)** Vary the hop h. **(c)** Vary τ.

Figure B.6: Hyperparameter analysis on different hyperparameter settings on the **complete** DSBM with 1000 nodes, 5 clusters, $\rho = 1$, and $p = 0.02$ **without** ambient nodes.



(a) Vary d. **(b)** Vary the hop h. **(c)** Vary τ.

Figure B.7: Hyperparameter analysis on different hyperparameter settings on the **complete** DSBM with 1000 nodes, 5 clusters, $\rho = 1$, and $p = 0.02$ **with** ambient nodes.

loss is given by

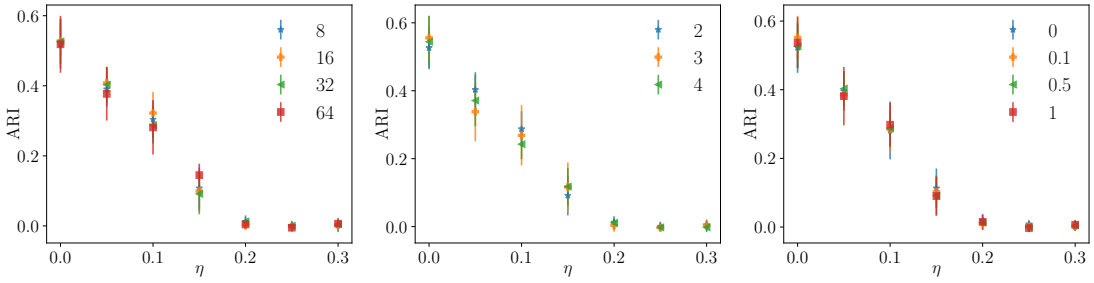
$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\mathcal{V}^{\text{seed}}|} \sum_{v_i \in \mathcal{V}^{\text{seed}}} \sum_{k=1}^{K} \mathbb{1}(v_i \in \mathcal{C}_k) \log\left((\mathbf{p}_i)_k\right), \tag{B.11}$$

where $\mathbb{1}$ is the indicator function, $\mathcal{C}_k$ denotes the k^{th} cluster, and $(\mathbf{p}_i)_k$ denotes the k^{th} entry of probability vector $(\mathbf{p}_i)$. With the function $L : \mathbb{R}^2 \to \mathbb{R}$ given by $L(x, y) = [x - y]_+$ (where the subscript $+$ indicates taking the maximum of the expression value and 0), the triplet loss is defined as

$$\mathcal{L}_{\text{triplet}} = \frac{1}{|\mathcal{S}|} \sum_{(v_i, v_j, v_k) \in \mathcal{S}} L(\text{CS}(\mathbf{z}_i, \mathbf{z}_j), \text{CS}(\mathbf{z}_i, \mathbf{z}_k), \tag{B.12}$$

where $\mathcal{S} \subseteq \mathcal{V}^{\text{seed}} \times \mathcal{V}^{\text{seed}} \times \mathcal{V}^{\text{seed}}$ is a set of node triplets: v_i is an anchor seed node, and v_j is a seed node from the same cluster as the anchor, while v_k is from a different cluster; and $\text{CS}(\mathbf{z}_i, \mathbf{z}_j)$ is the cosine similarity of the embeddings of nodes v_i and v_j. We choose cosine similarity so as to avoid sensitivity to the magnitude of the embeddings. The triplet loss is designed so that, given two seed nodes from the same cluster and one seed node from a different cluster, the respective embeddings of the pairs from different clusters should be farther away than the embedding of the pair within the same cluster.

We then consider the weighted sum $\mathcal{L}_{\text{CE}} + \gamma_t \mathcal{L}_{\text{triplet}}$ as the supervised part of the loss function for DIGRAC, for some parameter $\gamma_t > 0$. The parameter γ_t



(a) Vary d. **(b)** Vary the hop h. **(c)** Vary τ.

Figure B.8: Hyperparameter analysis on different hyperparameter settings on the **cycle** DSBM with 1000 nodes, 5 clusters, $\rho = 1$, and $p = 0.02$ **with** ambient nodes.

arises as follows. The cosine similarity between two randomly picked vectors in d dimensions is bounded by $\sqrt{\ln(d)/d}$ with high probability. In our experiments $d = 32$, and $\sqrt{\ln(2d)/(2d)} \approx 0.25$. In contrast, for fairly uniform clustering, the cross-entropy loss grows like $\log n$, which in our experiments ranges between 3 and 17. Thus some balancing of the contribution is required. Following [3], we choose $\gamma_t = 0.1$ in our experiments.

Overall Objective Function

By combining Eq. (B.11), Eq. (B.12), and Eq. (3.3)), our objective function for semi-supervised training with known seed nodes minimizes

$$\mathcal{L} = \mathcal{L}^{\text{sort}}_{\text{vol\_sum}} + \gamma_s(\mathcal{L}_{\text{CE}} + \gamma_t \mathcal{L}_{\text{triplet}}), \tag{B.13}$$

where $\gamma_s, \gamma_t > 0$ are weights for the supervised part of the loss and triplet loss within the supervised part, respectively. We set $\gamma_s = 50$ as we want our model to perform well on seed nodes. The weights could be tuned depending on how important each term is perceived to be.

B.3.6 Training

For all synthetic data, we train DIGRAC with a maximum of 1000 epochs, and stop training when no gain in validation performance is achieved for 200 epochs (early-stopping). For real-world data, no “ground-truth" labels are available; we use all nodes to train and stop training when the training loss does not decrease for 200 epochs, or when we reach the maximum number of epochs, 1000.

When using the “std" variant for training, for the initial 50 epochs, we apply the “sort" variant with $\beta = 3$ for a reasonable starting clustering probability matrix for training, as otherwise during the initial training epochs possibly no pairs could be picked out. During the epochs actually utilizing this “std" variant, if no pairs could be picked out, we temporarily switch to the “naive" variant to count all the pairs for that epoch.

For the two-layer MLP, we do not have a bias term for each layer, and we use Rectified Linear Unit (ReLU) followed by a dropout layer with 0.5 dropout probability between the two layers, following [3]. We use Adam [226] as the optimizer and ℓ_2 regularization with weight decay $5 \cdot 10^{-4}$ to avoid overfitting. We use as learning rate 0.01 throughout.

B.3.7 Implementation Details for the Comparison Methods

In our experiments, we compare DIGRAC against five spectral methods, InfoMap, and four GNN-based supervised methods on synthetic data, and spectral methods and InfoMap on real data. The reason we are not able to compare DIGRAC with the other GNNs (namely, DGCN, DiGCN, MagNet, and DiGCL) on these data sets is due to the fact that these data sets do not have labels, which are required by the other GNN methods. We use the same hyperparameter settings stated in these papers. Data splits for all models are the same; the comparison GNNs are trained with 80% nodes under label supervision.

For MagNet, we use $q = 0.25$ for the phase matrix as in [6], because it is mentioned that $q = 0.25$ lays the most emphasis on directionality, which is our main focus in this paper. Code for MagNet is from `https://github.com/matthew-hirn/magnet`. For DiGCN, we use the code from `https://github.com/flyingtango/DiGCN/blob/main/code/digcn_ib.py` with option "adj_type" equals "ib". As a recommended option in [114], we use three layers for DiGCN. All other settings are the same as in the original paper [114]. Code for DiGCL is from `https://github.com/flyingtango/DiGCL`, where we adopt the settings for Cora_ML for hyperparameters.

B.3.8 Enlarged Synthetic Result Figures

Figure B.9 enlarges the results in the main text on synthetic data, with the same conclusions to be drawn.

B.3.9 NMI Results Example and Reasons against Using NMI

As NMI is an often used measure for assessing similarities between partitions, Fig. B.10 provides NMI results on some synthetic models mentioned in the main text. The results are qualitatively similar to the ARI results in Fig. 3.3.

We do not use NMI in the main text to evaluate results as NMI is known to suffer from finite size effects [127, 227]. In particular NMI prefers a larger number of partitions. Moreover it has been observed that the NMI between two independent partitions can be much larger than zero. This feature makes NMI more difficult to interpret than for example ARI.

B.4 Additional Results on Real-World Data

B.4.1 Extended Result Tables

Tables B.4, B.5, B.6 and B.7 provide a detailed comparison of DIGRAC with spectral methods and InfoMap. Since no labeling information is available and all of the other competing GNN methods require labels, we do not compare DIGRAC with them on these real data sets.

In Tables B.4, B.5, B.6 and B.7, we report 12 combinations of global imbalance scores by data set. The naming convention of these imbalance scores is provided in Table B.1. To assess how balanced our recovered clusters are in terms of sizes, we also report the size ratio, which is defined as the size of the largest predicted cluster to the smallest one, and the standard deviation of sizes, size std, in order to show how varied the sizes of predicted clusters are. For a relatively balanced clustering, we expect the latter two terms to be small.

Table B.4: Performance comparison on *Telegram*. The best is marked in **bold red** and the second best is marked in underline blue .

Metric/Method	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.04±0.00	0.21±0.00	0.21±0.00	0.21±0.01	0.20±0.01	0.14±0.00	**0.32±0.01**
$\mathcal{O}^{sort}_{vol\_min}$	0.47±0.00	0.67±0.00	0.61±0.00	0.66±0.02	0.66±0.02	0.19±0.00	**0.79±0.06**
$\mathcal{O}^{sort}_{vol\_max}$	0.03±0.00	0.20±0.00	0.20±0.00	0.20±0.01	0.19±0.01	0.12±0.00	**0.29±0.01**
$\mathcal{O}^{sort}_{plain}$	**1.00±0.00**	0.80±0.00	0.75±0.00	0.78±0.03	0.76±0.04	0.59±0.00	0.96±0.01
$\mathcal{O}^{std}_{vol\_sum}$	0.01±0.00	0.26±0.00	0.26±0.00	0.26±0.01	0.25±0.02	**0.35±0.00**	0.28±0.01
$\mathcal{O}^{std}_{vol\_min}$	0.16±0.00	**0.84±0.00**	0.76±0.00	0.82±0.03	0.82±0.03	0.49±0.00	0.73±0.03
$\mathcal{O}^{std}_{vol\_max}$	0.01±0.00	0.25±0.00	0.25±0.00	0.25±0.01	0.24±0.02	**0.29±0.00**	0.25±0.01
$\mathcal{O}^{std}_{plain}$	0.68±0.00	**1.00±0.00**	0.94±0.00	0.98±0.04	0.95±0.04	0.99±0.00	0.90±0.05
$\mathcal{O}^{naive}_{vol\_sum}$	0.01±0.00	0.26±0.00	0.26±0.00	0.26±0.01	0.25±0.02	0.23±0.00	**0.27±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.11±0.00	**0.84±0.00**	0.76±0.00	0.82±0.03	0.82±0.03	0.32±0.00	0.72±0.04
$\mathcal{O}^{naive}_{vol\_max}$	0.00±0.00	**0.25±0.00**	**0.25±0.00**	**0.25±0.01**	0.24±0.02	0.20±0.00	0.24±0.01
$\mathcal{O}^{naive}_{plain}$	0.63±0.00	**1.00±0.00**	0.94±0.00	0.98±0.04	0.95±0.04	0.99±0.00	0.89±0.06
size ratio	24.750	242.000	242.000	242.000	242.00	53	**3.090**
size std	35.57	104.360	104.360	104.360	104.360	63.460	**26.39**

Table B.5: Performance comparison on *Blog*. The best is marked in **bold red** and the second best is marked in underline blue .

Metric/Method	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.07±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
$\mathcal{O}^{sort}_{vol\_min}$	0.02±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{sort}_{vol\_max}$	0.05±0.00	0.05±0.00	0.00±0.00	0.04±0.00	0.26±0.00	0.00±0.00	**0.40±0.00**
$\mathcal{O}^{sort}_{plain}$	**1.00±0.00**	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	0.89±0.00	0.76±0.00
$\mathcal{O}^{std}_{vol\_sum}$	0.00±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
$\mathcal{O}^{std}_{vol\_min}$	0.00±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{std}_{vol\_max}$	0.00±0.00	0.05±0.00	0.00±0.00	0.04±0.00	0.26±0.00	0.00±0.00	**0.40±0.00**
$\mathcal{O}^{std}_{plain}$	0.73±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{naive}_{vol\_sum}$	0.00±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
$\mathcal{O}^{naive}_{vol\_min}$	0.00±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{naive}_{vol\_max}$	0.00±0.00	0.05±0.00	0.00±0.00	0.04±0.00	0.26±0.00	0.00±0.00	**0.40±0.00**
$\mathcal{O}^{naive}_{plain}$	0.76±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
size ratio	**1.270**	8.700	2.450	6.100	11.93	44.26	1.860
size std	**64.50**	485	256.200	439	516.500	584	183.20

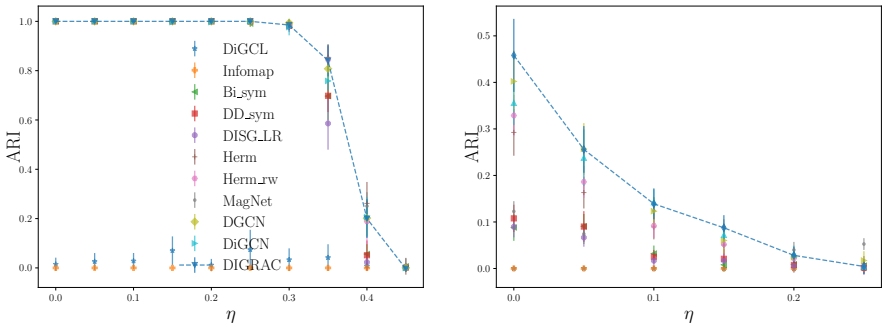
Tables B.4, B.5, B.6, B.7, B.8, B.9 and B.10 reveal that DIGRAC provides competitive global imbalance scores in all of the 12 objectives introduced, and across all the real data sets, usually outperforming all the other methods. Among the tables, Table B.10 provides results in terms of the distance to the best yearly performance, averaged across the 19 years; DIGRAC usually outperforms all the other methods across all the years. Note that Bi_sym and DD_sym are not able to generate results for *WikiTalk*, as large $n \times n$ matrix multiplication with its transpose causes memory issue, when $n = 2,388,953$. Small values of the size ratio and size standard deviation suggest that the normalization in the loss function penalizes tiny clusters, and that DIGRAC tends to predict balanced cluster sizes.

Table B.6: Performance comparison on *Migration*. The best is marked in **bold red** and the second best is marked in underline blue . InfoMap results are omitted here as it predicts a single huge cluster and could not generate imbalance results.

Metric/Method	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.03±0.00	0.01±0.00	0.02±0.00	0.04±0.00	0.02±0.00	**0.05±0.00**
$\mathcal{O}^{sort}_{vol\_min}$	**0.19±0.00**	0.08±0.00	0.08±0.00	0.15±0.02	0.05±0.00	0.18±0.03
$\mathcal{O}^{sort}_{vol\_max}$	0.03±0.00	0.01±0.00	0.01±0.00	0.03±0.00	0.02±0.00	**0.04±0.00**
$\mathcal{O}^{sort}_{plain}$	0.24±0.00	0.20±0.00	0.17±0.00	0.40±0.01	**0.49±0.06**	0.29±0.04
$\mathcal{O}^{std}_{vol\_sum}$	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.02±0.00	**0.04±0.01**
$\mathcal{O}^{std}_{vol\_min}$	0.10±0.00	0.05±0.00	0.05±0.00	0.08±0.01	0.04±0.00	**0.16±0.03**
$\mathcal{O}^{std}_{vol\_max}$	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	**0.03±0.01**
$\mathcal{O}^{std}_{plain}$	0.13±0.00	0.12±0.00	0.11±0.00	0.20±0.01	0.20±0.01	**0.26±0.01**
$\mathcal{O}^{naive}_{vol\_sum}$	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.01±0.00	**0.04±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.09±0.00	0.04±0.00	0.04±0.00	0.08±0.01	0.01±0.00	**0.16±0.03**
$\mathcal{O}^{naive}_{vol\_max}$	0.01±0.00	0.00±0.00	0.01±0.00	0.01±0.00	0.00±0.00	**0.03±0.01**
$\mathcal{O}^{naive}_{plain}$	0.12±0.00	0.10±0.00	0.08±0.00	0.19±0.00	0.19±0.03	**0.26±0.01**
size ratio	7.780	6.070	**4.360**	36.05	1035.90	4.420
size std	135.210	132.76	**103.43**	335.790	353.060	264.500

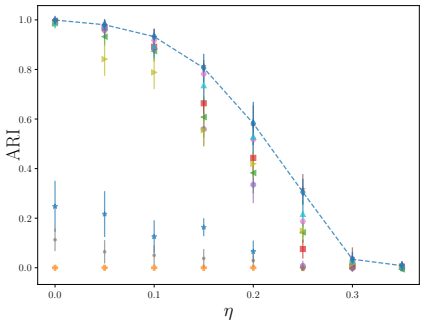
Table B.7: Performance comparison on *WikiTalk*. The best is marked in **bold red** and the second best is marked in underline blue . InfoMap results are omitted here as its large number of predicted clusters leads to memory error in imbalance calculation.

Metric/Method	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.18±0.03	0.15±0.02	0.00±0.00	**0.24±0.05**
$\mathcal{O}^{sort}_{vol\_min}$	0.10±0.03	0.22±0.05	0.26±0.00	**0.28±0.13**
$\mathcal{O}^{sort}_{vol\_max}$	0.16±0.03	0.09±0.01	0.00±0.00	**0.19±0.04**
$\mathcal{O}^{sort}_{plain}$	0.87±0.08	0.99±0.01	0.98±0.00	**1.00±0.00**
$\mathcal{O}^{std}_{vol\_sum}$	**0.17±0.04**	0.06±0.01	0.01±0.00	0.14±0.02
$\mathcal{O}^{std}_{vol\_min}$	0.09±0.02	0.09±0.02	**0.27±0.00**	0.18±0.08
$\mathcal{O}^{std}_{vol\_max}$	**0.15±0.04**	0.04±0.00	0.00±0.00	0.11±0.02
$\mathcal{O}^{std}_{plain}$	0.72±0.03	0.70±0.05	**0.98±0.00**	0.84±0.06
$\mathcal{O}^{naive}_{vol\_sum}$	0.10±0.02	0.04±0.00	0.00±0.00	**0.12±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.06±0.03	0.07±0.02	**0.26±0.00**	0.15±0.07
$\mathcal{O}^{naive}_{vol\_max}$	**0.09±0.02**	0.03±0.00	0.00±0.00	**0.09±0.01**
$\mathcal{O}^{naive}_{plain}$	0.64±0.04	0.61±0.04	**0.98±0.00**	0.76±0.06
size ratio	1190162.25	2217434.50	**250.48**	71765.14
size std	713813.72	660060.33	657941.88	**643220.37**

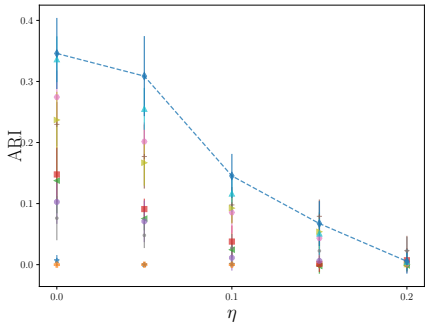


(a) DSBM(“cycle", F, $n = 1000, K = 3, p = 0.1, \rho = 1$)

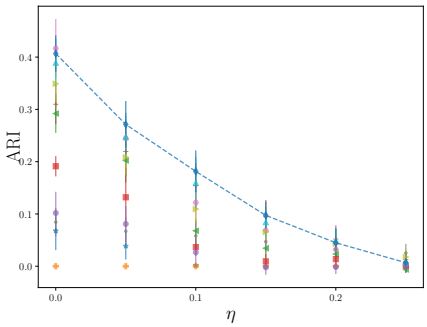
(b) DSBM(“complete", T, $n = 1000, K = 10, p = 0.02, \rho = 1$)



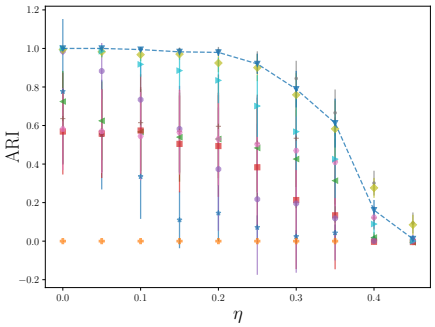
(c) DSBM(“cycle", F, $n = 1000, K = 3, p = 0.02, \rho = 1.5$)



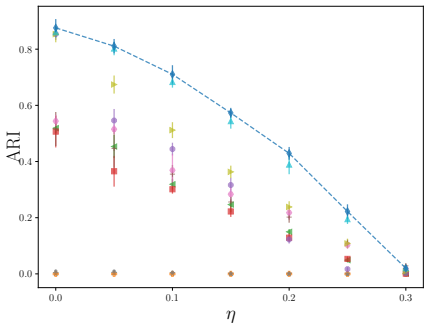
(d) DSBM(“path", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)



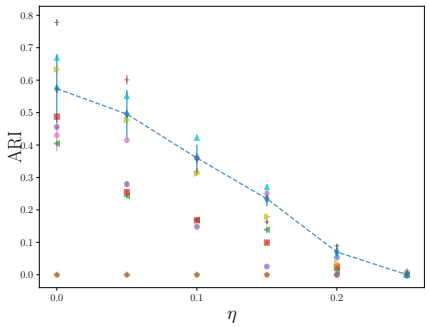
(e) DSBM(“star", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)



(f) DSBM(“complete", F, $n = 1000, K = 3, p = 0.1, \rho = 1$)

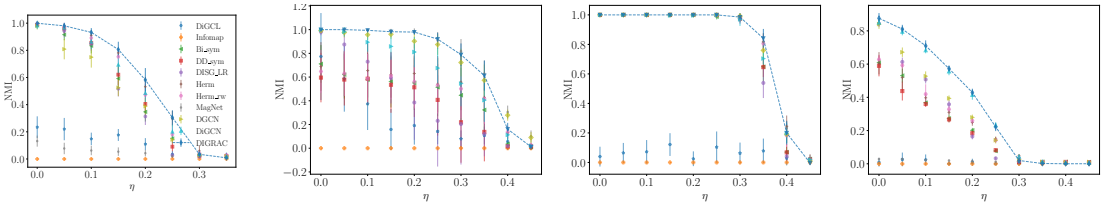


(g) DSBM(“cycle", T, $n = 5000, K = 5, p = 0.01, \rho = 1.5$)



(h) DSBM(“cycle", F, $n = 30000, K = 5, p = 0.001, \rho = 1$)

Figure B.9: Test ARI comparison on synthetic data. Dashed lines highlight DIGRAC's performance. Error bars are given by one standard error.



(a) DSBM("cycle", F, $n = 1000, K = 3, p = 0.02, \rho = 1.5$)

(b) DSBM("complete", F, $n = 1000, K = 3, p = 0.1, \rho = 1$)

(c) DSBM("cycle", F, $n = 1000, K = 3, p = 0.1, \rho = 1$)

(d) DSBM("cycle", T, $n = 5000, K = 5, p = 0.01, \rho = 1.5$)

Figure B.10: Test NMI comparison on some synthetic data. Dashed lines highlight DIGRAC's performance. Error bars are given by one standard error.

Table B.8: Performance comparison on *Lead-Lag* for year 2015. The best is marked in **bold red** and the second best is marked in underline blue . InfoMap results are omitted here as it usually predicts a single huge cluster and could not generate imbalance results.

Metric/Method	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{\text{sort}}_{\text{vol\_sum}}$	0.07±0.00	0.07±0.00	0.06±0.00	0.07±0.00	0.06±0.01	**0.15±0.00**
$\mathcal{O}^{\text{sort}}_{\text{vol\_min}}$	**0.53±0.06**	0.50±0.02	0.45±0.07	0.50±0.03	0.46±0.06	0.50±0.02
$\mathcal{O}^{\text{sort}}_{\text{vol\_max}}$	0.07±0.00	0.06±0.00	0.06±0.00	0.06±0.00	0.06±0.00	**0.15±0.01**
$\mathcal{O}^{\text{sort}}_{\text{plain}}$	0.65±0.03	**0.67±0.03**	0.59±0.03	0.65±0.03	0.65±0.02	0.55±0.07
$\mathcal{O}^{\text{std}}_{\text{vol\_sum}}$	0.04±0.00	0.04±0.00	0.04±0.00	0.04±0.00	0.04±0.00	**0.11±0.02**
$\mathcal{O}^{\text{std}}_{\text{vol\_min}}$	0.27±0.03	0.27±0.02	0.24±0.02	0.27±0.02	0.26±0.04	**0.35±0.04**
$\mathcal{O}^{\text{std}}_{\text{vol\_max}}$	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	**0.10±0.02**
$\mathcal{O}^{\text{std}}_{\text{plain}}$	0.39±0.02	0.39±0.01	0.37±0.02	0.39±0.02	**0.40±0.02**	0.38±0.04
$\mathcal{O}^{\text{naive}}_{\text{vol\_sum}}$	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	**0.08±0.03**
$\mathcal{O}^{\text{naive}}_{\text{vol\_min}}$	0.20±0.02	0.20±0.02	0.17±0.03	0.20±0.02	0.20±0.03	**0.25±0.08**
$\mathcal{O}^{\text{naive}}_{\text{vol\_max}}$	0.02±0.00	0.03±0.00	0.02±0.00	0.03±0.00	0.03±0.00	**0.08±0.03**
$\mathcal{O}^{\text{naive}}_{\text{plain}}$	0.29±0.01	0.29±0.01	0.26±0.02	0.30±0.01	0.30±0.01	**0.31±0.05**
size ratio	3.070	3.110	3.060	**2.89**	2.95	15.640
size std	8.390	7.94	8.680	**7.28**	8.050	18.680

Table B.9: Performance comparison on *Lead-Lag*. Results in each year is averaged over ten runs. Mean and standard deviation (after ±) are calculated over the 19 years. The best is marked in **bold red** and the second best is marked in underline blue. InfoMap results are omitted here as it usually predicts a single huge cluster and could not generate imbalance results.

Metric/Method	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.07±0.01	0.07±0.01	0.07±0.01	0.07±0.02	0.07±0.02	**0.15±0.03**
$\mathcal{O}^{sort}_{vol\_min}$	**0.51±0.10**	0.48±0.09	0.47±0.10	**0.51±0.11**	0.50±0.10	0.47±0.09
$\mathcal{O}^{sort}_{vol\_max}$	0.07±0.01	0.06±0.01	0.06±0.01	0.07±0.01	0.07±0.01	**0.14±0.03**
$\mathcal{O}^{sort}_{plain}$	**0.66±0.09**	0.64±0.08	0.63±0.08	**0.66±0.09**	0.65±0.09	0.53±0.09
$\mathcal{O}^{std}_{vol\_sum}$	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	**0.12±0.03**
$\mathcal{O}^{std}_{vol\_min}$	0.27±0.04	0.27±0.04	0.25±0.04	0.27±0.03	0.27±0.03	**0.38±0.07**
$\mathcal{O}^{std}_{vol\_max}$	0.04±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	**0.11±0.02**
$\mathcal{O}^{std}_{plain}$	0.40±0.05	0.39±0.05	0.38±0.05	0.40±0.05	0.40±0.05	**0.44±0.07**
$\mathcal{O}^{naive}_{vol\_sum}$	0.03±0.01	0.03±0.01	0.03±0.01	0.03±0.01	0.03±0.01	**0.08±0.04**
$\mathcal{O}^{naive}_{vol\_min}$	0.20±0.05	0.19±0.05	0.18±0.05	0.19±0.04	0.19±0.04	**0.26±0.10**
$\mathcal{O}^{naive}_{vol\_max}$	0.03±0.01	0.02±0.01	0.02±0.01	0.02±0.00	0.02±0.00	**0.08±0.03**
$\mathcal{O}^{naive}_{plain}$	0.30±0.06	0.28±0.06	0.27±0.06	0.29±0.05	0.29±0.05	**0.32±0.11**
size ratio	3.67	**3.34**	3.900	4.110	3.880	8.070
size std	9.31	**9.14**	10.090	10.490	10.360	17.060

Table B.10: Performance comparison on *Lead-Lag*, where we evaluate the performance distance to the best one in each year. Results in each year is averaged over ten runs. Mean and standard deviation (after ±) are calculated over the 19 years. The best is marked in **bold red** and the second best is marked in underline blue. InfoMap results are omitted here as it usually predicts a single huge cluster and could not generate imbalance results.

Metric/Method	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.07±0.02	0.08±0.02	0.08±0.02	0.07±0.02	0.07±0.02	**0.00±0.00**
$\mathcal{O}^{sort}_{vol\_min}$	**0.01±0.01**	0.05±0.03	0.06±0.03	0.02±0.02	0.02±0.02	0.06±0.04
$\mathcal{O}^{sort}_{vol\_max}$	0.07±0.02	0.07±0.02	0.07±0.02	0.07±0.02	0.07±0.02	**0.00±0.00**
$\mathcal{O}^{sort}_{plain}$	**0.01±0.02**	0.03±0.03	0.05±0.03	**0.01±0.02**	0.02±0.02	0.14±0.03
$\mathcal{O}^{std}_{vol\_sum}$	0.08±0.02	0.08±0.02	0.08±0.02	0.08±0.02	0.08±0.02	**0.00±0.00**
$\mathcal{O}^{std}_{vol\_min}$	0.10±0.05	0.11±0.04	0.13±0.05	0.11±0.05	0.11±0.05	**0.00±0.00**
$\mathcal{O}^{std}_{vol\_max}$	0.07±0.02	0.08±0.02	0.08±0.02	0.08±0.02	0.08±0.02	**0.00±0.00**
$\mathcal{O}^{std}_{plain}$	0.04±0.03	0.05±0.04	0.06±0.04	0.04±0.04	0.04±0.03	**0.00±0.00**
$\mathcal{O}^{naive}_{vol\_sum}$	0.05±0.03	0.06±0.03	0.06±0.03	0.05±0.03	0.05±0.03	**0.00±0.00**
$\mathcal{O}^{naive}_{vol\_min}$	0.06±0.07	0.07±0.06	0.08±0.07	0.07±0.08	0.07±0.08	**0.00±0.00**
$\mathcal{O}^{naive}_{vol\_max}$	0.05±0.03	0.05±0.03	0.05±0.03	0.05±0.03	0.05±0.03	**0.00±0.00**
$\mathcal{O}^{naive}_{plain}$	0.03±0.06	0.05±0.05	0.06±0.06	0.04±0.06	0.04±0.06	**0.01±0.02**
size ratio	1.04	**0.71**	1.270	1.480	1.250	5.440
size std	0.58	**0.41**	1.360	1.770	1.630	8.340

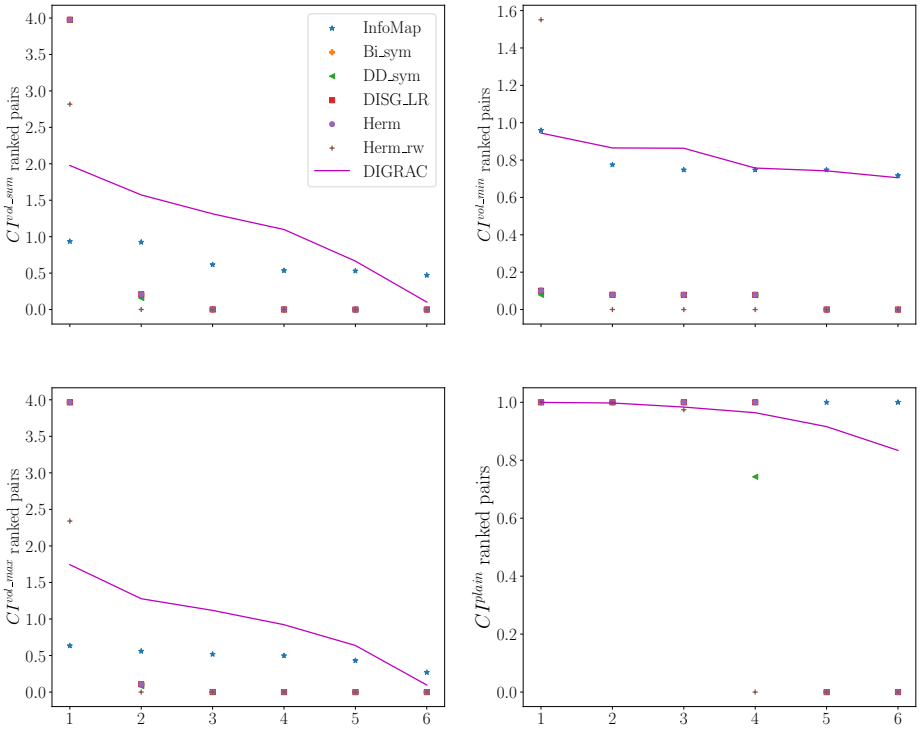


Figure B.11: Ranked pairs of pairwise imbalance recovered by comparing methods for different choices of normalization on the *Telegram* data set. Lines are used to highlight DIGRAC's performance.

B.4.2 Ranked Pairwise Imbalance Scores

We also plot the ranked pairwise imbalance scores for all data sets except *Blog*, which has only one possible pairwise imbalance score. For *Lead-Lag*, we only plot the year 2015 as an example; the plots for the other years are similar. Figures B.11, B.12, B.13 and B.14 illustrate that DIGRAC is able to provide comparable or higher pairwise imbalance scores for the leading pairs, especially on $\mathrm{CI}^{\mathbf{vol\_min}}$ pairs. We also observe that except for $\mathrm{CI}^{\mathbf{plain}}$, DIGRAC has a less rapid drop in pairwise imbalance scores after the first leading pair compared to Herm and Herm_rw, which can have a few pairs with higher imbalance scores than DIGRAC.

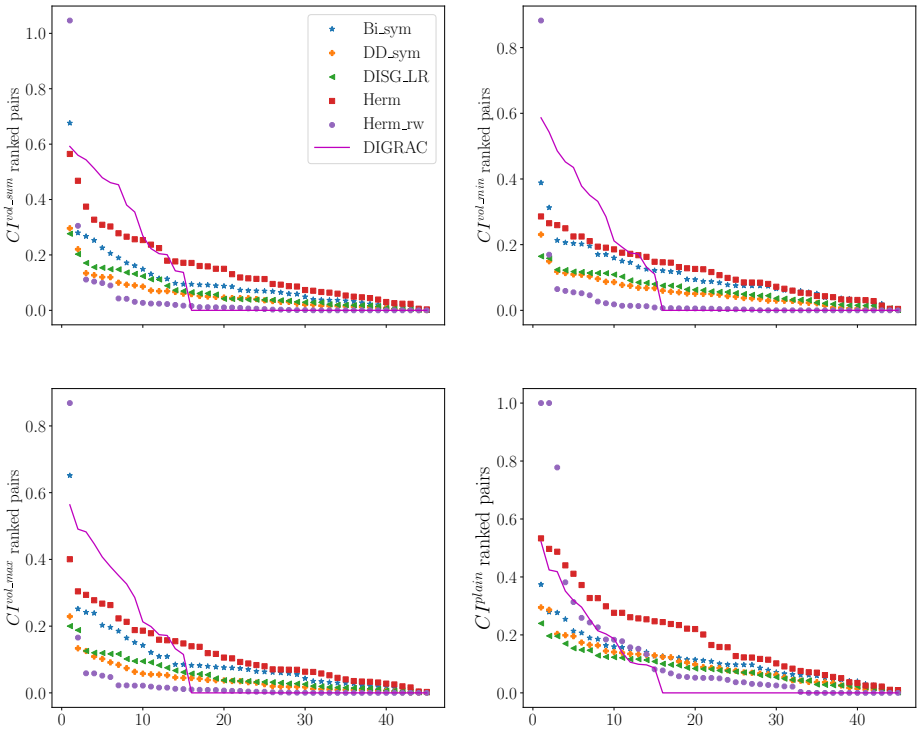


Figure B.12: Ranked pairs of pairwise imbalance recovered by comparing methods for different choices of normalization on the *Migration* data set. Lines are used to highlight DIGRAC's performance. InfoMap results are omitted as it predicts one single huge cluster and could not produce imbalance results.

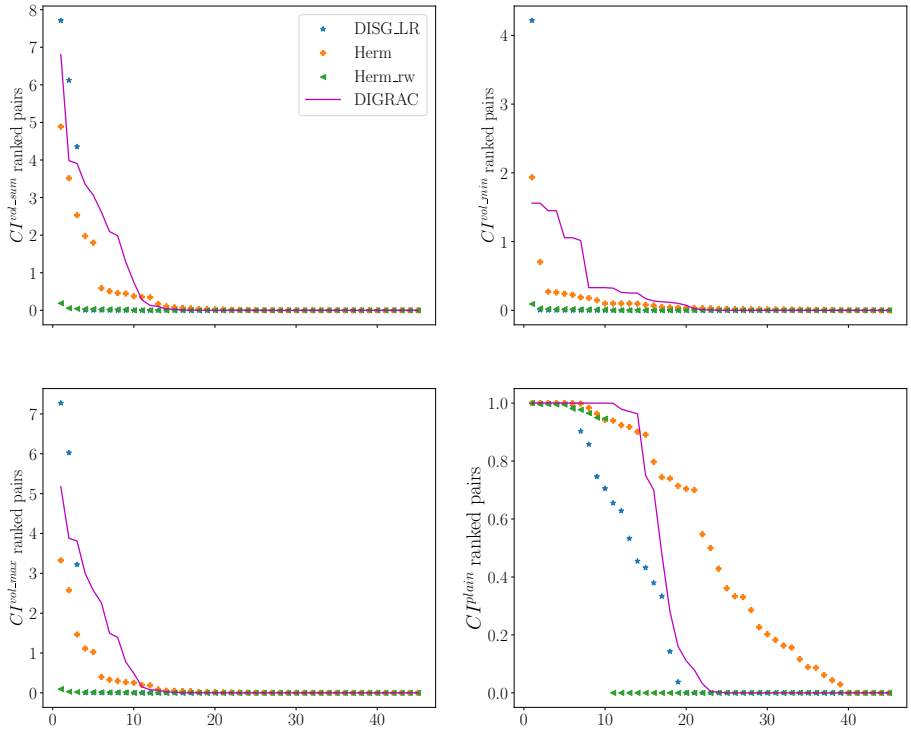


Figure B.13: Ranked pairs of pairwise imbalance recovered by comparing methods for different choices of normalization on *WikiTalk* data set. Lines are used to highlight DIGRAC's performance. InfoMap results are omitted here because it triggers memory error due to the large number of predicted clusters.

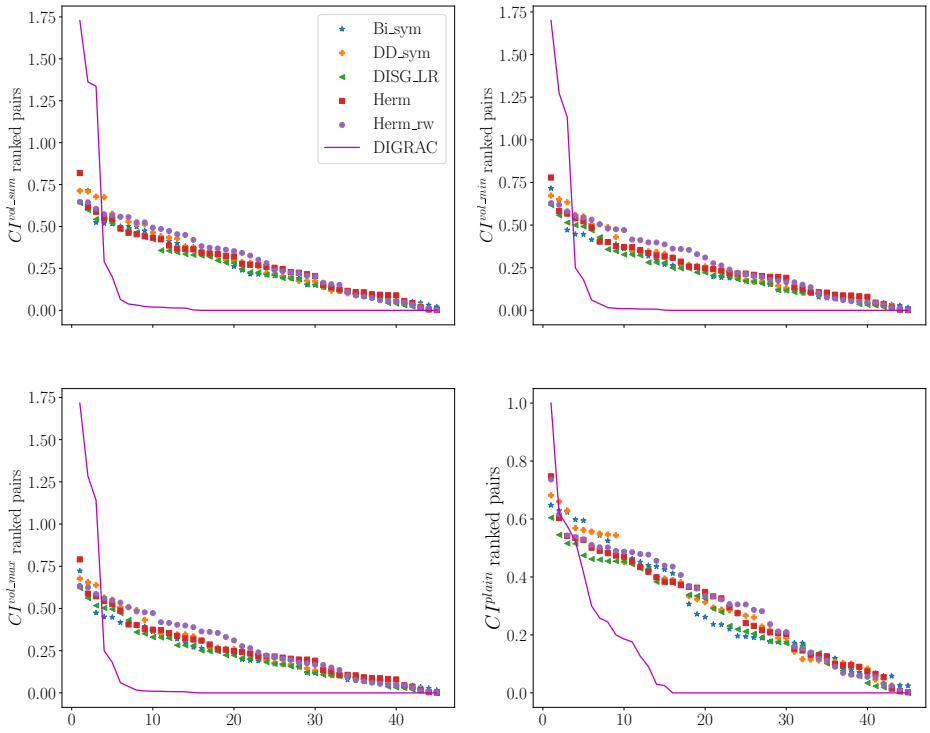
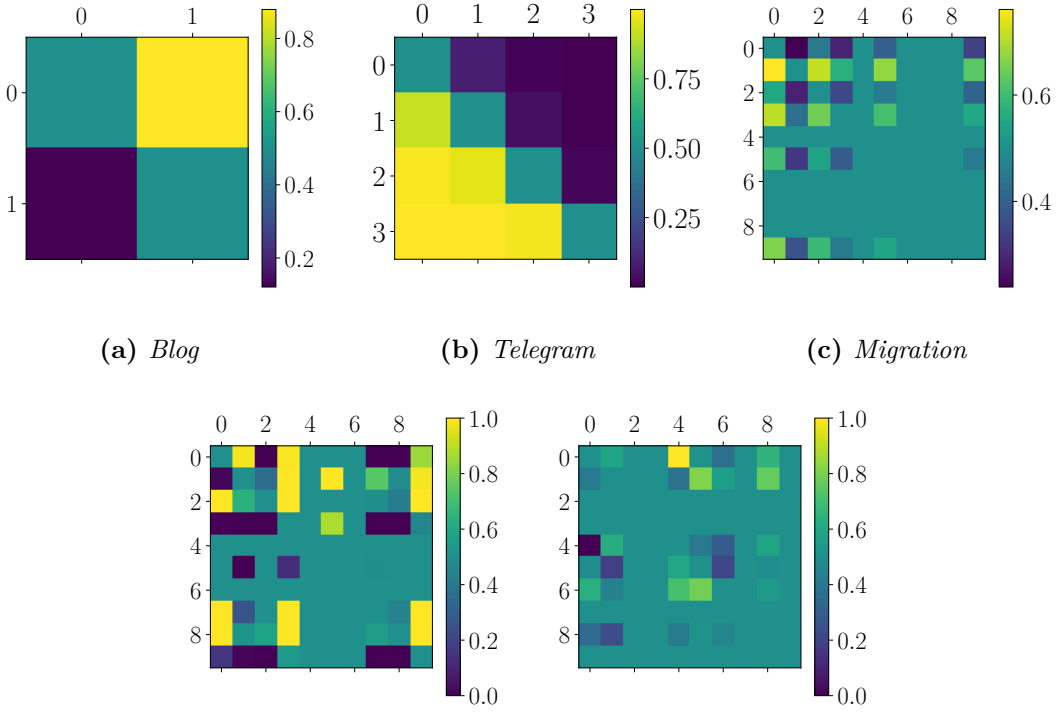


Figure B.14: Ranked pairs of pairwise imbalance recovered by comparing methods for different choices of normalization on *Lead-Lag* data set. Lines are used to highlight DIGRAC's performance. InfoMap results are omitted here because it only predicts a single cluster.

B.4.3 Predicted Meta-Graph Flow Matrix Plots

For each data set, we plot the predicted meta-graph flow matrix $\mathbf{F}'$ defined in Eq. (B.9).



(a) *Blog* **(b)** *Telegram* **(c)** *Migration*

(d) *WikiTalk* **(e)** *Lead-Lag* 2015

Figure B.15: Predicted meta-graph flow matrix from DIGRAC of five real-world data sets.

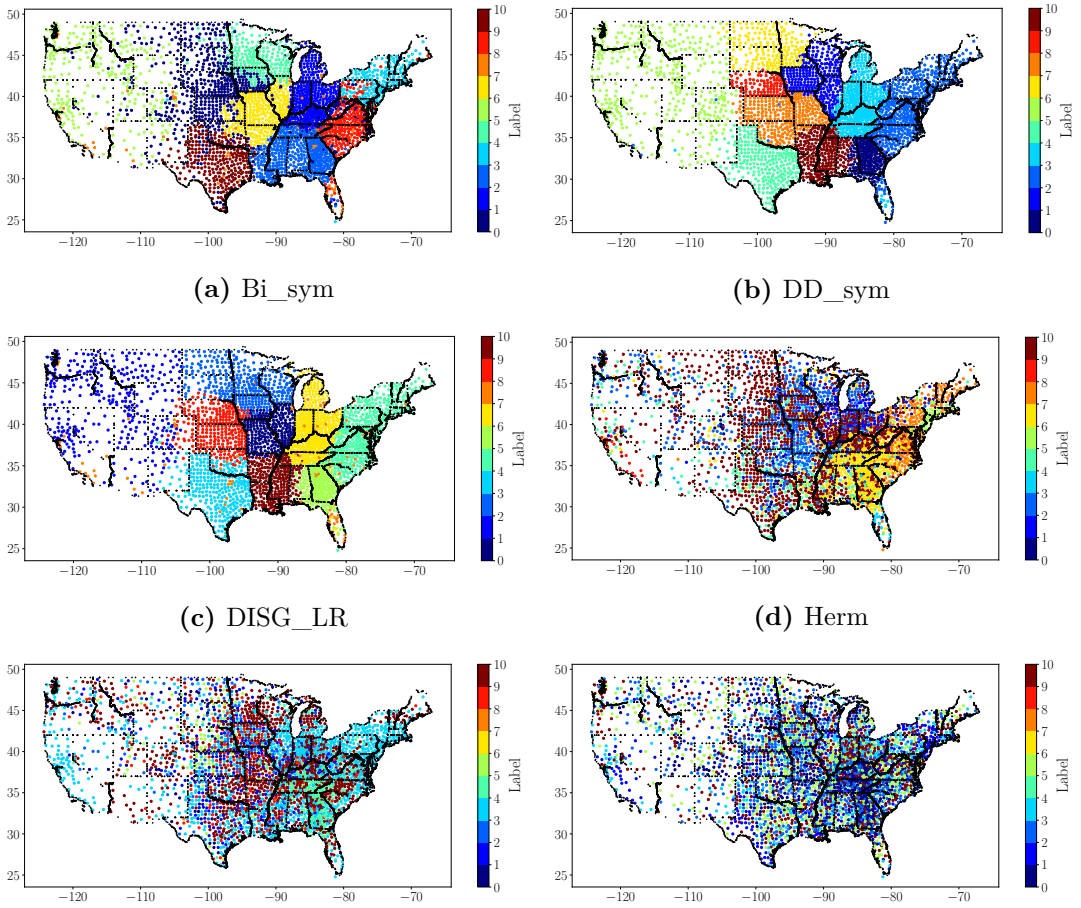
From Fig. B.15, we conclude that DIGRAC is able to recover a directed flow imbalance between clusters in all of the selected data sets. Fig. B.15a shows a clear cut imbalance between two clusters, possibly corresponding to the Republican and Democratic parties. Fig. B.15b plots imbalance flows in the real data set *Telegram*, where cluster 3 is a core-transient cluster, cluster 0 is a core-sink cluster, cluster 2 is a periphery-upstream cluster, while cluster 1 is a periphery-downstream cluster [4, 90]. For *WikiTalk*, illustrated in Fig. B.15d, the lower-triangular part entries are typically source nodes for edges, while the upper-triangular part are target nodes. For *Lead-Lag*, taking the year 2015 as an example, DIGRAC is also able to recover high imbalance in the data.

We also note that DIGRAC would not necessarily predict the same number of clusters as assumed, so that we do not need to specify the exact number of clusters before training DIGRAC; specifying the maximum number of possible clusters suffices.

B.4.4 Migration Plots

We compare DIGRAC to five spectral methods for recovering clusters for the US migration data set, and plot the recovered clusters on a map, in Fig. B.16.

The visualization in Fig. (a-c) shows that clusters align particularly well with the political and administrative boundaries of the US states, as previously observed in [228]. This outcome is not deemed too insightful, as it trivially reveals the fact there there is significant intra-state and inter-state migration, and does not uncover any of the information on latent migration patterns between far-away states, and more generally, between regions which are not necessarily geographically cohesive. DIGRAC outcomes, however, reveal nontrivial migration patterns, for example migration from New York to Florida, and from California to Arizona, which is consistent with the patterns discovered by [36]. Fig. B.17 details on the top pair migration patterns uncovered by DIGRAC.



(a) Bi_sym **(b)** DD_sym

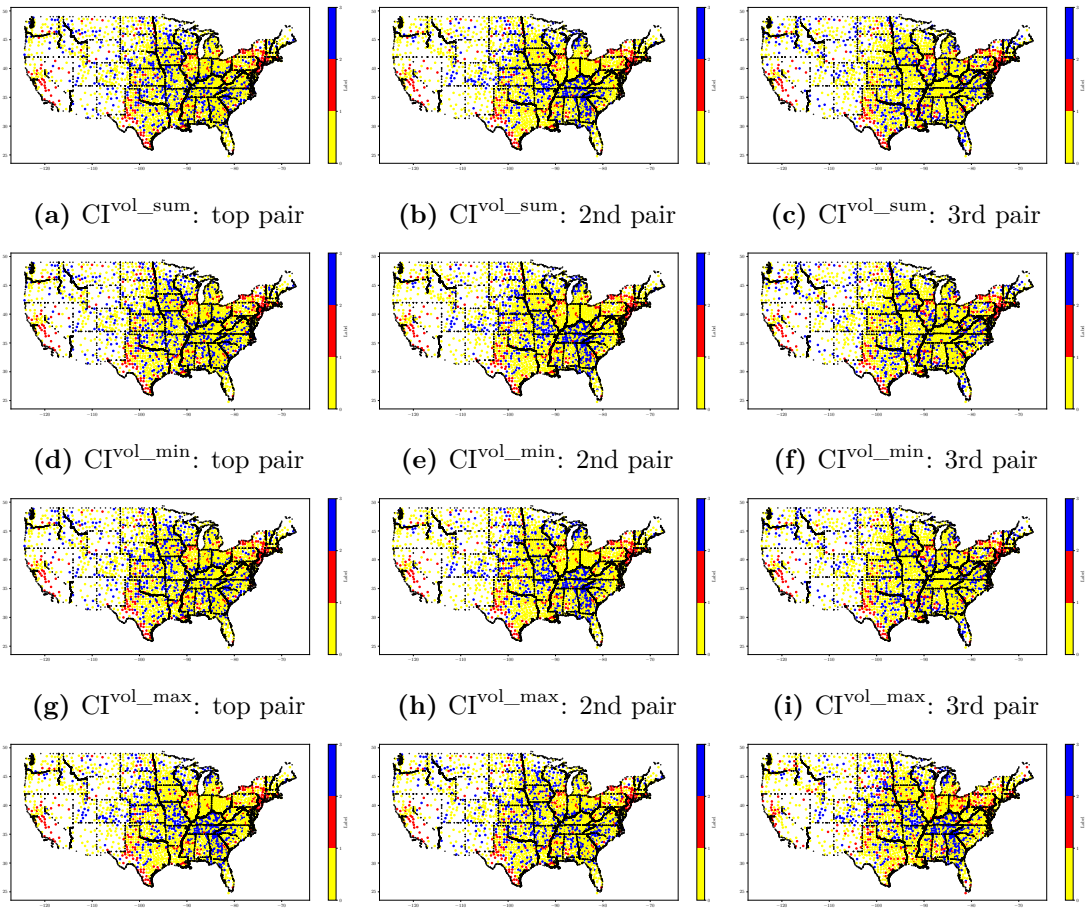
(c) DISG_LR **(d)** Herm

(e) Herm_rw **(f)** DIGRAC

Figure B.16: US migration predicted clusters, along with the geographic locations of the counties as well as state boundaries (in black). InfoMap results are omitted here because it only produces one huge cluster. The input data is normalized, following Eq. (B.10).

B.4.5 Coping with Outliers

As mentioned in Section B.3.3, the preprocessing step to use ratio of migration instead of absolute migration numbers is a way to cope with outliers (here, extremely large entries in the original digraph) in *Migration*. To validate the effectiveness of this approach to cope with outliers, Table B.11 provides imbalance results for



(a) $\mathrm{CI}^{\mathrm{vol\_sum}}$: top pair **(b)** $\mathrm{CI}^{\mathrm{vol\_sum}}$: 2nd pair **(c)** $\mathrm{CI}^{\mathrm{vol\_sum}}$: 3rd pair

(d) $\mathrm{CI}^{\mathrm{vol\_min}}$: top pair **(e)** $\mathrm{CI}^{\mathrm{vol\_min}}$: 2nd pair **(f)** $\mathrm{CI}^{\mathrm{vol\_min}}$: 3rd pair

(g) $\mathrm{CI}^{\mathrm{vol\_max}}$: top pair **(h)** $\mathrm{CI}^{\mathrm{vol\_max}}$: 2nd pair **(i)** $\mathrm{CI}^{\mathrm{vol\_max}}$: 3rd pair

(j) $\mathrm{CI}^{\mathrm{plain}}$: top pair **(k)** $\mathrm{CI}^{\mathrm{plain}}$: 2nd pair **(l)** $\mathrm{CI}^{\mathrm{plain}}$: 3rd pair

Figure B.17: US migration predicted cluster pairs with top imbalance, along with the geographic locations of the counties as well as state boundaries (in black). Red (label 1) is the sending cluster while blue (label 2) is the receiving cluster. Yellow (label 0) denotes all the other locations being considered. Subcaptions show the imbalance score and the rank based on that score.

Migration when we do not transform the nonzero entries into ratios. Comparing with Table B.6, we witness an overall decrease in the performance. In this case InfoMap no longer predicts a single huge cluster. However, its predicted number of clusters is about 44, which is too large. This also implies that InfoMap is very sensitive to the magnitude of digraph entries, while DIGRAC is not. Indeed, InfoMap gives 43 (too many) clusters for *Blog*, 19 (too many) for *Telegram*, 1 (too small) for *Migration*, and 17498 (far too many) for *WikiTalk*.

We compare DIGRAC to five spectral methods as well as InfoMap for recovering clusters for the US migration data set without the preprocessing step discussed earlier, and plot the recovered clusters on a map in Fig. B.18. Note that all methods, except DIGRAC, recover either clusters which are trivially small in size or contain one very large dominant cluster (as in (a), (b), (e) and to some extent, also (f)). The DISG_LR clustering and InfoMap clustering provide clear geographic boundaries, but were not able to recover the imbalance among clusters. Other spectral methods generally have a dominant cluster containing most of the nodes, whereas DIGRAC has more balanced cluster sizes.

Table B.11: Performance comparison on *Migration* (without preprocessing). The best is marked in **bold red** and the second best is marked in underline blue .

Metric/Method	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.02±0.00	0.03±0.00	0.01±0.00	0.01±0.00	**0.07±0.00**	0.01±0.00	0.04±0.00
$\mathcal{O}^{sort}_{vol\_min}$	**0.24±0.00**	0.20±0.01	0.12±0.02	0.14±0.00	0.21±0.01	0.05±0.02	0.18±0.02
$\mathcal{O}^{sort}_{vol\_max}$	0.02±0.00	0.03±0.00	0.01±0.00	0.01±0.00	**0.06±0.00**	0.00±0.00	0.04±0.00
$\mathcal{O}^{sort}_{plain}$	0.61±0.00	0.46±0.00	0.29±0.02	0.26±0.00	**0.62±0.02**	0.40±0.00	0.32±0.11
$\mathcal{O}^{std}_{vol\_sum}$	0.00±0.00	0.01±0.00	0.00±0.00	0.00±0.00	0.02±0.00	0.00±0.00	**0.03±0.01**
$\mathcal{O}^{std}_{vol\_min}$	0.03±0.00	0.09±0.00	0.04±0.01	0.05±0.00	0.08±0.01	0.02±0.01	**0.11±0.03**
$\mathcal{O}^{std}_{vol\_max}$	0.00±0.00	0.01±0.00	0.00±0.00	0.00±0.00	0.01±0.00	0.00±0.00	**0.02±0.01**
$\mathcal{O}^{std}_{plain}$	0.19±0.00	0.23±0.00	0.14±0.01	0.12±0.00	**0.32±0.01**	0.25±0.01	0.21±0.03
$\mathcal{O}^{naive}_{vol\_sum}$	0.00±0.00	0.01±0.00	0.00±0.00	0.00±0.00	0.02±0.00	0.00±0.00	**0.03±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.02±0.00	0.08±0.00	0.04±0.01	0.05±0.00	0.08±0.01	0.02±0.01	**0.11±0.04**
$\mathcal{O}^{naive}_{vol\_max}$	0.00±0.00	0.01±0.00	0.00±0.00	0.00±0.00	0.01±0.00	0.00±0.00	**0.02±0.01**
$\mathcal{O}^{naive}_{plain}$	0.16±0.00	0.22±0.00	0.13±0.01	0.11±0.00	**0.31±0.01**	0.22±0.00	0.21±0.03
size ratio	8.500	3043.80	722.620	25.780	**3059.20**	415.880	203.230
size std	**58.96**	912.100	861.280	409.900	917.230	844.750	342.38

When employing methods that symmetrize the adjacency matrix (as in (a) and (b)), the migration flows between counties in different states will be lost in the process. Furthermore, the visualization in Fig. (c) shows that clusters align particularly well with the political and administrative boundaries of the US states, as previously observed in [228]. The same is for Fig. (d). This outcome is not deemed very insightful, as it trivially reveals the fact that there is significant intra-state and inter-state migration, and does not uncover any of the information on latent migration patterns between far-away states, and more generally, between regions which are not necessarily geographically cohesive.

Fig. B.17 further plots the top three pairs of clusters based on four different imbalance scores given by DIGRAC. As shown in the figure, DIGRAC uncovers the migration trend from coastal to interior, across states. This trend of the directed flow agrees with that discussed in [36], with many people migrating from New York and California to the interior states.

B.5 Discussion of Related Methods that are not Compared against in the Main Text

To further emphasize the importance of directionality, our synthetic data sets have no difference in density between clusters; their sole signal is in the directionality of the edges. If all edge directions were to be removed, then no algorithm should be available to detect the clusters. To further support our claim why some methods mentioned in Section 3.2 in the main text are not appropriate for comparison, we have applied the default setting versions of the Louvain method [121], the Leiden algorithm [10] and OSLOM [120], to our synthetic data sets, and find that they do not detect the structure in the data, with ARI and NMI values very close to zero, and very low imbalance values. In particular, Louvain and Leiden tend to give a larger number of clusters than the ground truth which is designed to have small cluster sizes. OSLOM outputs clusters with extreme sizes, either a huge cluster containing (almost always) all the nodes, or every node forming a cluster by itself. To further demonstrate that comparing DIGRAC with density-based methods is unfair, We report the test ARI results for Infomap, Louvain and Leiden in Figure B.19.We

can see that Infomap, Louvain and Leiden normally produces nero-zero ARI values, which are much worse than the results from DIGRAC given in Figure 3.3.

On the real-world data sets, these methods often give numbers of clusters that do not match our expectations. (*Blog* has two underlying parties, *Telegram* has a four-cluster core-periphery structure). Louvain clusters nodes from *Blog* into 8-13 clusters (too many), *Telegram* into 4-5 clusters (acceptable), *Migration* into 5-7 clusters (acceptable), *WikiTalk* into 150-219 clusters (too many), and *Lead-Lag* into 10-55 clusters (acceptable or a bit too many). Leiden gives 12 (too many) clusters for *Blog*, 4-5 for *Telegram*, 5-6 for *Migration*, 170-248 (too many) for *WikiTalk*, and 10-55 clusters (acceptable or a bit too many) for *Lead-Lag*. OSLOM gives 6 clusters for *Blog* (too many), 16 for *Telegram* (too many), and 46 for *Migration* (too many). It could not generate results for *WikiTalk* after running for 12 hours, and hence we omit its discussion here. On *Lead-Lag*, OSLOM places every node in a single cluster for most of the years, and clusters the rest of the years into either a huge single cluster or two clusters.

None of the methods outperform DIGRAC on our chosen performance measures from Table 3.1 , except on the *Lead-Lag* data set (See Tables B.13, B.14, B.15 and B.16 for the other results). With regards to the 12 imbalance measures from Appendix Table B.5, leaving out OSLOM as before, Louvain and Leiden perform poorly on all of the real data sets, except on *Lead-Lag*. Indeed, for *Lead-Lag*, the number of clusters we use for DIGRAC is ten according to the GICS sector memberships. However, if we use the sector memberships as labels, the imbalance values are poor, which implies that ten may not be a desirable choice of the number of clusters. Further, DIGRAC usually clusters the nodes into smaller number of clusters, while Louvain and Leiden usually cluster the nodes into a larger number of clusters (usually around 30, and sometimes above 50 clusters).

Table B.12: Performance comparison on *Lead-Lag*, including Louvain and Leiden. Results in each year is averaged over ten runs. Mean and standard deviation (after ±) are calculated over the 19 years. The best is marked in **bold red** and the second best is marked in underline blue . InfoMap results are omitted here as it usually predicts a single huge cluster and could not generate imbalance results. Louvain and Leiden yield essentially identical results and often attain the highest objectives, while DIGRAC almost always places either first or second across all methods considered.

Metric/Method	Louvain/Leiden	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{\text{sort}}_{\text{vol\_sum}}$	0.08±0.02	0.07±0.01	0.07±0.01	0.07±0.01	0.07±0.02	0.07±0.02	**0.15±0.03**
$\mathcal{O}^{\text{sort}}_{\text{vol\_min}}$	0.15±0.04	**0.51±0.10**	0.48±0.09	0.47±0.10	**0.51±0.11**	0.50±0.10	0.47±0.09
$\mathcal{O}^{\text{sort}}_{\text{vol\_max}}$	0.08±0.02	0.07±0.01	0.06±0.01	0.06±0.01	0.07±0.01	0.07±0.01	**0.14±0.03**
$\mathcal{O}^{\text{sort}}_{\text{plain}}$	0.15±0.04	**0.66±0.09**	0.64±0.08	0.63±0.08	**0.66±0.09**	0.65±0.09	0.53±0.09
$\mathcal{O}^{\text{std}}_{\text{vol\_sum}}$	**0.23±0.06**	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	0.12±0.03
$\mathcal{O}^{\text{std}}_{\text{vol\_min}}$	**0.46±0.11**	0.27±0.04	0.27±0.04	0.25±0.04	0.27±0.03	0.27±0.03	0.38±0.07
$\mathcal{O}^{\text{std}}_{\text{vol\_max}}$	**0.23±0.05**	0.04±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.11±0.02
$\mathcal{O}^{\text{std}}_{\text{plain}}$	**0.46±0.11**	0.40±0.05	0.39±0.05	0.38±0.05	0.40±0.05	0.40±0.05	0.44±0.07
$\mathcal{O}^{\text{naive}}_{\text{vol\_sum}}$	**0.23±0.06**	0.03±0.01	0.03±0.01	0.03±0.01	0.03±0.01	0.03±0.01	0.08±0.04
$\mathcal{O}^{\text{naive}}_{\text{vol\_min}}$	**0.46±0.11**	0.20±0.05	0.19±0.05	0.18±0.05	0.19±0.04	0.19±0.04	0.26±0.10
$\mathcal{O}^{\text{naive}}_{\text{vol\_max}}$	**0.23±0.05**	0.03±0.01	0.02±0.01	0.02±0.01	0.02±0.00	0.02±0.00	0.08±0.03
$\mathcal{O}^{\text{naive}}_{\text{plain}}$	**0.46±0.11**	0.30±0.06	0.28±0.06	0.27±0.06	0.29±0.05	0.29±0.05	0.32±0.11
size ratio	124.530	3.67	**3.34**	3.900	4.110	3.880	8.070
size std	47.960	9.31	**9.14**	10.090	10.490	10.360	17.060

Finally, we provide more examples/explanations on why these density-based methods or even other methods that are based on random-walk should fail. We

Table B.13: Performance comparison on *Telegram*, including Louvain and Leiden. The best is marked in **bold red** and the second best is marked in underline blue .

Metric/Method	Louvain/Leiden	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.08±0.01	0.04±0.00	0.21±0.00	0.21±0.00	0.21±0.01	0.20±0.01	0.14±0.00	**0.32±0.01**
$\mathcal{O}^{sort}_{vol\_min}$	0.39±0.07	0.47±0.00	0.67±0.00	0.61±0.00	0.66±0.02	0.66±0.02	0.19±0.00	**0.79±0.06**
$\mathcal{O}^{sort}_{vol\_max}$	0.06±0.01	0.03±0.00	0.20±0.00	0.20±0.00	0.20±0.01	0.19±0.01	0.12±0.00	**0.29±0.01**
$\mathcal{O}^{sort}_{plain}$	0.71±0.05	**1.00±0.00**	0.80±0.00	0.75±0.00	0.78±0.03	0.76±0.04	0.59±0.00	0.96±0.01
$\mathcal{O}^{std}_{vol\_sum}$	0.07±0.01	0.01±0.00	0.26±0.00	0.26±0.00	0.26±0.01	0.25±0.02	**0.35±0.00**	0.28±0.01
$\mathcal{O}^{std}_{vol\_min}$	0.33±0.08	0.16±0.00	**0.84±0.00**	0.76±0.00	0.82±0.03	0.82±0.03	0.49±0.00	0.73±0.03
$\mathcal{O}^{std}_{vol\_max}$	0.05±0.01	0.01±0.00	0.25±0.00	0.25±0.00	0.25±0.01	0.24±0.02	**0.29±0.00**	0.25±0.01
$\mathcal{O}^{std}_{plain}$	0.59±0.05	0.68±0.00	**1.00±0.00**	0.94±0.00	0.98±0.04	0.95±0.04	0.99±0.00	0.90±0.05
$\mathcal{O}^{naive}_{vol\_sum}$	0.06±0.02	0.01±0.00	0.26±0.00	0.26±0.00	0.26±0.01	0.25±0.02	0.23±0.00	**0.27±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.28±0.11	0.11±0.00	**0.84±0.00**	0.76±0.00	0.82±0.03	0.82±0.03	0.32±0.00	0.72±0.04
$\mathcal{O}^{naive}_{vol\_max}$	0.04±0.01	0.00±0.00	**0.25±0.00**	**0.25±0.00**	**0.25±0.01**	0.24±0.02	0.20±0.00	0.24±0.01
$\mathcal{O}^{naive}_{plain}$	0.56±0.01	0.63±0.00	**1.00±0.00**	0.94±0.00	0.98±0.04	0.95±0.04	0.99±0.00	0.89±0.06

Table B.14: Performance comparison on *Blog*, including Louvain and Leiden. The best is marked in **bold red** and the second best is marked in underline blue .

Metric/Method	Louvain/Leiden	InfoMap	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.00±0.00	0.07±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
$\mathcal{O}^{sort}_{vol\_min}$	0.01±0.01	0.02±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{sort}_{vol\_max}$	0.01±0.01	0.05±0.00	0.05±0.00	0.00±0.00	0.04±0.00	0.26±0.00	0.00±0.00	**0.40±0.00**
$\mathcal{O}^{sort}_{plain}$	**1.00±0.00**	**1.00±0.00**	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	0.89±0.00	0.76±0.00
$\mathcal{O}^{std}_{vol\_sum}$	0.00±0.00	0.00±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
$\mathcal{O}^{std}_{vol\_min}$	0.00±0.00	0.00±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{std}_{vol\_max}$	0.00±0.00	0.00±0.00	0.05±0.00	0.00±0.00	0.04±0.00	0.26±0.00	0.00±0.00	**0.40±0.00**
$\mathcal{O}^{std}_{plain}$	0.56±0.13	0.73±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{naive}_{vol\_sum}$	0.00±0.00	0.00±0.00	0.07±0.00	0.00±0.00	0.05±0.00	0.37±0.00	0.00±0.00	**0.44±0.00**
$\mathcal{O}^{naive}_{vol\_min}$	0.00±0.00	0.00±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00
$\mathcal{O}^{naive}_{vol\_max}$	0.00±0.00	0.00±0.00	0.05±0.00	0.00±0.00	0.04±0.00	0.26±0.00	0.00±0.00	**0.40±0.00**
$\mathcal{O}^{naive}_{plain}$	0.76±0.00	0.76±0.00	0.33±0.00	0.05±0.00	0.31±0.00	0.78±0.01	**0.89±0.00**	0.76±0.00

would mainly like to point out a family of illustrative examples demonstrating the subtle nuance concerning edge density.

Consider a meta graph with $K = 3$ nodes (clusters) A,B,C with directed edges AB, BC, CA, hence a directed cycle (our "cycle" DSBM models). Each pair of nodes (v_i, v_j) in the graph of size n is connected by an edge independently with probability p (which can even be equal to 1, in the case of a complete graph), hence the graph has the same density throughout. Now suppose we consider a pair of nodes (v_i, v_j) such that v_i belongs to cluster A, and v_j to cluster B. Since this edge is part of the metagraph, with probability 1-eta, it is directed from v_i to v_j, and with probability eta, v_j sends an edge to v_i (here, eta is the noise level parameter). Similar arguments can be made when v_i (resp v_j) belongs to cluster B (resp C); and when v_i (resp v_j) belongs to cluster C (resp A). See Figure B.20 for an illustration. We also see that when the network is complete (see Figure B.19 (g) and Table B.8), InfoMap [99] fails empirically as it produces a single huge cluster. As a method based on random walks, this failure might occur as the chain could hardly be trapped inside a cluster as in the usual setting.

In such synthetic DSBM models with a "cycle" meta-graph structure, it can be shown that all nodes have the same in-degree and out-degree in expectation. Therefore, any density-based methods or modularity-based methods should fail. As the simplest possible example, one could just consider $K = 3$ clusters as above, without any noise (thus $\eta = 0$). InfoMap [99] tries to minimize the description

Table B.15: Performance comparison on *Migration*, including Louvain and Leiden. The best is marked in **bold red** and the second best is marked in underline blue. InfoMap results are omitted here as it predicts a single huge cluster and could not generate imbalance results.

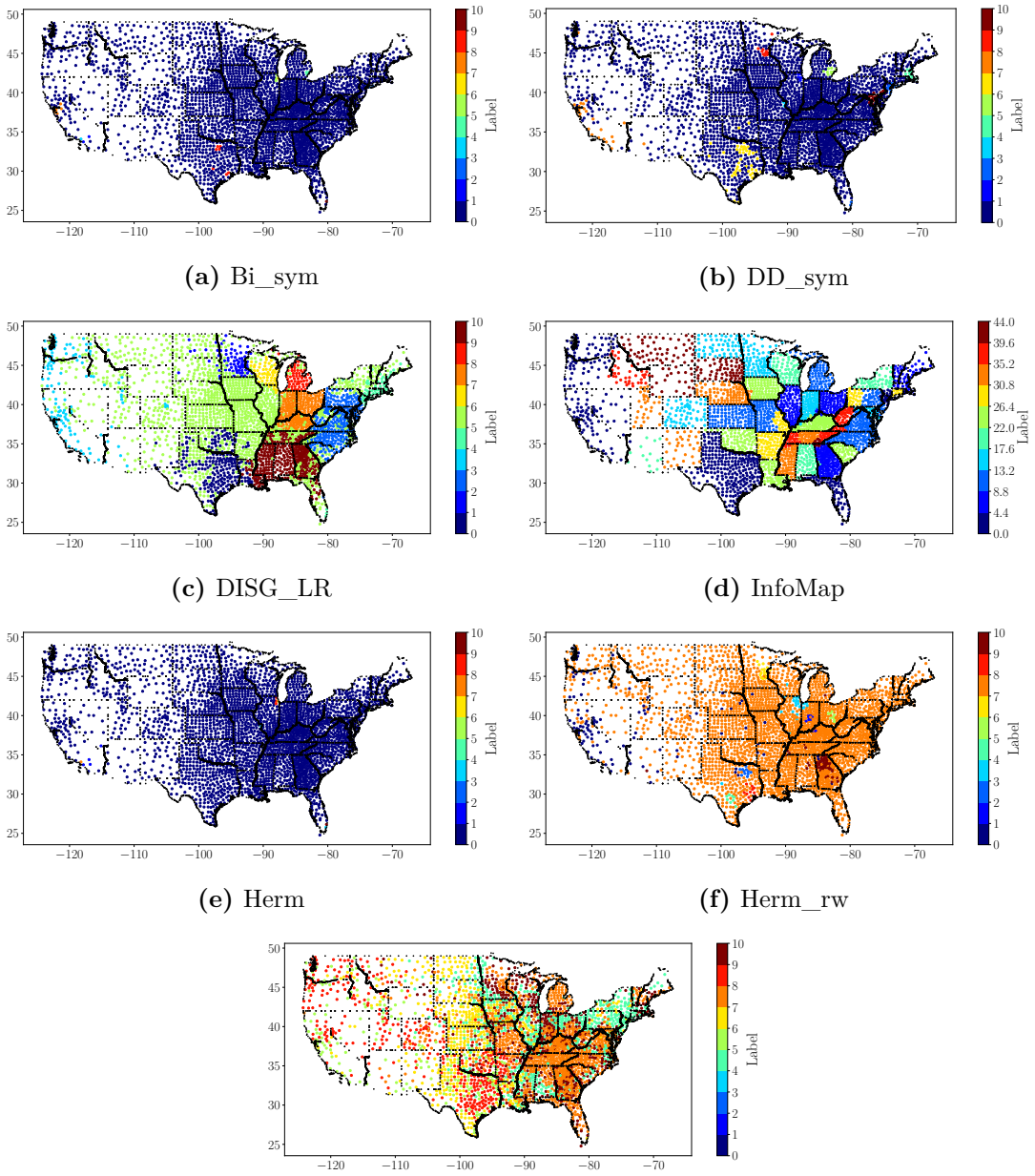
Metric/Method	Louvain/Leiden	Bi_sym	DD_sym	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.01±0.00	0.03±0.00	0.01±0.00	0.02±0.00	0.04±0.00	0.02±0.00	**0.05±0.00**
$\mathcal{O}^{sort}_{vol\_min}$	0.05±0.01	**0.19±0.00**	0.08±0.00	0.08±0.00	0.15±0.02	0.05±0.00	0.18±0.03
$\mathcal{O}^{sort}_{vol\_max}$	0.01±0.00	0.03±0.00	0.01±0.00	0.01±0.00	0.03±0.00	0.02±0.00	**0.04±0.00**
$\mathcal{O}^{sort}_{plain}$	0.09±0.02	0.24±0.00	0.20±0.00	0.17±0.00	0.40±0.01	**0.49±0.06**	0.29±0.04
$\mathcal{O}^{std}_{vol\_sum}$	0.00±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.02±0.00	**0.04±0.01**
$\mathcal{O}^{std}_{vol\_min}$	0.04±0.01	0.10±0.00	0.05±0.00	0.05±0.00	0.08±0.01	0.04±0.00	**0.16±0.03**
$\mathcal{O}^{std}_{vol\_max}$	0.00±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	**0.03±0.01**
$\mathcal{O}^{std}_{plain}$	0.07±0.01	0.13±0.00	0.12±0.00	0.11±0.00	0.20±0.01	0.20±0.01	**0.26±0.01**
$\mathcal{O}^{naive}_{vol\_sum}$	0.00±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00	0.01±0.00	**0.04±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.04±0.01	0.09±0.00	0.04±0.00	0.04±0.00	0.08±0.01	0.01±0.00	**0.16±0.03**
$\mathcal{O}^{naive}_{vol\_max}$	0.00±0.00	0.01±0.00	0.00±0.00	0.01±0.00	0.01±0.00	0.00±0.00	**0.03±0.01**
$\mathcal{O}^{naive}_{plain}$	0.07±0.00	0.12±0.00	0.10±0.00	0.08±0.00	0.19±0.00	0.19±0.03	**0.26±0.01**

Table B.16: Performance comparison on *WikiTalk*, including Louvain and Leiden. The best is marked in **bold red** and the second best is marked in underline blue. InfoMap results are omitted here as its large number of predicted clusters leads to memory error in imbalance calculation.

Metric/Method	Louvain/Leiden	DISG_LR	Herm	Herm_rw	DIGRAC
$\mathcal{O}^{sort}_{vol\_sum}$	0.01±0.00	0.18±0.03	0.15±0.02	0.00±0.00	**0.24±0.05**
$\mathcal{O}^{sort}_{vol\_min}$	0.15±0.00	0.10±0.03	0.22±0.05	0.26±0.00	**0.28±0.13**
$\mathcal{O}^{sort}_{vol\_max}$	0.01±0.00	0.16±0.03	0.09±0.01	0.00±0.00	**0.19±0.04**
$\mathcal{O}^{sort}_{plain}$	**1.00±0.00**	0.87±0.08	0.99±0.01	0.98±0.00	**1.00±0.00**
$\mathcal{O}^{std}_{vol\_sum}$	0.00±0.00	**0.17±0.04**	0.06±0.01	0.01±0.00	0.14±0.02
$\mathcal{O}^{std}_{vol\_min}$	0.01±0.00	0.09±0.02	0.09±0.02	**0.27±0.00**	0.18±0.08
$\mathcal{O}^{std}_{vol\_max}$	0.00±0.00	**0.15±0.04**	0.04±0.00	0.00±0.00	0.11±0.02
$\mathcal{O}^{std}_{plain}$	0.42±0.00	0.72±0.03	0.70±0.05	**0.98±0.00**	0.84±0.06
$\mathcal{O}^{naive}_{vol\_sum}$	0.00±0.00	0.10±0.02	0.04±0.00	0.00±0.00	**0.12±0.01**
$\mathcal{O}^{naive}_{vol\_min}$	0.01±0.00	0.06±0.03	0.07±0.02	**0.26±0.00**	0.15±0.07
$\mathcal{O}^{naive}_{vol\_max}$	0.00±0.00	**0.09±0.02**	0.03±0.00	0.00±0.00	**0.09±0.01**
$\mathcal{O}^{naive}_{plain}$	0.43±0.00	0.64±0.04	0.61±0.04	**0.98±0.00**	0.76±0.06

length, but as no description length difference occurs in the ground-truth clustering structure for such "cycle" DSBMs, if we consider a brute-force optimization of the map equation. Indeed, for any method that is based on a random walk, the probability of the random walker going from one cluster to another is the same as staying within the cluster. Therefore, we could hardly optimize anything if we base our clustering structure on a random walker's visit frequencies/path lengths. Similarly, the Markov clustering algorithm [229] is based on the intuition that higher-length paths would be relatively more likely to stay within clusters – an assumption that is not warranted when there is no density difference. [100] and [101] are two interesting Markov aggregation algorithms based on information theory and automatic control ideas that might be able to cover the above example and

may inspire some further comparison, but we omit comparison to them for now as we already have more than ten comparison methods and that InfoMap shares similar ideas to these two papers. As another example, as shown in [104], using belief propagation, in our model community structure should not be detectible (the right-hand side of (20) in [104] is zero for our "cycle" DSBMs). Therefore, at least methods that rely on belief propagation will fail on our benchmark models.



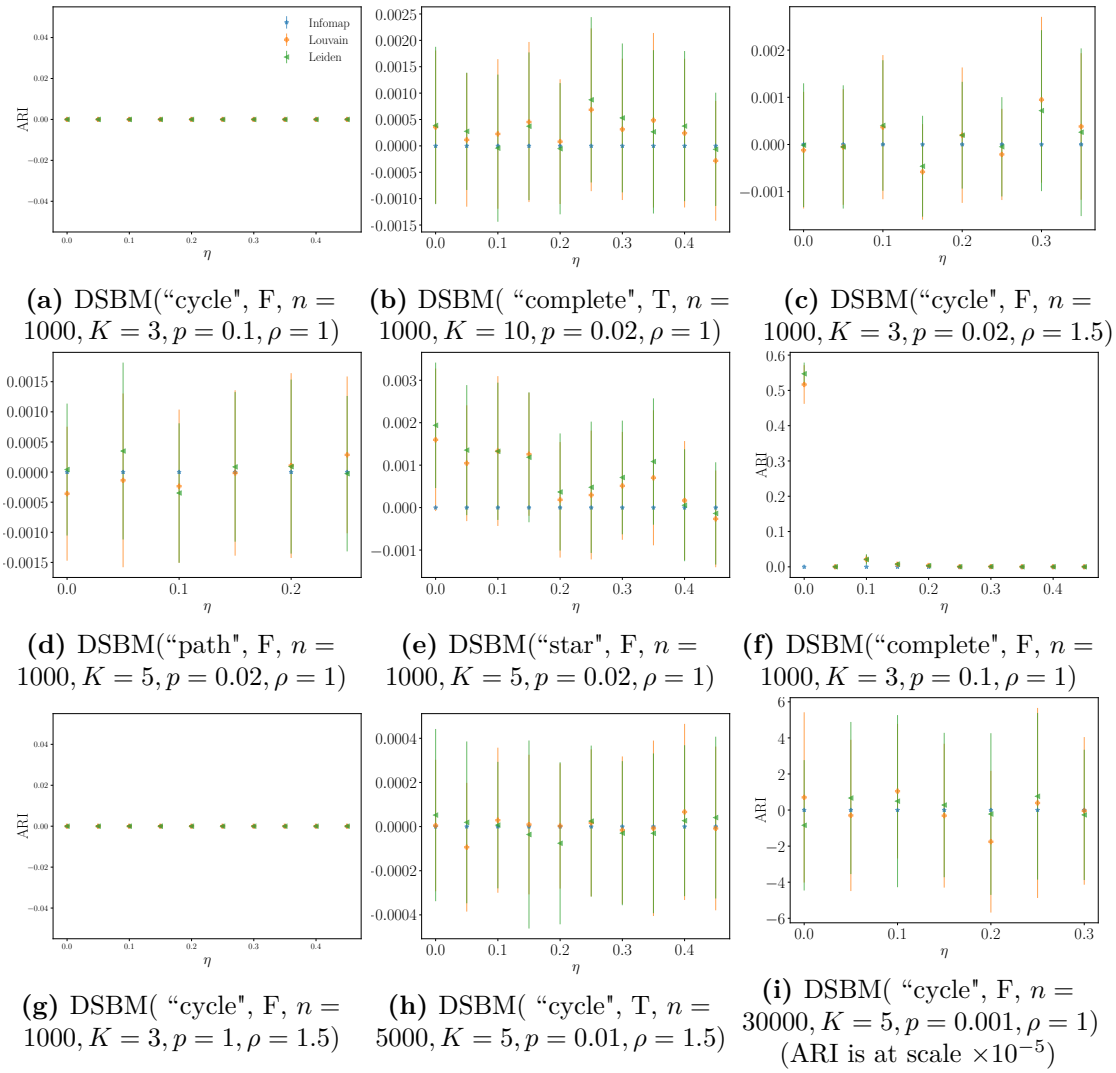
(a) Bi_sym **(b)** DD_sym

(c) DISG_LR **(d)** InfoMap

(e) Herm **(f)** Herm_rw

(g) DIGRAC

Figure B.18: US migration predicted clusters, along with the geographic locations of the counties as well as state boundaries (in black). The input digraph has extremely large entries; unlike in Fig. B.16, we do not employ here the normalization given by Eq. (B.10). Altogether, this demonstrates the robustness of DIGRAC to outliers in the data, which is not a characteristic of other state-of-the-art methods such as Herm and Herm_rw.



(a) DSBM("cycle", F, $n = 1000, K = 3, p = 0.1, \rho = 1$)

(b) DSBM("complete", T, $n = 1000, K = 10, p = 0.02, \rho = 1$)

(c) DSBM("cycle", F, $n = 1000, K = 3, p = 0.02, \rho = 1.5$)

(d) DSBM("path", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)

(e) DSBM("star", F, $n = 1000, K = 5, p = 0.02, \rho = 1$)

(f) DSBM("complete", F, $n = 1000, K = 3, p = 0.1, \rho = 1$)

(g) DSBM("cycle", F, $n = 1000, K = 3, p = 1, \rho = 1.5$)

(h) DSBM("cycle", T, $n = 5000, K = 5, p = 0.01, \rho = 1.5$)

(i) DSBM("cycle", F, $n = 30000, K = 5, p = 0.001, \rho = 1$) (ARI is at scale $\times 10^{-5}$)

Figure B.19: Test ARI comparison on synthetic data for Infomap, Louvain and Leiden. Error bars are given by one standard error.

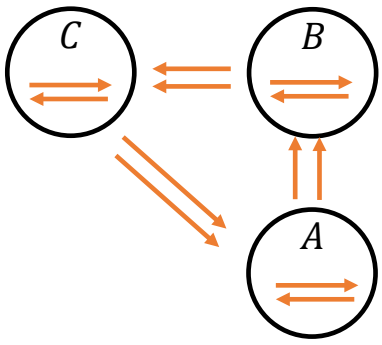


Figure B.20: An example of a "cycle" meta-graph.

C

MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian Supplementary Information

C.1 Proof of Theorems

C.1.1 Proof of Theorem 1

Proof. Let $\mathbf{x} \in \mathbb{C}^n$. Since $\mathbf{L}_U^{(q)}$ is Hermitian, we have $\text{Imag}(\mathbf{x}^\dagger \mathbf{L}_U^{(q)} \mathbf{x}) = 0$. Next, we note by the triangle inequality that $\tilde{\mathbf{D}}_{i,i} = \frac{1}{2}\sum_{j=1}^n (|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|) \geq \sum_{j=1}^n |\tilde{\mathbf{A}}_{i,j}|$. Therefore, we may use the fact that $\tilde{\mathbf{A}}$ is symmetric to obtain

$$
\begin{aligned}
&2\text{Real}\left(\mathbf{x}^\dagger \mathbf{L}_U^{(q)} \mathbf{x}\right) \\
=&2\sum_{i=1}^n \tilde{\mathbf{D}}_{i,i}|\mathbf{x}(i)|^2 - 2\sum_{i,j=1}^n \tilde{\mathbf{A}}_{i,j}\mathbf{x}(i)\overline{\mathbf{x}(j)}\cos(\mathbf{\Theta}_{i,j}^{(q)}) \\
\geq&2\sum_{i,j=1}^n |\tilde{\mathbf{A}}_{i,j}||\mathbf{x}(i)|^2 - 2\sum_{i,j=1}^n |\tilde{\mathbf{A}}_{i,j}||\mathbf{x}(i)||\mathbf{x}(j)| \\
=&\sum_{i,j=1}^n |\tilde{\mathbf{A}}_{i,j}||\mathbf{x}_i|^2 + \sum_{i,j=1}^n |\tilde{\mathbf{A}}_{i,j}||\mathbf{x}_j|^2 - 2\sum_{i,j=1}^n |\tilde{\mathbf{A}}_{i,j}||\mathbf{x}_i||\mathbf{x}_j| \\
=&\sum_{i,j=1}^n |\tilde{\mathbf{A}}_{i,j}|\left(|\mathbf{x}(i)| - |\mathbf{x}(j)|\right)^2 \geq 0.
\end{aligned}
$$

Thus, $\mathbf{L}_U^{(q)}$ is positive semidefinite. For the normalized magnetic Laplacian, one may verify $\left(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\right) \odot \exp(\mathbb{i}\boldsymbol{\Theta}^{(q)}) = \tilde{\mathbf{D}}^{-1/2}\left(\tilde{\mathbf{A}} \odot \exp(\mathbb{i}\boldsymbol{\Theta}^{(q)})\right)\tilde{\mathbf{D}}^{-1/2}$, and hence

$$\mathbf{L}_N^{(q)} = \tilde{\mathbf{D}}^{-1/2}\mathbf{L}_U^{(q)}\tilde{\mathbf{D}}^{-1/2}. \tag{C.1}$$

Thus, letting $\mathbf{y} = \tilde{\mathbf{D}}^{-1/2}\mathbf{x}$, the fact that $\tilde{\mathbf{D}}$ is diagonal implies

$$\mathbf{x}^\dagger \mathbf{L}_N^{(q)}\mathbf{x} = \mathbf{x}^\dagger \tilde{\mathbf{D}}^{-1/2}\mathbf{L}_U^{(q)}\tilde{\mathbf{D}}^{-1/2}\mathbf{x} = \mathbf{y}^\dagger \mathbf{L}_U^{(q)}\mathbf{y} \geq 0. \qquad \square$$

C.1.2 Proof of Theorem 2

Proof. By Theorem 1, it suffices to show that the lead eigenvalue, λ_n, is less than or equal to 2. The Courant-Fischer theorem shows that

$$\lambda_n = \max_{\mathbf{x}\neq 0} \frac{\mathbf{x}^\dagger \mathbf{L}_N^{(q)}\mathbf{x}}{\mathbf{x}^\dagger\mathbf{x}}.$$

Therefore, using equation C.1 and setting $\mathbf{y} = \tilde{\mathbf{D}}^{-1/2}\mathbf{x}$, we have

$$\lambda_n = \max_{\mathbf{x}\neq 0} \frac{\mathbf{x}^\dagger \tilde{\mathbf{D}}^{-1/2}\mathbf{L}_U^{(q)}\tilde{\mathbf{D}}^{-1/2}\mathbf{x}}{\mathbf{x}^\dagger\mathbf{x}} = \max_{\mathbf{y}\neq 0} \frac{\mathbf{y}^\dagger \mathbf{L}_U^{(q)}\mathbf{y}}{\mathbf{y}^\dagger\tilde{\mathbf{D}}\mathbf{y}}.$$

First, we observe that since $\tilde{\mathbf{D}}$ is diagonal, we have

$$\mathbf{y}^\dagger\tilde{\mathbf{D}}\mathbf{y} = \sum_{i,j=1}^n \tilde{\mathbf{D}}_{i,j}\mathbf{y}_i\overline{\mathbf{y}_j} = \sum_{i=1}^n \tilde{\mathbf{D}}_{i,i}|\mathbf{y}(i)|^2 = \frac{1}{2}\sum_{i,j=1}^n (|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|)|\mathbf{y}(i)|^2.$$

The triangle inequality implies that $|\tilde{\mathbf{A}}_{i,j}| \leq \frac{1}{2}(|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|)$. Therefore, we may repeatedly expand the sums and interchange the roles of i and j to obtain

$$\begin{aligned}
&\mathbf{y}^\dagger \mathbf{L}_U^{(q)}\mathbf{y} \\
\leq &\frac{1}{2}\sum_{i,j=1}^n (|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|)|\mathbf{y}(i)|^2 + \frac{1}{2}\sum_{i,j=1}^n (|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|)|\mathbf{y}(i)||\mathbf{y}(j)| \\
= &\frac{1}{2}\sum_{i,j=1}^n |\mathbf{A}_{i,j}|(|\mathbf{y}_i|^2 + |\mathbf{y}_j|^2 + 2|\mathbf{y}_i||\mathbf{y}_j|) \\
= &\frac{1}{2}\sum_{i,j=1}^n |\mathbf{A}_{i,j}|(|\mathbf{y}(i)| + |\mathbf{y}(j)|)^2 \leq \sum_{i,j=1}^n |\mathbf{A}_{i,j}|(|\mathbf{y}(i)|^2 + |\mathbf{y}(j)|^2) \\
= &\sum_{i,j=1}^n (|\mathbf{A}_{i,j}| + |\mathbf{A}_{j,i}|)|\mathbf{y}(i)|^2 = 2\mathbf{y}^\dagger\tilde{\mathbf{D}}\mathbf{y}.
\end{aligned}$$

$\square$

C.2 Implementation Details

Experiments were conducted on two compute nodes, each with 8 Nvidia Tesla T4, 96 Intel Xeon Platinum 8259CL CPUs @ 2.50GHz and 378GB RAM. Table C.1 reports total runtime after data preprocessing (in seconds) on link tasks for competing

Table C.1: Runtime (seconds) comparison on link tasks. The fastest is marked in **bold red** and the second fastest is marked in underline blue .

Data Set	Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
BitCoin-Alpha	SP	352	124	277	438	59	151	**29**
	DP	328	196	432	498	78	152	**37**
	3C	403	150	288	446	77	245	**37**
	4C	385	133	293	471	57	143	**36**
	5C	350	373	468	570	82	182	**37**
BitCoin-OTC	SP	340	140	397	584	68	222	**30**
	DP	471	243	426	941	80	155	**38**
	3C	292	252	502	551	92	230	**37**
	4C	347	143	487	607	68	209	**37**
	5C	460	507	500	959	86	326	**38**
Slashdot	SP	4218	3282	1159	5792	342	779	**227**
	DP	4231	3129	1200	5773	311	817	**222**
	3C	3686	6517	1117	6628	**263**	642	322
	4C	4038	5296	948	7349	**202**	535	232
	5C	4269	7394	904	8246	424	390	**327**
Epinions	SP	6436	4725	2527	8734	**300**	1323	370
	DP	6437	4605	2381	8662	404	1319	**369**
	3C	6555	8746	2779	10536	**471**	885	510
	4C	6466	6923	2483	10380	**272**	727	384
	5C	7974	9310	2719	11780	**460**	551	517
FiLL (avg.)	SP	591	320	367	617	61	63	**32**
	DP	387	316	363	386	53	38	**36**
	3C	542	471	298	657	79	114	**43**
	4C	608	384	343	642	56	78	**35**
	5C	318	534	266	521	63	66	**44**

methods. We conclude that for large graphs SNEA is the least efficient method in terms of speed due to the attention mechanism employed, followed by SDGNN, which needs to count motifs. MSGNN is generally the fastest. Averaged results are reported with error bars representing one standard deviation in the figures, and plus/minus one standard deviation in the tables. For all the experiments, we use Adam [226] as our optimizer with a learning rate of 0.01 and employ ℓ_2 regularization with weight decay parameter $5{\cdot}10^{-4}$ to avoid overfitting. We use the open-source library PyTorch Geometric Signed Directed [28] for data loading, node and edge splitting, node feature preparation, and implementation of some baselines. For SSSNET [3], we use hidden size 16, 2 hops, and $\tau = 0.5$, and we adapt the architecture so that SSSNET is suitable for link prediction tasks. For SigMaNet [138], we use the code and parameter settings from `https://anonymous.4open.science/r/SigMaNet`. We set the number of layers to two for all methods.

C.2.1 Node Clustering

We conduct semi-supervised node clustering, with 10% of all nodes from each cluster as test nodes, 10% as validation nodes to select the model, and the remaining 80% as training nodes (10% of which are seed nodes). For each of the synthetic models,

we first generate five different networks, each with two different data splits, then conduct experiments on them and report average performance over these 10 runs. To train the GNNs on the signed undirected data sets (SSBMs and POL-SSBMs), we use the semi-supervised loss function $\mathcal{L}_1 = \mathcal{L}_{\text{PBNC}} + \gamma_s(\mathcal{L}_{\text{CE}} + \gamma_t \mathcal{L}_{\text{triplet}})$ as in [3], with the same hyperparameter setting $\gamma_s = 50, \gamma_t = 0.1$, where $\mathcal{L}_{\text{PBNC}}$ is the self-supervised probablistic balanced normalized cut loss function penalizing unexpected signs. For these signed undirected graphs, we use validation ARI for early stopping. For the SDSBMs, our loss function is the sum of $\mathcal{L}_1$ and the imbalance loss function $\mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$ from [5] (absolute edge weights as input), i.e., $\mathcal{L}_2 = \mathcal{L}^{\text{sort}}_{\text{vol\_sum}} + \mathcal{L}_1$, and we use the self-supervised part of the validation loss ($\mathcal{L}_{\text{PBNC}} + \mathcal{L}^{\text{sort}}_{\text{vol\_sum}}$) for early stopping. We further restrict the GNNs to be trained on the subgraph induced by only the training nodes while applying the training loss function. For MSGNN on SDSBMs, we set $q = 0.25$ to emphasize directionality. The input node feature matrix $\mathbf{X}_{\mathcal{V}}$ for undirected signed networks in our experiments is generated by stacking the eigenvectors corresponding to the largest C eigenvalues of a regularized version of the symmetrized adjacency matrix $\tilde{\mathbf{A}}$. For signed, directed networks, we calculate the in- and out-degrees based on both signs to obtain a four-dimensional feature vector for each node. We train all GNNs for the node clustering task for at most 1000 epochs with a 200-epoch early-stopping scheme.

C.2.2 Link Prediction

We train all GNNs for each link prediction task for 300 epochs. We use the proposed loss functions from their original papers for SGCN [48], SNEA [54], SiGAT [137], and SDGNN [133], and we use cross-entropy loss $\mathcal{L}_{\text{CE}}$ for SigMaNet [138], SSSNET [3] and MSGNN. For all link prediction experiments, we sample 20% edges as test edges, and use the rest of the edges for training. Five splits were generated randomly for each input graph. We calculate the in- and out-degrees based on both signs from the observed input graph (removing test edges) to obtain a four-dimensional feature vector for each node for training SigMaNet [138], SSSNET [3], and MSGNN, and we use the default settings from [28] for SGCN [48], SNEA [54], SiGAT [137], and SDGNN [133].

C.3 Ablation Study and Discussion

Table C.2 compares different variants of MSGNN on the link prediction tasks (Table C.9 reports runtime), with respect to whether we use a traditional signed Laplacian that is initially designed for undirected networks (in which case we have $q = 0$) and a magnetic signed Laplacian, with $q = q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$, which strongly emphasizes directionality. We also assess whether to include sign in input node features, and whether we take edge weights into account. Note that, by default, the degree calculated given signed edge weights are net degrees, meaning that we sum the edge weights up without taking absolute values, which means that -1 and 1 would cancel out during calculation. The features that sum up absolute values of edge weights are denoted with T' in the table. Taking the standard error into account, we find no significant difference between the two options T and T' as weight features when signed features are not considered. We provide a toy example here to help better understand what the tuples mean. Consider

Table C.2: Link prediction test performance (accuracy in percentage) comparison for variants of MSGNN. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The best is marked in **bold red** and the second best is marked in underline blue .

q value				0			$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Data Set	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
BitCoin-Alpha	SP	71.3±1.2	70.5±1.4	70.1±1.3	**71.9±0.8**	71.3±1.2	71.6±0.9	71.6±1.6	70.2±1.1	71.8±1.1	71.3±1.0
	DP	73.8±1.5	72.5±0.8	73.5±0.7	73.2±1.4	69.9±1.6	74.8±1.0	71.8±1.6	71.6±1.8	**75.3±1.3**	72.5±1.5
	3C	85.6±0.3	84.4±0.5	84.3±0.6	85.7±0.5	84.3±0.5	**85.8±0.9**	84.2±0.9	84.4±0.6	85.6±0.6	84.4±0.6
	4C	59.3±2.4	56.4±2.2	56.5±1.9	58.6±1.0	58.9±0.6	58.8±1.1	55.2±2.3	56.6±1.6	**59.4±1.4**	58.5±0.7
	5C	83.8±0.4	82.3±0.6	81.3±1.7	**83.9±0.6**	82.1±0.5	83.3±0.6	82.0±0.5	81.9±0.5	83.2±0.3	81.9±0.9
BitCoin-OTC	SP	74.0±0.6	72.9±0.9	71.8±1.9	73.7±1.2	73.0±1.4	73.7±1.5	73.0±0.6	73.3±0.8	**74.1±1.1**	72.1±2.5
	DP	73.4±2.2	72.3±1.4	72.3±0.6	73.8±0.9	73.6±0.8	74.8±0.7	73.6±1.1	72.6±1.4	**75.2±1.4**	71.8±1.1
	3C	83.7±0.8	82.9±0.7	82.4±1.0	84.2±0.4	83.0±0.6	**85.0±0.5**	83.9±0.4	83.3±1.0	84.8±0.9	83.3±0.7
	4C	60.5±1.2	59.6±0.9	59.4±2.6	63.0±1.4	61.7±0.7	61.4±0.4	58.4±0.9	55.9±2.1	**63.3±1.4**	59.8±0.7
	5C	81.1±0.6	79.8±0.6	79.0±0.9	82.4±0.3	80.0±1.3	82.4±0.8	78.0±2.5	80.0±0.7	**82.6±0.7**	80.9±0.9
Slashdot	SP	92.3±0.2	92.4±0.2	92.3±0.2	92.4±0.2	92.4±0.2	93.0±0.0	93.0±0.1	93.0±0.1	**93.1±0.1**	**93.1±0.1**
	DP	92.2±0.3	92.3±0.2	92.4±0.2	92.4±0.2	92.4±0.2	**93.1±0.1**	**93.1±0.1**	**93.1±0.1**	93.0±0.1	**93.1±0.1**
	3C	86.0±0.1	85.8±0.2	86.0±0.1	85.9±0.3	85.7±0.4	86.2±0.2	**86.3±0.4**	86.2±0.3	**86.3±0.2**	86.1±0.3
	4C	77.1±0.6	76.9±0.7	76.3±0.8	78.1±0.4	77.7±0.5	70.3±1.1	71.5±1.1	70.7±1.2	77.9±0.6	**78.2±0.3**
	5C	77.1±0.7	77.4±0.4	77.7±0.3	**78.1±0.4**	77.8±0.3	72.8±0.3	73.1±0.3	72.8±0.3	77.5±0.6	76.8±0.6
Epinions	SP	85.2±0.7	85.5±0.4	85.2±0.7	85.9±0.3	85.4±0.5	86.4±0.1	**86.6±0.1**	86.4±0.1	**86.6±0.2**	86.3±0.1
	DP	85.1±0.8	85.3±0.7	85.4±0.4	85.0±0.5	85.3±0.7	86.2±0.2	86.1±0.7	86.1±0.4	**86.3±0.1**	**86.3±0.3**
	3C	83.0±0.6	83.0±0.6	82.7±0.6	83.2±0.3	82.9±0.4	83.5±0.2	83.3±0.3	**83.6±0.4**	**83.6±0.3**	83.1±0.5
	4C	78.3±0.6	78.2±1.6	79.1±1.2	79.7±1.2	**80.0±1.0**	76.6±1.3	76.7±1.5	76.5±1.5	79.3±0.5	78.7±0.9
	5C	79.7±1.4	77.6±3.9	80.4±0.4	80.5±0.2	**80.9±0.5**	78.3±0.9	78.6±1.4	78.5±0.7	80.1±0.8	80.5±0.5
FiLL (avg.)	SP	74.0±0.1	75.5±0.1	75.5±0.1	75.1±0.1	**76.2±0.1**	73.8±0.1	75.5±0.0	75.5±0.1	75.1±0.1	76.1±0.1
	DP	74.0±0.1	75.3±0.1	75.3±0.1	75.0±0.1	**75.9±0.1**	73.5±0.3	75.2±0.0	75.2±0.1	75.0±0.0	**75.9±0.1**
	3C	74.1±0.1	75.5±0.0	75.6±0.0	75.2±0.1	**76.2±0.1**	73.8±0.1	75.4±0.1	75.5±0.1	75.1±0.0	76.1±0.0
	4C	74.0±0.2	75.6±0.1	75.6±0.0	75.2±0.1	**76.3±0.1**	74.0±0.3	75.5±0.1	75.6±0.1	75.1±0.1	76.2±0.1
	5C	75.0±0.1	76.4±0.1	76.4±0.1	76.0±0.1	**77.0±0.1**	74.6±0.3	76.3±0.1	76.2±0.2	75.8±0.2	**77.0±0.1**

a signed directed graph with adjacency matrix

$$\begin{bmatrix} 0 & 0.5 & -0.1 & 3 \\ -3 & 0 & 0 & 3 \\ 3 & 0 & 0 & 0 \\ 0 & -1 & 10 & 0 \end{bmatrix}$$

Corresponding to our tuple definition, we have, for the node corresponding to the first row and first column, $[2, 3]$ for (F,F), $[0, 3.4]$ for (F,T), $[6, 3.6]$ for (F,T'), $[1, 2, 1, 1]$ for (T,F), and $[3, 3.5, 3, 0.1]$ for (T,T).

To see the effect of using edge weights as input instead of treating all weights as having unit-magnitude, we report the main results as well as ablation study results correspondingly in Tables C.4,C.5 and C.6, and their runtimes are reported in Tables C.11,C.12 and C.13. We conclude that using unit-magnitude weights could be either beneficial or harmful depending on the data set and task at hand. However, in general, edge weights are important for *FiLL* but might not be helpful for the bitcoin data sets. Besides, we could draw similar conclusions as in Sec. 4.4.4 for the influence of input features as well as the q value.

Table C.7 compares tue performance of MSGNN with and without symmetric normalization (i.e., whether we use $\mathbf{L}_U^{(q)}$ or $\mathbf{L}_N^{(q)}$) and Table C.8 shows how MSGNN's performance varies with the number of layers. The corresponding runtimes are reported in Tables C.14 and C.15, respectively. In general, we see that there are not significant differences in performance between normalizing and not normalizing; and in the vast majority of cases these differences are less than one standard deviation. We also note that in most cases, adding slightly more layers (from 2 to 4) yields a modest increase in performance. However, we again note that these

Table C.3: Link prediction test performance (accuracy in percentage) comparison for MSGNN with different q values (multiples of $q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$). The best is marked in **bold red** and the second best is marked in underline blue.

Data Set	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
BitCoin-Alpha	SP	71.3±1.2	70.8±1.7	71.4±2.3	**71.5±1.6**	70.8±1.8	71.3±1.0
	DP	69.9±1.6	72.3±2.7	71.1±2.4	72.4±1.6	72.1±1.7	**72.5±1.5**
	3C	84.3±0.5	84.6±0.4	84.5±0.9	84.4±0.5	**84.8±0.9**	84.4±0.6
	4C	58.9±0.6	55.7±1.8	58.1±1.2	58.5±1.5	**59.0±1.3**	58.5±0.7
	5C	82.1±0.5	82.3±0.7	**82.8±0.4**	82.4±0.5	81.9±1.1	81.9±0.9
BitCoin-OTC	SP	**73.0±1.4**	70.9±2.2	72.9±1.4	72.6±1.2	72.9±0.8	72.1±2.5
	DP	**73.6±0.8**	**73.6±2.1**	72.3±1.5	72.9±1.2	72.8±0.4	71.8±1.1
	3C	83.0±0.6	**83.7±0.5**	83.5±0.4	83.1±0.6	83.6±0.6	83.3±0.7
	4C	**61.7±0.7**	60.3±0.6	59.6±0.9	61.5±0.7	59.7±1.6	59.8±0.7
	5C	80.0±1.3	**81.1±1.3**	80.9±0.8	81.0±0.8	**81.1±0.6**	80.9±0.9
Slashdot	SP	92.4±0.2	92.6±0.2	92.9±0.1	92.9±0.1	93.0±0.1	**93.1±0.1**
	DP	92.4±0.2	92.7±0.1	92.9±0.1	92.9±0.1	**93.1±0.1**	**93.1±0.1**
	3C	85.7±0.4	86.0±0.2	**86.3±0.2**	86.2±0.2	86.2±0.2	86.1±0.3
	4C	77.7±0.5	77.7±0.3	77.9±0.4	78.5±0.4	**78.6±0.2**	78.2±0.3
	5C	77.8±0.3	**78.2±0.3**	78.1±0.1	77.6±0.5	77.6±0.4	76.8±0.6
Epinions	SP	85.4±0.5	85.7±0.3	86.0±0.4	**86.5±0.1**	86.2±0.1	86.3±0.1
	DP	85.3±0.7	85.9±0.3	86.2±0.1	86.2±0.1	**86.5±0.2**	86.3±0.3
	3C	82.9±0.4	83.5±0.2	**83.6±0.3**	83.5±0.2	83.2±0.3	83.1±0.5
	4C	80.0±1.0	80.8±0.5	**81.1±0.5**	80.1±0.8	79.7±0.7	78.7±0.9
	5C	80.9±0.5	80.7±0.2	**81.2±0.4**	80.3±0.6	80.8±0.6	80.5±0.5
FiLL (avg.)	SP	**76.2±0.1**	**76.2±0.1**	**76.2±0.1**	76.1±0.0	76.1±0.0	76.1±0.1
	DP	**75.9±0.1**	**75.9±0.1**	**75.9±0.1**	**75.9±0.0**	**75.9±0.0**	**75.9±0.1**
	3C	**76.2±0.1**	**76.2±0.0**	**76.2±0.0**	76.1±0.0	76.1±0.1	76.1±0.0
	4C	**76.3±0.1**	**76.3±0.1**	**76.3±0.0**	**76.3±0.0**	76.2±0.0	76.2±0.1
	5C	**77.0±0.1**	**77.0±0.1**	**77.0±0.1**	**77.0±0.1**	**77.0±0.1**	**77.0±0.1**

Table C.4: Test accuracy (%) comparison the signed and directed link prediction tasks introduced in Sec. 4.4.1 where all networks are treated as unweighted. The best is marked in **bold red** and the second best is marked in underline blue.

Data Set	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
BitCoin-Alpha	SP	64.7±0.9	64.5±1.1	62.9±0.9	64.1±1.3	64.3±2.9	50.2±0.9	**70.3±1.5**
	DP	60.4±1.7	61.5±1.0	61.9±1.9	60.9±1.7	71.8±1.3	51.9±5.6	**74.7±1.5**
	3C	81.4±0.5	79.2±0.9	77.1±0.7	83.2±0.5	79.4±1.3	34.9±17.5	**86.1±0.5**
	4C	51.1±0.8	52.5±1.1	49.3±0.7	52.4±1.8	56.3±1.4	28.6±7.6	**59.5±2.2**
	5C	79.5±0.3	78.2±0.5	76.5±0.3	81.1±0.3	78.8±0.8	25.8±17.0	**83.8±0.8**
BitCoin-OTC	SP	65.6±0.9	65.3±1.2	62.8±1.3	67.7±0.5	68.3±2.5	50.4±5.4	**73.7±1.3**
	DP	63.8±1.2	63.2±1.5	64.0±2.0	65.3±1.2	70.4±1.7	47.3±4.2	**75.6±1.0**
	3C	79.0±0.7	77.3±0.7	73.6±0.7	82.2±0.4	78.0±0.5	35.3±15.3	**85.3±0.4**
	4C	51.5±0.4	55.3±0.8	51.2±1.8	56.9±0.7	60.4±0.9	24.3±6.6	**62.8±1.0**
	5C	77.4±0.7	77.3±0.8	74.1±0.5	80.5±0.5	76.8±0.5	18.9±11.1	**83.0±0.9**
FiLL (avg.)	SP	88.4±0.0	82.0±0.3	76.9±0.1	90.0±0.0	88.7±0.2	51.1±0.7	**90.8±0.0**
	DP	88.5±0.1	82.0±0.2	76.9±0.1	90.0±0.0	86.9±0.6	50.7±1.1	**90.9±0.0**
	3C	63.0±0.1	59.3±0.0	55.3±0.1	64.3±0.1	57.3±0.4	34.1±0.4	**64.6±0.1**
	4C	81.7±0.0	78.8±0.1	70.5±0.1	**83.2±0.1**	76.8±0.1	25.5±1.3	82.1±0.1
	5C	63.8±0.0	61.1±0.1	55.5±0.1	**64.8±0.1**	55.8±0.5	20.5±0.7	62.3±0.2

Table C.5: Link prediction test performance (accuracy in percentage) comparison for variants of MSGNN where input networks are treated as unweighted. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The best is marked in **bold red** and the second best is marked in underline blue .

q value		0					$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Data Set	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
BitCoin-Alpha	SP	72.0±0.9	70.2±1.1	70.9±1.4	72.1±0.4	70.3±1.5	72.0±1.2	70.4±0.6	70.2±2.0	**72.2±1.2**	70.9±1.7
	DP	73.9±0.5	73.9±1.0	72.8±2.3	74.1±1.1	74.5±1.5	73.9±1.2	73.8±1.0	73.3±1.1	73.6±2.4	**74.7±1.5**
	3C	85.4±0.5	85.4±0.3	85.2±0.8	85.5±0.5	85.7±0.3	85.9±0.4	85.9±0.5	85.9±0.5	86.0±0.4	**86.1±0.5**
	4C	57.9±1.9	58.8±1.5	57.7±1.2	58.9±1.1	58.4±1.8	53.6±1.7	54.6±2.2	55.2±1.8	**59.6±2.2**	59.5±2.2
	5C	83.4±0.3	83.1±0.3	83.5±0.5	**84.2±0.5**	83.7±0.5	82.6±0.6	82.8±0.5	82.8±0.7	83.7±0.4	83.8±0.8
BitCoin-OTC	SP	74.5±1.0	71.7±2.3	73.8±1.0	74.1±1.0	73.7±1.3	74.2±1.1	73.4±0.8	73.1±0.9	**74.9±1.0**	73.5±0.6
	DP	75.0±0.5	75.2±1.8	74.7±1.1	75.2±1.4	74.8±2.1	75.6±1.4	75.2±1.1	**75.7±0.9**	74.8±0.8	75.6±1.0
	3C	85.2±0.6	84.7±1.0	85.0±0.5	84.7±0.9	85.1±0.6	85.2±0.6	85.4±0.6	85.3±0.7	**85.5±0.4**	85.3±0.4
	4C	61.0±1.1	61.4±1.9	60.6±1.7	**64.5±2.4**	63.7±1.8	57.8±1.5	56.8±1.4	55.6±1.1	63.3±0.5	62.8±1.0
	5C	82.0±0.6	82.3±0.7	82.1±0.8	82.9±0.7	82.7±0.5	81.0±0.3	80.9±0.5	80.5±0.9	82.6±0.8	**83.0±0.9**
FiLL (avg.)	SP	74.1±0.2	74.0±0.3	74.0±0.4	**75.2±0.1**	**75.2±0.1**	69.3±0.6	69.6±0.2	69.6±0.5	74.8±0.2	74.8±0.2
	DP	73.9±0.1	74.0±0.1	73.9±0.2	75.0±0.1	**75.1±0.1**	70.0±0.2	70.2±0.3	70.3±0.4	74.6±0.1	74.7±0.1
	3C	74.1±0.1	74.1±0.1	74.0±0.2	**75.3±0.1**	75.2±0.1	69.8±0.3	69.5±0.4	69.4±0.5	74.8±0.1	74.8±0.1
	4C	74.1±0.3	74.2±0.2	74.3±0.1	**75.3±0.2**	75.2±0.1	69.1±0.2	68.8±0.4	68.8±0.5	74.8±0.3	74.9±0.3
	5C	75.1±0.1	75.1±0.2	75.0±0.2	76.1±0.1	**76.2±0.1**	70.0±0.2	70.1±0.5	70.3±0.3	75.7±0.2	75.6±0.3

Table C.6: Link prediction test performance (accuracy in percentage) comparison for MSGNN with different q values (multiples of $q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$) when input networks are treated as unweighted. The best is marked in **bold red** and the second best is marked in underline blue .

Data Set	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
BitCoin-Alpha	SP	70.3±1.5	71.0±0.7	70.9±1.5	**71.1±0.9**	69.2±1.4	70.9±1.7
	DP	74.5±1.5	74.0±1.7	74.3±1.2	74.0±1.6	**74.7±0.7**	**74.7±1.5**
	3C	85.7±0.3	86.1±0.4	**86.4±0.3**	85.8±0.6	86.0±0.5	86.1±0.5
	4C	58.4±1.8	59.1±2.0	60.0±1.4	**60.2±1.3**	60.1±1.2	59.5±2.2
	5C	83.7±0.5	84.0±0.5	**84.1±0.4**	**84.1±0.4**	83.8±0.2	83.8±0.8
BitCoin-OTC	SP	**73.7±1.3**	73.2±0.7	**73.7±0.9**	73.0±1.5	73.3±1.1	73.5±0.6
	DP	74.8±2.1	75.1±1.2	**75.8±1.1**	75.3±0.8	75.6±1.0	75.6±1.0
	3C	85.1±0.6	85.1±0.5	**85.3±0.5**	**85.3±0.8**	**85.3±0.6**	**85.3±0.4**
	4C	63.7±1.8	**64.4±0.6**	63.1±2.2	64.3±2.1	63.0±1.2	62.8±1.0
	5C	82.7±0.5	82.8±0.9	82.7±0.7	**83.2±0.8**	82.7±0.9	83.0±0.9
FiLL (avg.)	SP	**75.2±0.1**	**75.2±0.2**	**75.2±0.1**	75.0±0.3	74.8±0.2	74.8±0.2
	DP	**75.1±0.1**	75.0±0.2	74.9±0.1	74.9±0.1	74.6±0.1	74.7±0.1
	3C	**75.2±0.1**	**75.2±0.1**	**75.2±0.1**	**75.2±0.1**	74.9±0.2	74.8±0.1
	4C	75.2±0.1	**75.3±0.1**	75.2±0.1	75.1±0.2	74.9±0.2	74.9±0.3
	5C	**76.2±0.1**	**76.2±0.1**	76.1±0.1	75.9±0.2	75.8±0.1	75.6±0.3

increases in performance are typically quite small and often less than one standard deviation. When we further increase the number of layers (up to 10), performance of MSGNN begin to drop slightly. Overall, we do not see much evidence of severe oversmoothing. However, there does not seem to be significant advantages to very deep networks. Therefore, in our experiments, to be both effective in performance and efficient in computational expense we stick to two layers. (See Tables C.14 and C.15 for runtime comparison.)

It is of interest to explore the behavior of our proposed magnetic signed Laplacian matrix further. Hence here we assess its capability of separating clusters based on its top eigenvectors. Figures C.1 and C.2 show the top eigenvector of our proposed magnetic signed Laplacian matrix with $q = 0.25$ and symmetric normalization, where the x-axis denotes the real parts and the y-axis denotes the imaginary parts,

Table C.7: Link prediction test performance (accuracy in percentage) comparison for MSGNN with various number of layers. The better variant is marked in **bold red** and the worse variant is marked in underline blue .

Data Set	Link Task	no normalization	symmetric normalization
BitCoin-Alpha	SP	**71.9±1.4**	71.3±1.2
	DP	**73.4±1.0**	72.5±1.5
	3C	**84.8±0.8**	84.4±0.6
	4C	57.9±2.2	**58.5±0.7**
	5C	**82.2±0.6**	81.9±0.9
BitCoin-OTC	SP	**74.1±0.7**	73.0±1.4
	DP	**73.5±0.8**	71.8±1.1
	3C	**83.5±0.7**	83.3±0.7
	4C	59.5±1.8	**59.8±0.7**
	5C	80.4±0.8	**80.9±0.9**
Slashdot	SP	**92.7±0.1**	92.4±0.2
	DP	92.8±0.1	**93.1±0.1**
	3C	**86.6±0.1**	86.1±0.3
	4C	77.9±0.9	**78.2±0.3**
	5C	**78.1±0.5**	76.8±0.6
Epinions	SP	**86.1±0.5**	85.4±0.5
	DP	85.8±0.2	**86.3±0.3**
	3C	82.8±1.0	**83.1±0.5**
	4C	**79.7±1.1**	78.7±0.9
	5C	**80.6±0.6**	80.5±0.5
FiLL (avg.)	SP	**76.2±0.1**	76.1±0.1
	DP	**76.0±0.0**	75.9±0.1
	3C	**76.2±0.1**	76.1±0.0
	4C	**76.3±0.1**	76.2±0.1
	5C	**77.0±0.1**	**77.0±0.1**

Table C.8: Link prediction test performance (accuracy in percentage) comparison for MSGNN with various number of layers. The best is marked in **bold red** and the second best is marked in underline blue .

Data Set	Link Task	2 layers	3 layers	4 layers	5 layers	6 layers	7 layers	8 layers	9 layers	10 layers
BitCoin-Alpha	SP	71.3±1.2	71.2±0.6	71.9±1.0	**72.7±0.5**	72.2±0.8	70.9±2.1	70.3±0.7	70.5±0.6	69.4±1.7
	DP	72.5±1.5	72.0±0.9	**73.4±1.0**	**73.4±1.6**	71.0±2.6	70.6±1.2	70.4±1.5	68.3±1.7	69.8±1.7
	3C	84.4±0.6	84.8±0.8	85.0±0.4	**85.1±0.8**	84.8±0.3	84.7±0.2	84.2±0.2	84.6±1.1	84.1±1.1
	4C	58.5±0.7	58.2±2.0	**59.8±2.3**	58.9±1.5	58.5±1.9	58.2±1.2	57.5±2.0	57.7±1.3	57.0±2.0
	5C	81.9±0.9	81.6±1.2	**83.2±0.4**	82.9±0.3	82.4±0.2	82.3±0.6	81.9±0.7	82.2±0.6	81.7±0.4
BitCoin-OTC	SP	73.0±1.4	71.5±1.3	**74.1±1.1**	73.2±1.4	73.2±2.2	71.3±1.1	71.6±0.9	70.3±1.0	69.2±1.9
	DP	71.8±1.1	72.7±0.6	**73.9±1.0**	73.0±1.2	73.7±0.6	72.6±0.7	71.9±1.3	72.2±2.0	71.4±1.7
	3C	83.3±0.7	82.5±1.1	**83.4±0.5**	83.2±0.8	83.2±0.8	83.0±0.7	83.0±0.5	83.0±1.0	82.8±0.2
	4C	59.8±0.7	59.2±1.6	61.2±1.1	**61.6±0.6**	58.7±1.2	57.9±1.7	58.1±1.7	57.9±1.9	54.4±1.9
	5C	**80.9±0.9**	80.1±0.8	80.8±0.6	80.3±1.0	79.9±0.5	79.4±0.9	79.7±0.3	79.5±0.5	79.6±0.5
Slashdot	SP	**92.4±0.2**	92.0±0.3	92.3±0.2	92.0±0.3	91.7±0.3	91.8±0.2	91.3±0.3	91.4±0.3	90.9±0.7
	DP	93.1±0.1	93.0±0.1	**93.2±0.1**	93.1±0.1	93.0±0.2	92.9±0.1	93.1±0.1	93.0±0.2	93.0±0.2
	3C	86.1±0.3	86.0±0.5	**86.2±0.1**	86.1±0.1	85.8±0.2	85.9±0.2	**86.2±0.2**	85.9±0.2	85.8±0.2
	4C	78.2±0.3	78.2±0.4	**78.5±0.5**	78.0±0.2	77.7±0.3	76.9±0.3	77.4±1.1	77.6±0.7	76.9±0.7
	5C	76.8±0.6	77.5±0.2	**77.6±0.4**	77.5±0.4	77.1±0.8	76.8±0.8	76.9±0.5	76.8±0.5	76.4±0.3
Epinions	SP	**85.4±0.5**	85.1±0.9	**85.4±0.5**	**85.4±0.5**	83.7±1.0	83.3±0.9	82.8±0.9	82.8±1.2	80.8±1.0
	DP	86.3±0.3	86.1±0.2	**86.7±0.1**	86.5±0.1	**86.7±0.2**	86.5±0.4	86.6±0.2	**86.7±0.1**	86.5±0.1
	3C	83.1±0.5	83.2±0.2	**83.6±0.2**	**83.6±0.3**	**83.6±0.2**	83.2±0.5	83.4±0.3	83.5±0.3	83.3±0.2
	4C	78.7±0.9	79.8±0.3	79.3±1.2	**80.1±0.4**	**80.1±0.5**	79.0±0.8	79.6±0.5	79.8±0.5	78.8±1.1
	5C	80.5±0.5	79.9±1.0	80.8±0.5	80.7±0.3	80.2±0.8	80.5±0.2	80.8±0.2	**80.9±0.3**	80.2±0.5
FiLL (avg.)	SP	76.1±0.1	76.2±0.0	**76.4±0.0**	76.2±0.1	76.0±0.1	76.0±0.1	76.0±0.0	76.0±0.1	75.8±0.1
	DP	75.9±0.1	76.0±0.1	**76.1±0.1**	76.0±0.1	75.8±0.1	75.8±0.0	75.8±0.1	75.7±0.1	75.6±0.0
	3C	76.1±0.0	76.2±0.1	**76.4±0.1**	76.2±0.0	76.0±0.0	76.0±0.0	76.0±0.1	75.9±0.1	75.8±0.1
	4C	76.2±0.1	76.3±0.0	**76.5±0.1**	76.3±0.0	76.1±0.1	76.1±0.0	76.1±0.0	76.0±0.1	75.8±0.1
	5C	77.0±0.1	77.0±0.1	**77.2±0.1**	77.0±0.1	76.9±0.1	76.8±0.1	76.9±0.1	76.7±0.1	76.6±0.1

Table C.9: Runtime (seconds) comparison on link tasks for variants of MSGNN. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The fastest is marked in **bold red** and the second fastest is marked in underline blue .

q value				0			$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Data Set	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
BitCoin-Alpha	SP	23	**21**	25	25	29	23	**21**	25	25	29
	DP	26	**25**	26	**25**	26	36	38	37	37	37
	3C	**28**	**28**	29	29	29	37	37	37	36	37
	4C	25	25	**24**	26	25	36	36	36	36	36
	5C	29	28	28	**27**	29	37	37	37	37	37
BitCoin-OTC	SP	**26**	27	27	28	30	**26**	27	27	28	30
	DP	28	27	28	28	**26**	38	37	38	37	38
	3C	33	32	32	33	**30**	38	38	38	39	37
	4C	26	**24**	27	27	26	37	37	36	36	37
	5C	32	30	31	31	**29**	38	38	38	37	38
Slashdot	SP	226	227	**221**	224	227	226	227	**221**	224	227
	DP	223	223	**222**	227	**222**	223	223	**222**	227	**222**
	3C	327	327	327	325	**322**	327	327	327	325	**322**
	4C	231	**227**	232	229	232	232	**227**	232	229	232
	5C	330	**324**	325	334	326	330	**324**	325	334	327
Epinions	SP	368	374	**367**	370	370	368	374	**367**	370	370
	DP	**363**	366	364	376	369	364	365	364	376	369
	3C	**506**	510	510	510	509	**506**	510	510	510	510
	4C	**374**	377	382	381	384	**374**	376	382	380	384
	5C	511	511	**508**	518	517	511	511	509	518	517
FiLL (avg.)	SP	36	36	35	36	**32**	36	36	35	36	**32**
	DP	36	36	**35**	**35**	36	36	36	36	37	36
	3C	44	**42**	43	**42**	43	43	44	44	44	43
	4C	35	36	34	34	**31**	35	35	35	36	35
	5C	**43**	45	**43**	**43**	**43**	44	44	44	45	44

for the two synthetic SDSBM models $\mathbf{F_1}(\gamma)$ and $\mathbf{F_1}(\gamma)$ from Subsection 4.4.2. Our Laplacian clearly picks up a signal using the top eigenvector, but does not detect all four clusters. Including information beyond the top eigenvector Figure C.3 reports ARI values when we apply K-means algorithm to the stacked top eigenvectors for clustering. Specifically, for $\mathbf{F}_1$ with three clusters, we stack the real and imaginary parts of the top $3+1=4$ eigenvectors as input features for K-means, while for $\mathbf{F}_2$ we employ the top $4+1=5$ eigenvectors. We conclude that the top eigenvectors of our proposed magnetic signed Laplacian can separate some of the clusters while it confuses a pair, and that the separation ability decreases as we increase the noise level (γ and/or η). In particular, simply using K-means on the top eigenvectors is not competitive compared to the performance of MSGNN after training.

Table C.16 reports input feature sum statistics on real-world data sets: m_1 and s_1 denote the average and one standard error of the sum of input features for each node corresponding to the (T, F) tuple in Table C.2, respectively, while m_2 and s_2 correspond to (T,T). We conclude that in the first four data sets the standard errors are much larger than the averages of the sums of the features, whereas in the *FiLL* data sets, the standard errors are much smaller than the average, see Table C.16. Hence in the *FiLL* data sets these features may not show enough variability around the average to be very informative.

Table C.10: Runtime (seconds) comparison on link tasks for variants of MSGNN with different q values (multiples of $q_0 \coloneqq 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$). The fastest is marked in **bold red** and the second fastest is marked in underline blue.

Data Set	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
BitCoin-Alpha	SP	29	29	29	29	30	**26**
	DP	**26**	31	31	31	33	37
	3C	**29**	37	37	37	37	37
	4C	**25**	31	31	32	32	36
	5C	**29**	36	36	36	36	37
BitCoin-OTC	SP	30	30	30	**27**	**27**	**27**
	DP	**26**	36	36	36	35	38
	3C	**30**	45	45	45	45	37
	4C	**26**	35	35	35	35	37
	5C	**29**	45	45	45	45	38
Slashdot	SP	227	**226**	**226**	**226**	**226**	**226**
	DP	**222**	**222**	**222**	**222**	**222**	**222**
	3C	**322**	**322**	**322**	**322**	**322**	**322**
	4C	**232**	**232**	**232**	**232**	**232**	**232**
	5C	**326**	327	327	327	327	327
Epinions	SP	**370**	**370**	**370**	**370**	**370**	**370**
	DP	**369**	**369**	**369**	**369**	**369**	**369**
	3C	**509**	510	510	510	510	510
	4C	**384**	**384**	**384**	**384**	**384**	**384**
	5C	**517**	**517**	**517**	**517**	**517**	**517**
FiLL (avg.)	SP	**32**	36	36	36	36	36
	DP	**36**	**36**	**36**	**36**	**36**	**36**
	3C	43	44	44	43	**42**	43
	4C	**31**	36	36	36	36	35
	5C	**43**	45	45	44	44	44

C.4 Experimental Results on Individual Years for *FiLL*

Table C.17, C.18, C.19 and C.20 provide full experimental results on financial data sets for individual years for the main experiments, while Table C.21, C.22, C.23, C.24, C.25, C.26, C.27 and C.28 contain individual results for the years for the ablation study.

Table C.11: Runtime (seconds) comparison the signed and directed link prediction tasks introduced in Sec. 4.4.1 where all networks are treated as unweighted. The fastest is marked in **bold red** and the second fastest is marked in underline blue.

Data Set	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
BitCoin-Alpha	SP	352	124	277	438	49	56	**32**
	DP	328	196	432	498	53	59	**34**
	3C	403	150	288	446	50	64	**40**
	4C	385	133	293	471	49	73	**33**
	5C	350	373	468	570	45	60	**40**
BitCoin-OTC	SP	340	140	397	584	50	52	**34**
	DP	471	243	426	941	48	77	**36**
	3C	292	252	502	551	53	58	**44**
	4C	347	143	487	607	47	77	**35**
	5C	460	507	500	959	**45**	52	**45**
FiLL (avg.)	SP	591	320	367	617	99	122	**36**
	DP	387	316	363	386	95	122	**35**
	3C	542	471	298	657	76	76	**43**
	4C	608	384	343	642	76	111	**35**
	5C	318	534	266	521	108	119	**44**

Table C.12: Runtime (seconds) comparison for variants of MSGNN where input networks are treated as unweighted. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The fastest is marked in **bold red** and the second fastest is marked in underline blue.

q value		0					$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Data Set	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
BitCoin-Alpha	SP	33	33	34	34	**32**	33	33	33	33	33
	DP	34	34	34	34	**32**	34	34	34	35	34
	3C	39	40	40	40	**36**	40	40	40	40	40
	4C	33	33	33	33	**31**	33	34	33	34	33
	5C	40	40	40	39	**36**	40	39	40	40	40
BitCoin-OTC	SP	**33**	34	**33**	34	34	**33**	34	34	34	35
	DP	35	**34**	**34**	35	35	35	**34**	**34**	**34**	36
	3C	41	41	**40**	41	**40**	41	**40**	41	41	44
	4C	**32**	33	33	34	33	34	34	34	34	35
	5C	**40**	41	41	**40**	41	41	41	41	41	45
FiLL (avg.)	SP	37	37	37	39	**36**	37	**36**	**36**	38	**36**
	DP	41	37	37	37	**35**	41	36	37	37	**35**
	3C	68	46	45	45	**43**	69	45	45	44	**43**
	4C	36	36	36	37	**35**	**35**	**35**	**35**	36	**35**
	5C	63	46	46	46	**44**	65	**44**	45	45	**44**

Table C.13: Runtime (seconds) comparison for MSGNN with different q values (multiples of $q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$) when input networks are treated as unweighted. The fastest is marked in **bold red** and the second fastest is marked in underline blue.

Data Set	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
BitCoin-Alpha	SP	32	**31**	32	**31**	34	33
	DP	**32**	**32**	**32**	**32**	33	34
	3C	**36**	**36**	**36**	**36**	38	40
	4C	**31**	32	32	**31**	33	33
	5C	**36**	**36**	37	37	40	40
BitCoin-OTC	SP	34	**33**	35	34	36	35
	DP	35	**34**	36	36	36	36
	3C	**40**	41	43	44	45	44
	4C	**33**	**33**	35	35	35	35
	5C	**41**	**41**	45	44	45	45
FiLL (avg.)	SP	**36**	**36**	**36**	**36**	**36**	**36**
	DP	**35**	36	36	**35**	36	**35**
	3C	**43**	44	44	**43**	**43**	**43**
	4C	35	36	36	**34**	35	35
	5C	**44**	45	45	**44**	45	**44**

Table C.14: Runtime (seconds) comparison for MSGNN with various number of layers. The faster variant is marked in **bold red** and the slower variant is marked in underline blue.

Data Set	Link Task	no normalization	symmetric normalization
BitCoin-Alpha	SP	46	**29**
	DP	89	**37**
	3C	101	**37**
	4C	63	**36**
	5C	81	**37**
BitCoin-OTC	SP	47	**30**
	DP	158	**38**
	3C	147	**37**
	4C	63	**37**
	5C	106	**38**
Slashdot	SP	1209	**227**
	DP	1365	**222**
	3C	1475	**322**
	4C	1378	**232**
	5C	1464	**327**
Epinions	SP	1095	**370**
	DP	1103	**369**
	3C	950	**510**
	4C	956	**384**
	5C	1194	**517**
FiLL (avg.)	SP	64	**32**
	DP	76	**36**
	3C	78	**43**
	4C	60	**35**
	5C	81	**44**

Table C.15: Runtime (seconds) comparison for MSGNN with various number of layers. The fastest is marked in **bold red** and the second fastest is marked in underline blue .

Data Set	Link Task	2 layers	3 layers	4 layers	5 layers	6 layers	7 layers	8 layers	9 layers	10 layers
BitCoin-Alpha	SP	**29**	33	36	52	46	50	54	58	61
	DP	**37**	38	**37**	51	46	51	55	58	61
	3C	37	**32**	38	46	44	52	57	60	61
	4C	36	**26**	**26**	48	43	49	53	58	61
	5C	37	**32**	38	50	47	52	57	60	61
BitCoin-OTC	SP	30	29	**28**	44	46	51	55	59	60
	DP	38	**31**	37	43	46	48	55	59	61
	3C	37	**36**	41	46	49	54	58	59	62
	4C	37	**27**	29	45	45	49	56	60	61
	5C	38	**35**	41	47	50	53	58	61	62
Slashdot	SP	**227**	360	374	403	403	500	501	608	610
	DP	**222**	298	299	402	403	497	498	603	604
	3C	**322**	399	399	497	497	597	599	707	708
	4C	**232**	304	314	412	413	503	505	615	617
	5C	**327**	399	402	504	504	600	601	711	713
Epinions	SP	**370**	524	536	696	704	822	837	980	987
	DP	**369**	509	535	687	695	821	838	983	984
	3C	**510**	647	650	814	819	962	968	1122	1127
	4C	**384**	508	519	676	684	812	816	980	986
	5C	**517**	649	655	810	813	967	985	1110	1110
FiLL (avg.)	SP	**32**	42	44	55	58	61	64	70	72
	DP	**36**	43	45	55	59	62	65	70	72
	3C	**43**	50	51	62	65	68	71	77	78
	4C	**35**	41	43	54	57	60	64	70	71
	5C	**44**	50	52	64	66	69	72	77	79

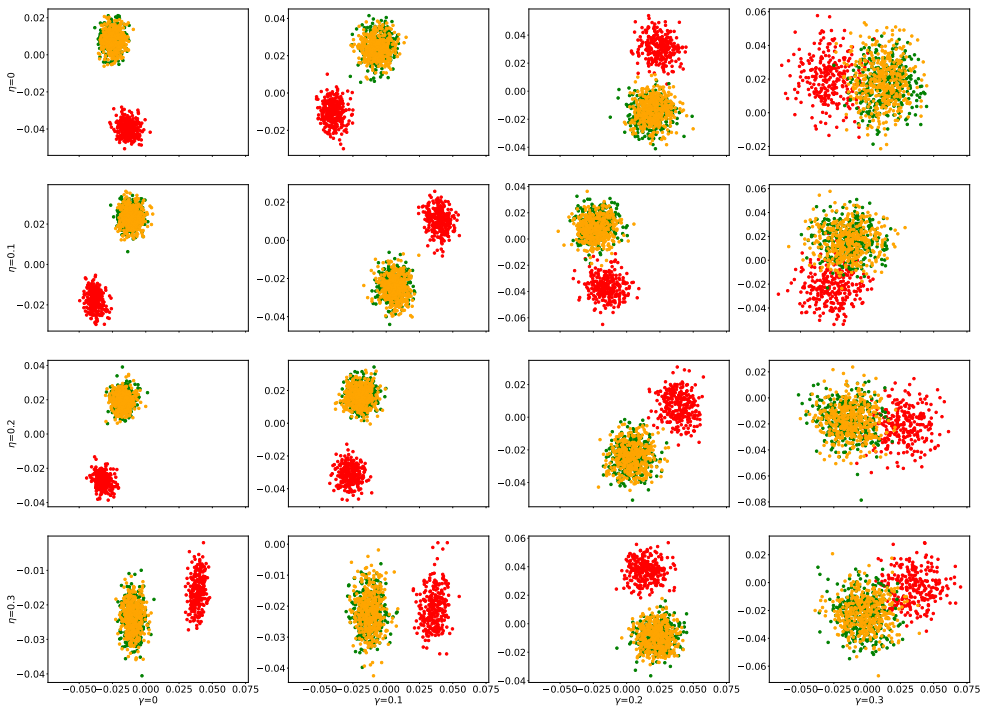


Figure C.1: Real (x-axis) and imaginary parts (y-axis) of the top eigenvector of the symmetric-normalized magnetic signed Laplacian with $q = 0.25$ versus clusters labels on SDSBM($\mathbf{F}_1(\gamma), n = 1000$,
$p = 0.1, \rho = 1.5, \eta$) with various γ and η values, where red, green, and orange denote $\mathcal{C}_0$, $\mathcal{C}_1$, and $\mathcal{C}_2$, respectively.

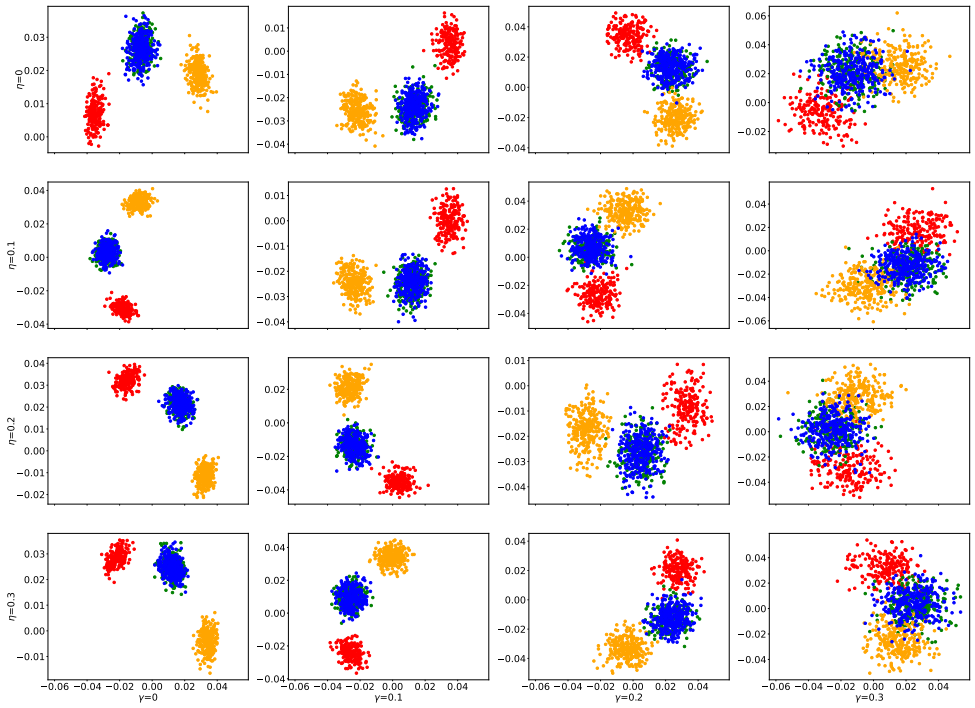
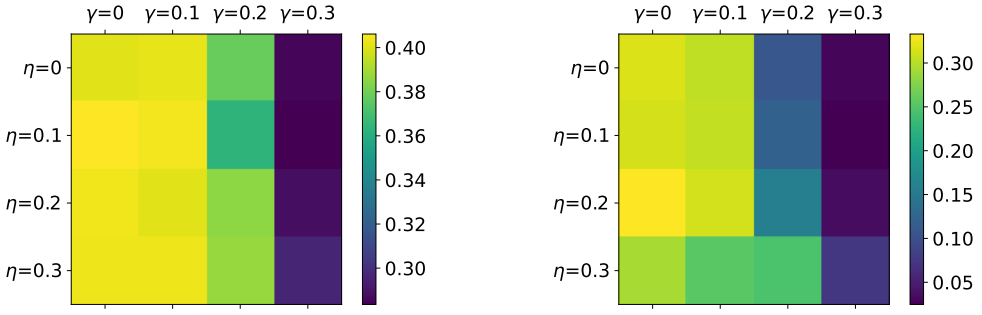


Figure C.2: Real (x-axis) and imaginary parts (y-axis) of the top eigenvector of the symmetric-normalized magnetic signed Laplacian with $q = 0.25$ versus clusters labels on SDSBM($\mathbf{F}_2(\gamma), n = 1000,$
$p = 0.1, \rho = 1.5, \eta$) with various γ and η values, where red, green, orange, and blue denote $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$, respectively.



(a) SDSBM($\mathbf{F}_1(\gamma), n = 1000,$ $p = 0.1, \rho = 1.5, \eta$).

(b) SDSBM($\mathbf{F}_2(\gamma), n = 1000,$ $p = 0.1, \rho = 1.5, \eta$).

Figure C.3: ARI using the proposed magnetic signed Laplacian's top eigenvectors and K-means.

Table C.16: Input feature sum statistics: m_1 and s_1 denote the average and one standard error of the sum of input features for each node corresponding to the (T, F) tuple in Table C.2, respectively, while m_2 and s_2 correspond to (T,T).

Data Set	m_1	s_1	$\frac{m_1}{s_1}$	m_2	s_2	$\frac{m_2}{s_2}$
BitCoin-Alpha	29.076	84.322	0.345	12.787	34.446	0.152
BitCoin-OTC	30.564	102.996	0.297	12.104	38.299	0.118
Slashdot	13.372	44.756	0.299	13.372	44.756	0.299
Epinions	12.765	60.549	0.211	12.765	60.549	0.211
FiLL-pvCLCL (2000)	29.878	9.906	3.016	177.599	40.650	17.928
FiLL-OPCL (2000)	28.182	8.834	3.190	172.000	38.860	19.471
FiLL-pvCLCL (2001)	31.404	11.624	2.702	177.599	44.367	15.278
FiLL-OPCL (2001)	30.026	10.243	2.931	172.000	40.412	16.792
FiLL-pvCLCL (2002)	32.865	19.055	1.725	177.599	64.000	9.320
FiLL-OPCL (2002)	30.154	15.255	1.977	172.000	59.662	11.275
FiLL-pvCLCL (2003)	28.622	12.286	2.330	177.599	59.288	14.456
FiLL-OPCL (2003)	27.085	11.552	2.345	172.000	55.738	14.889
FiLL-pvCLCL (2004)	26.031	8.965	2.904	177.599	45.693	19.810
FiLL-OPCL (2004)	24.633	8.222	2.996	172.000	43.912	20.921
FiLL-pvCLCL (2005)	26.047	8.885	2.932	177.599	47.953	19.988
FiLL-OPCL (2005)	23.884	7.106	3.361	172.000	40.651	24.206
FiLL-pvCLCL (2006)	28.621	11.214	2.552	177.599	50.197	15.837
FiLL-OPCL (2006)	26.219	8.764	2.992	172.000	43.565	19.626
FiLL-pvCLCL (2007)	33.365	18.674	1.787	177.599	80.030	9.510
FiLL-OPCL (2007)	30.564	14.449	2.115	172.000	60.453	11.904
FiLL-pvCLCL (2008)	45.040	31.693	1.421	177.599	97.070	5.604
FiLL-OPCL (2008)	42.205	26.869	1.571	172.000	84.361	6.401
FiLL-pvCLCL (2009)	43.435	31.176	1.393	177.599	109.911	5.697
FiLL-OPCL (2009)	36.304	19.429	1.869	172.000	75.508	8.853
FiLL-pvCLCL (2010)	25.883	14.252	1.816	177.599	67.060	12.461
FiLL-OPCL (2010)	26.908	12.924	2.082	172.000	61.325	13.309
FiLL-pvCLCL (2011)	41.301	31.316	1.319	177.604	116.559	5.671
FiLL-OPCL (2011)	36.046	26.133	1.379	172.000	102.552	6.582
FiLL-pvCLCL (2012)	29.015	13.159	2.205	177.599	54.519	13.496
FiLL-OPCL (2012)	26.120	9.786	2.669	172.000	46.006	17.576
FiLL-pvCLCL (2013)	26.538	13.247	2.003	177.599	69.771	13.407
FiLL-OPCL (2013)	25.754	13.107	1.965	172.000	53.168	13.123
FiLL-pvCLCL (2014)	25.220	9.101	2.771	177.599	48.146	19.514
FiLL-OPCL (2014)	25.366	10.319	2.458	172.000	50.838	16.668
FiLL-pvCLCL (2015)	25.859	14.186	1.823	177.599	57.031	12.519
FiLL-OPCL (2015)	27.853	15.188	1.834	172.000	63.501	11.325
FiLL-pvCLCL (2016)	30.540	18.321	1.667	177.599	54.241	9.694
FiLL-OPCL (2016)	29.418	18.629	1.579	172.000	48.518	9.233
FiLL-pvCLCL (2017)	28.700	10.860	2.643	177.599	39.566	16.353
FiLL-OPCL (2017)	27.020	9.549	2.830	172.000	33.605	18.012
FiLL-pvCLCL (2018)	25.543	10.128	2.522	177.599	51.628	17.535
FiLL-OPCL (2018)	25.859	11.772	2.197	172.000	64.979	14.611
FiLL-pvCLCL (2019)	28.781	13.533	2.127	177.599	47.949	13.123
FiLL-OPCL (2019)	26.188	11.474	2.282	172.000	43.134	14.990

Table C.17: Full link prediction test accuracy (%) comparison for directions (and signs) on *FiLL-pvCLCL* data sets on individual years 2000-2010. The best is marked in **bold red** and the second best is marked in underline blue. The link prediction tasks are introduced in Sec. 4.4.1.

Year	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
2000	SP	87.3±0.3	78.2±2.0	70.9±0.6	88.9±0.2	87.3±2.1	59.1±11.8	**89.0±0.4**
	DP	87.1±0.2	78.6±1.1	70.6±0.7	88.9±0.3	87.7±0.9	53.5±9.4	**89.1±0.5**
	3C	59.8±0.4	53.0±1.2	47.9±0.5	60.8±0.5	58.5±2.2	31.9±7.4	**61.5±0.7**
	4C	71.1±0.4	66.4±1.3	56.5±0.4	**72.5±0.5**	69.0±1.2	23.6±8.0	71.9±0.5
	5C	53.6±0.3	49.8±0.8	43.5±0.2	**54.0±0.3**	51.3±1.5	20.7±7.6	53.9±0.4
2001	SP	88.0±0.3	80.2±1.4	71.5±1.0	90.3±0.2	85.5±3.4	46.2±10.7	**90.7±0.2**
	DP	88.3±0.2	78.9±2.6	70.6±0.5	90.2±0.3	88.4±1.3	47.1±5.6	**90.7±0.1**
	3C	60.3±0.3	54.1±1.0	48.6±0.6	61.7±0.3	58.8±4.8	35.9±6.6	**63.1±0.5**
	4C	74.1±0.3	69.7±1.5	58.4±0.9	**76.4±0.4**	72.1±0.9	23.2±7.9	75.7±0.3
	5C	55.9±0.3	52.5±0.3	45.2±0.4	**57.0±0.2**	54.5±1.4	16.4±4.6	56.8±0.2
2002	SP	88.7±0.2	83.8±1.2	79.1±0.5	90.7±0.2	89.1±1.3	48.4±7.6	**91.4±0.2**
	DP	88.8±0.2	84.4±0.9	78.9±0.7	90.7±0.3	89.8±1.1	51.8±11.7	**91.5±0.2**
	3C	62.1±0.4	60.6±0.6	56.3±0.8	64.2±0.4	63.8±0.2	31.0±1.8	**66.2±0.2**
	4C	84.3±0.5	82.8±0.7	75.3±0.7	85.6±0.3	80.0±2.4	19.7±4.1	**85.7±0.4**
	5C	65.7±0.3	64.2±0.1	58.0±0.9	**67.1±0.1**	57.4±3.4	21.5±3.6	66.7±0.4
2003	SP	86.7±0.6	80.7±1.1	76.4±0.7	87.9±0.5	86.6±1.3	54.8±7.7	**89.5±0.4**
	DP	87.2±0.4	80.9±1.1	76.9±0.6	88.6±0.4	87.6±1.3	44.4±5.4	**89.6±0.4**
	3C	59.5±0.3	56.6±0.9	53.2±0.5	61.1±0.6	58.0±2.2	35.1±7.1	**63.1±0.5**
	4C	80.9±0.3	78.4±0.8	69.6±0.4	82.2±0.3	77.9±3.0	28.9±9.3	**82.7±0.4**
	5C	61.4±0.1	59.6±0.4	53.4±0.6	62.5±0.1	56.4±1.3	17.9±7.3	**62.7±0.4**
2004	SP	86.3±0.3	76.8±2.7	72.4±0.5	88.0±0.3	86.2±1.7	46.3±9.6	**88.7±0.3**
	DP	86.1±0.4	75.2±1.5	72.8±0.5	87.9±0.5	87.2±0.8	50.0±6.7	**88.8±0.3**
	3C	58.7±0.2	50.8±1.2	49.1±0.3	59.7±0.4	59.5±0.9	34.4±2.1	**61.6±0.4**
	4C	77.1±0.3	71.9±1.6	61.1±1.4	**78.9±0.4**	75.9±0.9	19.9±2.1	78.7±0.4
	5C	57.7±0.4	53.8±1.0	47.9±0.6	**58.8±0.4**	55.0±0.7	21.7±3.4	58.7±0.6
2005	SP	85.1±0.2	76.3±1.4	74.9±0.6	86.5±0.6	85.5±1.5	53.0±13.7	**87.8±0.4**
	DP	84.9±0.3	76.3±2.3	74.1±0.7	86.7±0.3	85.9±0.9	47.8±5.2	**87.8±0.4**
	3C	57.1±0.2	53.3±1.2	50.1±0.5	59.1±0.3	57.9±1.0	30.7±4.4	**60.9±0.6**
	4C	79.2±0.4	75.3±1.0	67.4±0.3	**80.5±0.1**	77.6±0.7	20.3±7.3	**80.5±0.2**
	5C	60.4±0.5	57.6±0.4	52.2±0.2	**61.4±0.4**	57.8±1.0	21.1±5.6	61.0±0.2
2006	SP	88.7±0.2	82.7±1.2	75.1±1.0	90.4±0.1	90.0±0.7	38.8±6.5	**91.0±0.2**
	DP	88.8±0.3	83.2±0.9	75.7±0.7	90.6±0.5	89.1±1.2	46.4±7.6	**91.1±0.1**
	3C	61.9±0.3	56.9±1.4	52.8±0.4	63.1±0.4	61.5±1.9	37.4±7.1	**64.1±0.3**
	4C	81.2±0.2	77.8±0.9	66.9±0.6	83.0±0.4	80.6±0.4	25.7±7.3	**83.1±0.4**
	5C	62.1±0.4	58.4±0.7	53.2±0.2	**63.3±0.1**	58.7±1.8	16.2±7.3	62.8±0.3
2007	SP	87.8±0.3	83.9±0.8	79.2±0.3	89.4±0.3	89.3±0.5	55.4±13.3	**90.4±0.5**
	DP	88.3±0.4	83.6±0.4	79.7±0.4	89.7±0.4	88.4±2.0	42.0±12.9	**90.6±0.4**
	3C	65.4±0.6	64.0±0.8	60.6±0.2	67.0±0.4	66.5±0.6	29.6±6.9	**69.0±0.3**
	4C	86.5±0.4	84.2±0.9	77.5±0.2	87.7±0.3	86.2±0.9	23.6±5.9	**88.4±0.2**
	5C	68.8±0.4	66.9±0.6	61.5±0.5	**69.8±0.1**	65.7±2.3	25.5±11.1	69.7±0.2
2008	SP	94.9±0.2	92.5±0.8	83.4±0.8	95.7±0.3	94.2±2.1	45.0±16.6	**96.4±0.2**
	DP	95.2±0.2	93.2±0.4	82.2±0.3	95.9±0.1	95.1±0.8	39.9±18.4	**96.3±0.2**
	3C	75.4±0.5	76.6±0.5	68.3±0.5	77.2±0.5	73.3±3.2	22.4±9.5	**78.9±0.2**
	4C	95.7±0.3	95.5±0.3	87.0±1.0	**96.2±0.2**	95.4±0.4	25.0±16.6	**96.2±0.3**
	5C	80.9±0.2	80.4±0.4	71.1±0.4	81.9±0.1	75.0±4.4	19.4±10.4	**82.0±0.4**
2009	SP	96.0±0.3	91.2±1.4	87.0±0.5	96.9±0.2	96.3±1.1	45.9±13.0	**97.8±0.2**
	DP	96.3±0.3	91.6±0.5	87.2±0.5	97.2±0.1	96.5±0.6	43.0±14.2	**97.6±0.1**
	3C	75.3±0.2	73.1±0.5	70.0±0.3	76.5±0.2	74.5±2.1	37.3±8.5	**78.6±0.3**
	4C	94.1±0.3	92.4±0.4	87.7±0.9	94.8±0.2	93.4±0.4	30.2±10.5	**95.0±0.2**
	5C	78.8±0.3	77.0±0.4	72.8±0.4	79.5±0.3	74.6±3.2	16.7±11.3	**79.8±0.2**
2010	SP	90.9±0.4	85.1±0.7	79.2±0.9	92.1±0.3	90.4±2.4	52.5±10.5	**92.8±0.3**
	DP	91.0±0.2	86.0±1.1	78.4±0.8	91.9±0.3	90.5±0.8	45.8±6.1	**92.7±0.4**
	3C	64.5±0.3	63.1±0.6	56.2±0.9	65.7±0.3	61.6±2.3	33.8±2.2	**68.5±0.4**
	4C	89.8±0.2	88.3±0.4	79.6±0.9	90.3±0.3	87.0±1.2	28.4±5.5	**91.1±0.4**
	5C	71.5±0.3	69.7±0.4	62.6±0.8	72.3±0.2	68.1±1.1	17.8±5.7	**72.4±0.1**

Table C.18: Full link prediction test accuracy (%) comparison for directions (and signs) on *FiLL-pvCLCL* data sets on individual years 2011-2020. The best is marked in **bold red** and the second best is marked in underline blue . The link prediction tasks are introduced in Sec. 4.4.1.

Year	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
2011	SP	97.3±0.2	94.5±0.9	89.3±0.3	97.7±0.2	98.3±0.1	66.2±16.0	**98.7±0.2**
	DP	97.4±0.3	95.4±0.9	89.8±0.5	97.8±0.2	98.1±0.2	45.0±20.6	**98.7±0.1**
	3C	84.3±0.2	82.0±0.4	77.7±0.4	84.7±0.3	82.3±3.1	33.0±17.4	**86.2±0.3**
	4C	97.2±0.1	96.7±0.5	90.0±0.3	97.7±0.1	98.1±0.2	13.9±7.9	**98.3±0.2**
	5C	84.9±0.3	83.2±0.8	78.2±0.4	85.5±0.2	82.0±3.6	18.6±12.3	**87.2±0.4**
2012	SP	90.9±0.4	83.4±1.3	74.5±1.1	**92.7±0.2**	89.0±4.3	42.3±6.0	**92.7±0.3**
	DP	90.8±0.2	83.7±1.8	72.8±1.2	**92.8±0.1**	91.1±1.4	38.1±9.7	92.6±0.3
	3C	64.4±0.2	58.6±1.3	52.1±0.8	65.9±0.3	62.7±0.7	33.1±5.0	**67.4±0.3**
	4C	86.1±0.3	82.0±0.6	69.8±1.2	**87.2±0.4**	85.2±0.4	26.7±19.2	86.9±0.4
	5C	66.2±0.3	62.4±0.7	53.7±0.8	**67.2±0.5**	63.7±1.9	15.2±2.7	67.0±0.2
2013	SP	88.1±0.2	82.1±0.8	80.7±0.4	89.2±0.3	88.8±1.0	43.5±9.2	**90.5±0.3**
	DP	87.5±0.4	82.4±0.7	80.6±0.6	88.7±0.4	87.7±0.6	56.6±14.4	**90.4±0.3**
	3C	63.2±0.2	61.5±0.7	59.1±0.3	64.5±0.2	64.7±0.8	33.1±2.8	**66.1±0.3**
	4C	84.6±0.3	81.6±0.4	75.8±0.4	85.5±0.4	84.2±0.4	26.1±19.6	**86.1±0.3**
	5C	65.7±0.2	64.0±0.4	59.9±0.2	66.5±0.2	64.0±0.6	16.3±7.7	**66.8±0.3**
2014	SP	84.5±0.2	75.9±1.7	70.4±0.6	86.4±0.4	85.5±1.2	49.5±6.1	**87.3±0.2**
	DP	84.3±0.5	75.3±1.2	70.9±0.6	86.4±0.1	84.8±1.6	42.7±11.6	**87.2±0.3**
	3C	57.5±0.2	53.3±0.5	48.7±0.5	59.6±0.3	57.8±2.0	31.6±2.5	**60.6±0.2**
	4C	77.9±0.2	74.5±0.9	63.7±0.8	79.9±0.1	76.3±1.2	29.2±7.7	**80.2±0.2**
	5C	58.7±0.5	56.1±0.9	50.1±1.0	60.1±0.5	56.2±0.8	16.6±4.0	**60.3±0.4**
2015	SP	87.0±0.3	81.6±1.3	75.2±0.8	88.2±0.4	84.5±2.8	50.1±8.5	**89.1±0.4**
	DP	86.9±0.3	80.8±1.4	74.6±0.7	88.0±0.3	86.3±1.4	49.1±12.6	**89.3±0.3**
	3C	60.1±0.2	57.5±0.8	51.8±0.8	60.1±0.8	61.0±1.2	32.6±6.2	**63.8±0.4**
	4C	83.1±0.4	81.0±0.7	69.6±1.0	**84.7±0.5**	79.0±1.8	25.5±11.1	84.4±0.5
	5C	64.6±0.2	62.8±0.5	55.1±0.3	**65.9±0.1**	57.1±5.4	16.7±3.4	65.7±0.4
2016	SP	87.9±0.5	81.6±1.8	73.5±0.9	89.7±0.5	86.5±2.5	49.0±6.5	**90.2±0.5**
	DP	87.4±0.4	81.1±1.4	73.1±0.9	89.6±0.3	87.0±2.7	56.2±6.7	**90.0±0.5**
	3C	60.6±0.3	57.2±0.5	50.9±0.9	62.4±0.4	59.0±1.3	34.4±4.0	**63.9±0.2**
	4C	80.7±0.6	77.5±0.8	67.5±0.3	**82.2±0.6**	74.1±4.8	24.3±3.8	82.1±0.5
	5C	60.8±0.2	58.0±0.8	52.0±0.5	**62.6±0.3**	54.8±2.7	19.7±4.0	62.0±0.4
2017	SP	87.7±0.4	81.7±1.2	78.0±0.4	89.8±0.3	88.3±1.5	57.8±6.8	**90.2±0.3**
	DP	87.5±0.3	82.0±1.2	77.6±0.3	89.8±0.2	87.6±1.7	50.2±4.8	**90.2±0.3**
	3C	59.7±0.4	56.2±0.4	52.8±0.1	61.1±0.3	59.9±0.5	31.3±4.6	**62.3±0.4**
	4C	68.3±0.5	65.4±0.6	59.1±0.6	**70.5±0.3**	66.4±0.5	28.6±7.0	69.6±0.4
	5C	51.7±0.6	50.1±1.0	45.8±0.5	**53.4±0.4**	50.1±1.4	22.3±3.8	53.2±0.1
2018	SP	83.6±0.3	78.7±1.1	72.6±0.5	86.2±0.3	84.3±1.7	47.9±4.2	**87.0±0.4**
	DP	83.7±0.5	78.8±0.8	72.7±0.7	86.3±0.2	84.9±1.9	44.1±5.0	**87.0±0.4**
	3C	57.7±0.3	54.4±0.8	50.7±0.5	58.9±0.6	56.9±2.7	33.4±4.0	**61.4±0.3**
	4C	77.0±0.4	75.8±0.7	65.7±0.7	**79.5±0.4**	77.3±0.6	23.5±5.7	**79.5±0.5**
	5C	59.4±0.1	57.6±0.6	51.6±0.7	**61.0±0.4**	56.4±1.7	21.8±4.8	60.5±0.3
2019	SP	88.5±0.4	80.6±2.0	73.3±0.4	90.3±0.3	87.6±1.6	52.0±10.8	**90.8±0.3**
	DP	88.6±0.4	81.5±1.0	72.6±1.3	90.6±0.1	88.7±0.9	46.4±4.9	**90.9±0.3**
	3C	61.1±0.4	56.0±1.5	50.1±0.9	63.0±0.2	59.6±2.3	33.2±3.1	**64.3±0.2**
	4C	75.3±0.4	71.8±1.2	62.1±0.4	**78.0±0.4**	74.3±1.1	23.1±7.8	77.5±0.4
	5C	57.9±0.3	54.9±0.8	48.1±0.4	**59.6±0.5**	54.3±1.5	21.5±2.6	58.7±0.2
2020	SP	95.7±0.2	93.5±0.8	90.4±0.5	95.6±0.4	96.2±0.5	50.9±27.3	**97.3±0.2**
	DP	95.9±0.1	92.6±1.2	90.2±0.4	95.5±0.4	95.7±0.7	37.1±13.3	**97.2±0.1**
	3C	81.9±0.3	77.8±1.0	76.3±0.3	81.4±0.2	76.0±6.1	43.2±8.1	**82.9±0.5**
	4C	95.1±0.2	93.0±0.9	90.5±0.5	94.7±0.3	95.6±0.5	44.1±20.8	**96.5±0.1**
	5C	82.1±0.2	77.8±1.1	76.4±0.7	81.4±0.4	77.8±3.8	21.4±16.7	**82.8±0.4**

Table C.19: Full link prediction test accuracy (%) comparison for directions (and signs) on *FiLL-OPCL* data sets on individual years 2000-2010. The best is marked in **bold red** and the second best is marked in underline blue . The link prediction tasks are introduced in Sec. 4.4.1.

Year	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
2000	SP	85.7±0.4	77.1±3.1	69.0±1.0	87.5±0.5	87.1±0.5	49.5±8.1	**87.9±0.5**
	DP	85.6±0.2	77.2±1.5	68.4±1.4	87.4±0.6	87.0±0.4	53.6±12.4	**87.9±0.6**
	3C	58.6±0.5	50.7±1.6	46.6±0.8	59.8±0.3	58.9±0.8	32.9±4.9	**60.7±0.4**
	4C	70.2±0.5	65.2±0.8	54.7±1.0	**71.8±0.4**	68.7±0.9	34.8±4.0	71.3±0.3
	5C	52.5±0.5	47.9±0.6	42.8±0.5	53.3±0.7	51.2±0.9	22.6±5.2	**53.4±0.5**
2001	SP	87.6±0.3	79.6±1.8	69.9±1.6	89.8±0.3	88.2±0.9	60.3±6.6	**90.0±0.5**
	DP	87.3±0.5	81.2±1.1	69.6±0.8	89.3±0.5	86.9±2.8	44.0±14.9	**90.2±0.3**
	3C	59.4±0.4	54.8±0.7	47.4±0.3	61.0±0.3	60.5±1.2	32.5±3.6	**62.2±0.4**
	4C	74.2±0.5	69.2±1.7	57.5±0.9	**76.0±0.3**	70.6±2.6	20.8±5.0	75.4±0.6
	5C	54.5±0.2	51.1±0.2	44.5±0.5	**55.9±0.1**	51.4±2.5	18.8±3.8	55.8±0.4
2002	SP	87.3±0.4	82.4±0.3	75.5±0.6	89.6±0.2	85.4±3.2	49.9±12.3	**90.5±0.2**
	DP	87.2±0.2	82.4±1.2	75.8±0.4	89.3±0.3	85.5±3.7	49.1±7.1	**90.5±0.3**
	3C	60.9±0.4	59.4±0.9	53.1±0.8	63.0±0.5	60.5±3.2	35.9±2.8	**65.0±0.4**
	4C	82.7±0.3	81.1±0.3	70.9±0.7	84.1±0.2	80.7±1.9	24.1±7.6	**84.5±0.5**
	5C	63.9±0.4	62.6±0.7	55.4±0.5	65.1±0.2	58.9±2.2	23.0±7.0	**65.5±0.2**
2003	SP	86.0±0.6	80.2±1.3	74.5±0.3	87.7±0.3	87.6±0.3	47.7±8.0	**89.1±0.3**
	DP	85.8±0.4	77.1±1.8	75.3±0.3	87.7±0.4	85.8±2.5	50.3±5.9	**89.2±0.5**
	3C	58.4±0.5	55.7±1.1	51.0±0.6	60.2±0.3	60.2±1.1	33.4±2.4	**62.7±0.4**
	4C	80.3±0.5	78.2±1.3	68.5±0.8	81.9±0.5	79.4±0.6	24.0±10.1	**82.5±0.3**
	5C	60.7±0.4	59.1±0.5	52.1±0.4	61.8±0.3	58.5±1.3	23.7±5.8	**62.3±0.4**
2004	SP	85.2±0.3	74.0±2.4	71.8±0.8	86.8±0.3	86.4±0.9	52.3±8.0	**87.4±0.3**
	DP	85.4±0.4	76.3±2.2	71.9±0.8	87.2±0.2	86.5±0.8	51.0±8.2	**87.4±0.2**
	3C	57.5±0.5	50.8±1.7	48.6±0.6	58.4±0.5	57.0±2.2	33.9±3.7	**60.1±0.7**
	4C	76.8±0.5	72.5±1.7	61.8±0.7	**78.6±0.3**	73.0±2.9	20.1±6.2	78.3±0.3
	5C	57.6±0.3	53.9±0.8	47.5±0.5	**58.4±0.2**	54.0±2.6	18.1±3.6	57.9±0.3
2005	SP	83.8±0.3	73.4±1.1	71.0±1.0	85.4±0.4	85.4±0.4	53.4±7.4	**86.4±0.3**
	DP	83.7±0.6	73.0±2.2	71.8±0.5	85.4±0.6	85.6±0.7	52.9±4.7	**86.4±0.3**
	3C	56.4±0.3	49.7±1.3	48.1±0.4	57.2±0.5	57.3±1.1	32.4±1.9	**59.5±0.6**
	4C	77.3±0.3	72.9±1.0	65.8±0.4	79.2±0.5	75.6±1.3	21.6±5.9	**79.5±0.3**
	5C	58.2±0.4	55.3±0.7	51.3±0.7	**59.4±0.4**	56.0±2.3	12.9±3.1	58.8±0.5
2006	SP	87.7±0.3	77.2±3.2	73.9±1.2	89.2±0.4	88.4±1.0	47.9±7.7	**89.9±0.5**
	DP	87.6±0.4	76.6±2.3	74.1±0.7	89.1±0.3	89.1±0.3	57.0±11.9	**90.1±0.4**
	3C	59.8±0.4	52.5±0.7	51.0±0.4	61.1±0.5	56.7±2.9	33.4±1.5	**63.0±0.5**
	4C	80.5±0.2	74.9±1.4	67.0±0.6	**81.8±0.2**	78.5±1.0	24.7±2.8	81.6±0.3
	5C	60.7±0.5	55.6±0.8	51.4±0.6	**61.7±0.3**	58.0±0.8	21.7±6.9	60.8±0.6
2007	SP	85.4±0.4	77.8±1.3	75.3±0.7	86.7±0.4	85.7±1.4	58.3±7.1	**88.0±0.2**
	DP	86.0±0.3	77.5±1.5	75.5±0.9	86.9±0.4	85.9±1.6	55.1±10.4	**88.0±0.3**
	3C	59.1±0.6	56.4±0.9	53.4±0.8	61.0±0.2	57.8±4.5	30.5±3.2	**63.7±0.5**
	4C	81.6±0.2	78.4±0.7	69.9±0.7	82.5±0.2	79.5±0.7	23.1±9.2	**83.0±0.2**
	5C	63.1±0.3	60.5±0.5	54.9±0.5	63.7±0.5	60.5±0.4	19.4±8.4	**64.1±0.4**
2008	SP	94.7±0.4	92.2±0.6	85.3±0.3	95.6±0.4	95.1±0.4	53.5±15.7	**96.5±0.3**
	DP	94.4±0.4	93.0±0.9	85.4±0.6	95.3±0.2	94.6±1.7	34.1±11.4	**96.6±0.2**
	3C	74.1±0.3	74.1±0.1	67.8±0.5	75.3±0.3	71.7±4.0	36.8±5.9	**76.7±0.2**
	4C	95.0±0.1	94.3±0.4	88.3±0.8	**95.6±0.2**	94.3±0.4	11.4±5.2	95.4±0.5
	5C	78.4±0.3	77.6±0.3	70.6±0.6	**79.4±0.1**	76.1±2.7	13.6±6.6	78.9±0.7
2009	SP	93.4±0.3	83.9±1.9	79.4±0.9	94.4±0.2	93.7±1.1	47.6±9.4	**95.2±0.2**
	DP	93.5±0.2	84.0±2.2	80.0±0.5	94.4±0.3	93.7±0.5	42.0±11.9	**95.2±0.2**
	3C	67.6±0.3	62.6±1.5	56.1±0.4	69.2±0.3	62.4±4.1	38.0±6.2	**70.5±0.2**
	4C	90.3±0.2	86.9±0.5	80.3±0.3	91.1±0.1	88.4±1.4	30.5±13.0	**91.4±0.3**
	5C	72.5±0.4	68.5±0.4	62.7±0.5	**73.3±0.2**	68.0±1.4	21.7±8.9	73.2±0.2
2010	SP	90.5±0.6	85.9±0.6	80.4±0.4	91.8±0.3	90.3±1.1	46.3±11.7	**92.5±0.3**
	DP	90.5±0.6	85.1±0.6	80.0±0.9	91.5±0.3	90.4±1.2	48.7±6.7	**92.5±0.4**
	3C	63.9±0.4	62.6±0.3	57.8±0.6	65.4±0.3	60.7±2.9	33.5±6.7	**67.1±0.5**
	4C	88.3±0.5	85.8±1.0	78.4±0.3	88.8±0.3	85.5±0.8	21.5±3.9	**89.3±0.3**
	5C	69.2±0.6	67.3±1.0	60.6±0.7	**69.9±0.5**	65.8±0.7	19.8±5.2	**69.9±0.3**

Table C.20: Full link prediction test accuracy (%) comparison for directions (and signs) on *FiLL-OPCL* data sets on individual years 2011-2020. The best is marked in **bold red** and the second best is marked in underline blue. The link prediction tasks are introduced in Sec. 4.4.1.

Year	Link Task	SGCN	SDGNN	SiGAT	SNEA	SSSNET	SigMaNet	MSGNN
2011	SP	94.6±0.2	92.2±0.6	84.2±0.2	95.2±0.3	95.4±1.1	39.3±10.0	**96.2±0.3**
	DP	94.9±0.1	91.5±1.3	83.9±0.6	95.4±0.3	95.1±1.1	35.0±16.7	**96.3±0.3**
	3C	75.8±0.4	75.0±0.5	70.1±0.4	76.6±0.3	72.2±6.5	37.1±7.9	**78.6±0.4**
	4C	94.4±0.2	93.6±0.4	85.1±0.4	95.0±0.4	94.2±0.6	33.6±9.6	**95.5±0.3**
	5C	79.7±0.3	78.2±0.4	71.9±0.4	80.2±0.3	77.7±1.9	17.1±8.7	**80.6±0.3**
2012	SP	89.2±0.3	80.9±1.1	80.2±0.3	90.3±0.2	90.3±0.2	45.5±12.5	**91.1±0.2**
	DP	89.4±0.4	82.4±1.0	80.2±0.7	90.3±0.2	89.8±0.5	50.1±6.9	**91.2±0.2**
	3C	61.8±0.3	57.0±1.2	56.2±0.5	62.5±0.2	62.3±0.9	34.5±5.6	**64.4±0.5**
	4C	80.3±0.4	75.9±0.6	71.2±0.6	81.0±0.3	79.1±0.8	28.9±9.9	**81.5±0.4**
	5C	60.9±0.3	57.1±0.5	54.5±0.4	61.4±0.3	57.3±4.0	20.1±6.5	**61.5±0.3**
2013	SP	86.5±0.6	82.3±1.0	79.7±0.3	88.1±0.3	88.0±0.9	52.8±10.0	**89.5±0.3**
	DP	86.9±0.2	81.5±1.4	79.2±0.3	87.9±0.3	86.9±1.9	63.3±13.4	**89.3±0.2**
	3C	61.4±0.4	59.4±0.6	57.7±0.4	61.9±0.2	61.4±1.6	31.4±11.8	**64.3±0.3**
	4C	82.0±0.3	80.3±0.6	72.1±0.4	**83.0±0.2**	80.3±1.2	27.0±16.0	82.8±0.8
	5C	62.6±0.3	61.3±0.4	56.9±0.2	63.2±0.4	60.9±0.9	23.2±6.7	**63.9±0.4**
2014	SP	85.4±0.4	76.8±1.9	72.3±0.6	86.9±0.2	86.1±1.2	50.4±2.4	**87.7±0.4**
	DP	85.2±0.5	76.5±0.7	72.2±0.4	86.7±0.3	84.7±2.5	51.2±6.0	**87.9±0.4**
	3C	58.3±0.5	54.1±1.8	50.3±0.2	59.7±0.8	59.5±0.8	37.2±3.9	**61.9±0.2**
	4C	79.4±0.2	76.0±0.6	68.0±1.0	**81.3±0.2**	78.9±0.7	21.9±9.0	81.2±0.2
	5C	60.4±0.5	57.9±0.4	53.1±0.4	61.7±0.2	58.7±1.0	18.9±4.5	**61.8±0.3**
2015	SP	87.0±0.4	81.5±0.6	78.5±0.7	88.6±0.3	87.0±2.8	41.9±7.1	**89.8±0.3**
	DP	87.2±0.3	81.5±1.0	78.7±0.9	88.8±0.2	85.5±3.0	49.8±5.1	**89.8±0.3**
	3C	60.0±0.2	59.6±0.6	54.4±0.5	61.3±0.4	59.8±2.6	33.9±5.0	**64.1±0.2**
	4C	83.1±0.2	80.3±0.5	72.9±0.6	84.4±0.3	80.8±0.9	20.9±3.7	**84.8±0.2**
	5C	63.7±0.3	62.3±0.6	56.4±0.4	**65.0±0.3**	59.4±2.5	20.7±9.1	64.8±0.5
2016	SP	86.4±0.5	79.1±0.7	75.9±0.6	88.0±0.3	86.6±1.1	58.6±10.2	**89.0±0.2**
	DP	86.5±0.5	78.2±1.0	76.3±0.4	88.2±0.5	86.6±2.3	53.0±2.9	**88.9±0.3**
	3C	59.6±0.6	54.1±0.6	52.6±0.5	60.6±0.3	59.0±1.5	31.5±3.9	**62.2±0.5**
	4C	74.9±0.4	71.0±1.0	64.2±0.3	76.5±0.3	71.5±2.0	24.9±3.8	**76.7±0.5**
	5C	56.5±0.4	54.0±0.7	49.8±0.2	57.0±0.2	50.7±2.1	20.4±6.0	**58.1±0.3**
2017	SP	86.4±0.2	79.3±1.9	75.8±0.3	88.9±0.2	87.9±1.0	53.2±7.6	**89.3±0.3**
	DP	86.3±0.4	78.4±1.6	75.9±1.1	89.1±0.2	88.4±0.4	45.2±9.4	**89.5±0.2**
	3C	58.6±0.2	53.6±0.6	51.5±0.2	60.4±0.2	57.7±1.7	30.0±4.1	**61.3±0.2**
	4C	67.0±0.5	63.2±1.3	56.4±0.4	69.3±0.3	63.8±2.8	26.1±5.3	**69.4±0.4**
	5C	50.2±0.3	47.2±0.9	43.9±0.3	51.7±0.2	49.0±0.8	19.8±5.2	**51.8±0.3**
2018	SP	84.5±0.6	79.3±1.7	69.2±0.6	87.3±0.4	87.0±0.5	59.1±13.4	**88.2±0.4**
	DP	84.7±0.5	77.9±1.0	70.3±0.6	87.4±0.5	87.1±0.5	50.8±6.3	**88.1±0.4**
	3C	59.6±0.3	55.1±1.3	48.6±0.7	61.4±0.5	59.0±3.0	33.2±2.6	**64.0±0.6**
	4C	80.2±0.7	76.3±1.4	63.4±0.7	**83.2±0.6**	78.9±1.2	30.4±7.7	83.0±0.7
	5C	62.4±0.3	58.1±0.7	50.5±0.5	**63.9±0.3**	58.4±4.9	23.2±6.9	63.7±0.5
2019	SP	86.4±0.3	80.8±1.0	77.3±0.3	88.5±0.3	85.8±2.8	41.7±9.4	**89.3±0.4**
	DP	86.5±0.3	80.0±1.3	77.3±0.6	88.7±0.2	88.1±1.1	51.4±8.5	**89.3±0.2**
	3C	59.5±0.4	55.0±0.4	53.4±0.5	60.8±0.2	58.9±2.4	33.6±5.8	**62.4±0.5**
	4C	71.2±0.5	68.1±0.5	63.1±0.4	**74.3±0.3**	68.9±2.2	26.8±9.5	**74.3±0.4**
	5C	54.4±0.3	52.0±0.5	48.8±0.3	**56.2±0.3**	52.7±0.5	21.8±7.1	56.0±0.2
2020	SP	89.8±0.3	84.4±0.7	84.6±0.4	90.9±0.3	90.4±0.7	53.9±16.9	**92.3±0.1**
	DP	90.1±0.2	85.4±0.7	84.7±0.4	91.0±0.3	89.6±1.2	53.5±14.0	**92.1±0.1**
	3C	66.8±0.4	62.3±0.4	62.4±0.4	67.8±0.2	63.6±4.1	41.5±4.6	**69.2±0.4**
	4C	84.0±0.4	81.5±0.5	78.5±0.4	**85.1±0.3**	82.6±1.2	16.3±9.3	84.8±0.4
	5C	66.8±0.5	63.6±0.6	61.1±0.2	**68.1±0.4**	64.2±1.3	24.4±10.1	67.3±0.6

Table C.21: Link prediction test performance (accuracy in percentage) comparison for variants of MSGNN for individual years 2000-2010 of the *FiLL-pvCLCL* data set. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The best is marked in **bold red** and the second best is marked in underline blue .

	q value			0			$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Year	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
2000	SP	**89.0±0.4**	**89.0±0.5**	**89.0±0.4**	**89.0±0.4**	**89.0±0.4**	88.9±0.6	88.9±0.4	88.9±0.3	88.7±0.4	88.8±0.5
	DP	88.9±0.4	89.0±0.5	89.0±0.5	89.0±0.4	**89.1±0.4**	**89.1±0.4**	**89.1±0.4**	**89.1±0.4**	**89.1±0.6**	**89.1±0.5**
	3C	60.7±0.6	61.1±0.7	61.0±0.5	60.7±0.7	61.3±0.4	60.4±0.4	61.0±0.4	61.4±0.8	60.6±0.6	**61.5±0.7**
	4C	68.7±0.8	71.0±0.5	70.9±0.7	70.7±0.6	**72.3±0.5**	67.9±2.3	70.5±0.7	70.8±0.8	71.2±0.6	71.9±0.5
	5C	49.9±2.3	53.1±0.2	52.7±0.7	51.7±0.5	**54.0±0.4**	50.4±0.7	52.9±0.4	52.6±0.5	51.4±2.5	53.9±0.4
2001	SP	90.6±0.4	**90.7±0.3**	**90.7±0.1**	90.3±0.2	**90.7±0.2**	90.4±0.2	90.5±0.2	90.5±0.2	90.0±0.5	**90.7±0.2**
	DP	90.6±0.4	90.5±0.2	90.6±0.2	90.4±0.3	**90.7±0.2**	**90.7±0.4**	**90.7±0.3**	**90.7±0.2**	90.6±0.3	**90.7±0.1**
	3C	62.0±0.4	62.3±0.1	62.4±0.2	61.9±0.3	62.6±0.2	62.1±0.3	62.6±0.4	62.5±0.3	62.2±0.4	**63.1±0.5**
	4C	71.9±1.0	74.6±0.4	74.6±0.5	74.6±0.6	**75.8±0.3**	72.2±1.2	74.5±0.7	74.4±0.5	74.9±0.6	75.7±0.3
	5C	52.0±0.6	55.7±0.3	55.3±1.0	55.4±1.1	56.7±0.4	51.5±1.4	55.4±0.6	56.0±0.3	53.7±1.8	**56.8±0.2**
2002	SP	91.4±0.3	91.4±0.2	**91.5±0.2**	91.4±0.2	91.4±0.2	91.1±0.3	91.2±0.1	91.2±0.2	91.0±0.3	91.2±0.3
	DP	91.1±0.5	91.4±0.2	**91.5±0.2**	91.3±0.2	**91.5±0.1**	**91.5±0.3**	91.3±0.1	**91.5±0.2**	91.3±0.2	**91.5±0.2**
	3C	64.7±0.3	65.5±0.5	65.5±0.5	64.6±0.6	**66.2±0.4**	64.2±0.7	65.7±0.3	65.4±0.4	65.2±0.1	**66.2±0.2**
	4C	83.4±1.1	84.4±0.3	84.7±0.4	84.4±0.6	**85.7±0.5**	83.3±1.0	84.7±0.2	84.6±0.4	84.6±0.8	**85.7±0.4**
	5C	60.8±4.3	65.5±0.4	65.6±0.3	65.6±0.5	**66.8±0.4**	63.7±1.1	65.8±0.3	65.2±0.2	64.8±0.8	66.7±0.4
2003	SP	89.3±0.5	89.3±0.3	89.4±0.3	89.3±0.4	**89.5±0.4**	89.3±0.3	89.3±0.3	89.3±0.4	88.9±0.4	89.3±0.4
	DP	89.4±0.4	89.4±0.4	89.5±0.4	89.4±0.5	**89.6±0.4**	89.5±0.5	89.4±0.2	89.5±0.4	**89.6±0.3**	**89.6±0.4**
	3C	60.2±1.0	62.4±0.8	62.3±1.0	61.4±0.6	**63.2±0.5**	61.2±1.1	62.9±0.7	62.9±0.5	61.7±0.4	63.1±0.5
	4C	80.3±0.6	81.9±0.2	81.9±0.3	81.5±0.6	82.6±0.4	79.7±0.8	81.8±0.5	81.6±0.2	81.4±0.9	**82.7±0.4**
	5C	57.0±2.0	61.9±0.3	61.5±0.5	60.0±0.6	62.6±0.2	57.4±1.4	61.4±0.3	61.5±0.3	61.2±0.5	**62.7±0.4**
2004	SP	88.6±0.4	88.6±0.4	88.7±0.2	88.6±0.2	88.7±0.3	88.4±0.6	88.7±0.3	88.6±0.3	88.3±0.3	**88.8±0.3**
	DP	88.5±0.2	**88.8±0.3**	**88.8±0.2**	88.7±0.3	**88.8±0.3**	**88.8±0.3**	88.7±0.4	88.7±0.3	88.7±0.2	**88.8±0.3**
	3C	60.2±0.7	61.4±0.4	61.3±0.6	60.5±0.3	**61.6±0.5**	59.4±1.0	**61.6±0.5**	61.3±0.3	60.4±0.5	**61.6±0.4**
	4C	75.5±0.9	78.0±0.3	77.7±0.6	77.3±0.6	**78.8±0.6**	75.9±0.5	77.5±0.7	77.6±0.5	77.4±0.4	78.7±0.4
	5C	55.5±0.7	58.1±0.6	58.0±0.3	55.8±2.8	**58.7±0.3**	53.8±0.8	57.3±0.5	57.5±0.3	55.5±2.4	**58.7±0.6**
2005	SP	87.7±0.4	87.7±0.5	87.6±0.5	87.5±0.5	**87.8±0.4**	87.5±0.4	87.7±0.6	87.6±0.5	87.3±0.5	87.7±0.4
	DP	87.6±0.5	87.8±0.4	87.7±0.6	87.5±0.5	**87.9±0.4**	87.3±0.8	87.8±0.4	87.8±0.5	87.8±0.4	87.8±0.4
	3C	59.2±0.7	60.3±0.3	60.3±0.5	59.6±0.8	**61.2±0.2**	59.3±0.4	60.8±0.5	60.7±0.6	59.6±0.3	60.9±0.6
	4C	78.2±0.4	79.7±0.3	79.7±0.4	79.0±1.2	**80.8±0.2**	78.1±0.4	79.1±0.5	79.4±0.3	80.3±0.3	80.5±0.2
	5C	57.1±1.3	60.1±0.3	60.1±0.3	59.3±0.9	**61.3±0.4**	55.1±1.6	59.4±0.5	59.4±0.4	59.6±0.4	61.0±0.2
2006	SP	90.9±0.2	90.9±0.1	**91.0±0.1**	90.9±0.2	**91.0±0.2**	90.5±0.2	90.6±0.3	90.5±0.4	90.5±0.1	90.6±0.1
	DP	90.8±0.2	90.9±0.1	91.0±0.2	91.0±0.2	91.1±0.2	91.0±0.1	**91.2±0.1**	91.0±0.2	91.0±0.1	91.1±0.1
	3C	63.2±0.3	63.4±0.4	63.5±0.3	63.2±0.3	64.0±0.3	61.8±2.0	**64.1±0.4**	64.0±0.4	63.0±0.4	**64.1±0.3**
	4C	80.1±0.8	81.2±0.3	81.4±0.6	81.5±0.7	82.9±0.2	79.8±0.5	81.0±0.7	81.4±0.5	81.7±0.9	**83.1±0.4**
	5C	58.1±0.7	61.6±0.3	61.6±0.5	61.3±0.7	62.6±0.3	59.4±1.1	61.1±0.7	61.4±0.3	61.5±0.7	**62.8±0.3**
2007	SP	90.2±0.4	90.3±0.3	90.3±0.4	**90.4±0.3**	**90.4±0.5**	89.7±0.1	89.9±0.2	90.0±0.2	89.6±0.4	90.0±0.4
	DP	90.3±0.3	90.4±0.3	90.2±0.3	90.3±0.4	90.4±0.3	90.2±0.4	90.4±0.3	90.3±0.3	90.4±0.4	**90.6±0.4**
	3C	66.1±1.4	68.2±0.4	68.4±0.2	67.0±0.4	**69.0±0.4**	64.8±1.0	**69.0±0.4**	68.3±0.7	65.1±2.2	**69.0±0.3**
	4C	85.3±0.5	87.3±0.3	86.9±0.5	87.4±0.6	88.1±0.2	85.5±0.4	87.3±0.1	86.7±0.7	86.9±0.8	**88.4±0.2**
	5C	66.4±1.5	69.3±0.4	69.3±0.3	68.1±1.0	**69.9±0.5**	64.2±0.6	68.1±1.0	68.3±0.6	68.2±1.2	69.7±0.2
2008	SP	96.1±0.1	96.2±0.2	96.1±0.2	96.2±0.3	**96.4±0.2**	95.4±0.2	95.6±0.2	95.5±0.2	95.5±0.1	95.7±0.3
	DP	96.3±0.1	96.3±0.1	96.1±0.2	96.2±0.3	**96.4±0.1**	96.2±0.1	**96.4±0.1**	96.3±0.2	96.2±0.2	96.3±0.2
	3C	76.7±1.3	78.8±0.3	78.7±0.5	77.7±0.7	**79.3±0.7**	76.0±1.5	78.2±0.3	78.4±0.6	76.9±0.7	78.9±0.2
	4C	94.5±1.1	95.6±0.2	95.4±0.3	95.9±0.3	96.1±0.2	93.6±0.8	95.1±0.4	94.4±1.3	95.3±0.2	**96.2±0.3**
	5C	78.7±0.5	81.1±0.5	80.8±0.5	79.3±1.0	**82.4±0.4**	78.1±0.8	79.4±0.8	79.7±0.5	79.9±0.7	82.0±0.4
2009	SP	97.5±0.2	97.5±0.2	97.6±0.2	97.7±0.2	**97.8±0.2**	96.9±0.1	96.7±0.2	96.6±0.7	97.0±0.4	97.1±0.3
	DP	97.5±0.2	97.6±0.1	97.6±0.2	97.6±0.1	**97.8±0.1**	97.5±0.2	97.6±0.2	97.6±0.2	97.5±0.2	97.6±0.1
	3C	75.9±1.2	77.2±0.7	78.3±0.2	76.9±0.4	78.4±0.4	76.5±1.0	77.8±0.6	77.7±0.4	77.5±0.3	**78.6±0.3**
	4C	93.5±0.5	94.6±0.2	94.7±0.2	94.0±0.5	**95.2±0.3**	93.7±0.3	94.6±0.2	94.4±0.3	94.2±0.2	95.0±0.2
	5C	77.9±0.5	79.2±0.2	79.4±0.6	78.3±0.6	**79.9±0.2**	76.6±0.6	78.1±0.8	77.4±1.6	78.2±0.6	79.8±0.2
2010	SP	92.4±0.4	92.7±0.3	92.6±0.3	92.7±0.4	**92.8±0.3**	92.1±0.3	92.5±0.2	92.5±0.3	92.0±0.4	92.3±0.3
	DP	92.5±0.3	92.7±0.2	92.7±0.3	92.7±0.4	92.6±0.3	92.7±0.4	**92.8±0.4**	92.6±0.3	92.6±0.3	92.7±0.4
	3C	66.3±0.4	67.9±0.3	67.6±0.6	66.3±1.0	68.3±0.2	66.2±0.7	67.9±0.2	67.7±0.5	65.7±1.6	**68.5±0.4**
	4C	88.8±0.6	90.4±0.2	90.4±0.3	90.1±0.5	90.9±0.4	89.2±0.8	90.0±0.4	90.3±0.4	90.3±0.2	**91.1±0.4**
	5C	67.5±1.3	71.4±0.3	71.4±0.5	70.1±1.2	**72.5±0.4**	68.5±2.3	71.1±0.5	71.1±0.6	71.5±0.2	72.4±0.1

Table C.22: Link prediction test performance (accuracy in percentage) comparison for variants of MSGNN for individual years 2011-2020 of the *FiLL-pvCLCL* data set. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The best is marked in **bold red** and the second best is marked in underline blue.

	q value			0					$q_0 \coloneqq 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$			
Year	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
2011	SP	98.4±0.3	**98.7±0.2**	98.6±0.1	98.6±0.1	**98.7±0.2**	97.8±0.5	98.0±0.2	98.0±0.2	98.3±0.2	98.3±0.3
	DP	98.6±0.2	98.5±0.3	98.5±0.3	98.6±0.1	98.7±0.2	98.5±0.1	98.7±0.2	98.6±0.1	**98.8±0.1**	98.7±0.1
	3C	84.5±1.4	**86.7±0.2**	86.4±0.4	84.0±1.2	86.5±0.2	83.8±1.1	85.5±1.5	85.9±0.4	84.2±0.8	86.2±0.3
	4C	97.7±0.6	98.2±0.3	98.1±0.2	98.1±0.2	**98.3±0.3**	97.7±0.2	98.1±0.2	98.0±0.3	97.9±0.4	**98.3±0.2**
	5C	85.3±0.8	87.0±0.4	**87.4±0.4**	86.0±0.9	**87.4±0.3**	83.2±2.3	86.6±0.4	86.8±0.5	85.6±0.8	87.2±0.4
2012	SP	92.4±0.3	92.6±0.3	92.6±0.4	92.5±0.4	**92.7±0.3**	92.4±0.4	92.5±0.3	92.4±0.3	92.2±0.5	92.4±0.3
	DP	92.5±0.4	92.4±0.4	92.5±0.4	92.5±0.4	**92.6±0.2**	92.5±0.2	92.5±0.4	92.5±0.4	92.4±0.2	**92.6±0.3**
	3C	65.8±0.3	66.6±0.3	66.5±0.1	66.5±0.5	67.1±0.2	65.2±1.1	66.5±0.2	66.8±0.5	66.2±0.2	**67.4±0.3**
	4C	84.9±0.7	85.9±0.7	85.9±0.4	85.6±0.8	**87.0±0.5**	84.4±0.7	86.1±0.3	86.1±0.6	85.8±0.6	86.9±0.4
	5C	63.4±0.5	65.7±0.3	65.5±0.4	65.7±0.9	66.8±0.1	62.8±1.2	65.6±0.5	66.0±0.3	64.4±1.5	**67.0±0.2**
2013	SP	90.3±0.3	90.3±0.2	90.3±0.4	90.4±0.2	**90.5±0.3**	90.0±0.2	90.1±0.2	90.3±0.3	89.8±0.3	90.3±0.1
	DP	90.2±0.4	90.4±0.3	90.3±0.3	90.3±0.3	90.3±0.3	90.2±0.2	**90.5±0.3**	**90.5±0.3**	**90.5±0.2**	90.4±0.3
	3C	64.7±0.7	66.3±0.2	66.5±0.3	65.4±0.5	**66.9±0.2**	62.0±2.6	65.9±0.5	66.0±0.5	64.4±0.9	66.1±0.3
	4C	83.4±0.8	85.0±0.2	85.0±0.4	85.3±0.6	85.9±0.3	83.4±1.0	84.3±1.0	84.9±0.1	85.3±0.6	**86.1±0.3**
	5C	62.5±1.9	66.2±0.3	66.3±0.3	65.3±0.7	**67.0±0.3**	62.1±0.7	65.6±0.4	65.6±0.6	64.5±1.0	66.8±0.3
2014	SP	**87.3±0.3**	**87.3±0.3**	**87.3±0.2**	87.2±0.2	**87.3±0.2**	86.9±0.3	87.1±0.2	87.0±0.2	86.8±0.4	87.1±0.2
	DP	87.1±0.4	87.1±0.3	87.2±0.3	87.2±0.2	**87.3±0.2**	87.1±0.2	87.0±0.1	87.2±0.2	87.2±0.3	87.2±0.3
	3C	59.6±0.6	60.3±0.2	60.4±0.1	59.8±0.5	**60.8±0.3**	59.4±0.2	60.2±0.2	60.3±0.4	59.7±0.6	60.6±0.2
	4C	77.5±0.7	79.0±0.2	79.0±0.2	79.0±0.8	**80.2±0.3**	77.5±0.6	79.1±0.3	79.0±0.4	79.0±0.4	**80.2±0.2**
	5C	57.0±0.6	59.2±0.2	59.3±0.3	58.9±1.0	**60.3±0.4**	56.8±0.9	59.6±0.4	59.3±0.2	57.7±1.2	**60.3±0.4**
2015	SP	89.0±0.3	**89.2±0.3**	89.1±0.4	89.1±0.3	89.1±0.4	88.9±0.2	88.7±0.2	88.7±0.2	88.7±0.3	88.8±0.2
	DP	89.2±0.3	89.2±0.3	89.1±0.4	**89.3±0.4**	89.2±0.4	89.0±0.2	89.2±0.4	89.2±0.4	89.2±0.4	**89.3±0.3**
	3C	62.3±0.4	63.4±0.3	63.2±0.3	63.0±0.7	63.5±0.5	62.5±0.4	63.3±0.2	63.4±0.3	62.9±0.3	**63.8±0.4**
	4C	82.3±0.5	83.5±0.5	83.2±0.6	84.0±0.7	**84.6±0.5**	81.7±1.2	83.3±0.4	83.6±0.3	84.1±0.4	84.4±0.5
	5C	61.2±1.5	64.3±0.4	63.8±0.5	63.2±1.2	65.6±0.4	61.4±0.8	64.2±0.4	64.0±0.4	63.8±0.8	**65.7±0.4**
2016	SP	90.0±0.4	89.9±0.4	90.0±0.4	90.1±0.4	**90.2±0.5**	89.9±0.2	89.2±1.2	89.7±0.4	89.8±0.3	89.7±0.5
	DP	89.9±0.6	90.0±0.4	89.9±0.4	**90.1±0.5**	**90.1±0.5**	**90.1±0.5**	90.0±0.4	90.0±0.5	**90.1±0.4**	90.0±0.5
	3C	62.9±0.5	63.7±0.1	63.6±0.4	62.4±0.8	**64.0±0.3**	62.5±0.7	63.5±0.4	63.5±0.2	62.8±0.7	63.9±0.2
	4C	78.9±0.5	80.7±0.5	80.5±0.6	80.9±0.3	82.0±0.5	77.9±2.2	80.8±0.5	80.6±0.6	81.0±1.1	**82.1±0.5**
	5C	56.9±2.5	60.8±0.5	60.8±0.4	60.8±1.0	**62.2±0.3**	57.8±1.7	61.1±0.3	61.1±0.3	60.8±0.7	62.0±0.4
2017	SP	90.1±0.5	89.9±0.5	90.1±0.4	**90.2±0.3**	**90.2±0.3**	89.9±0.5	89.6±1.2	90.1±0.4	89.6±0.6	90.0±0.3
	DP	90.0±0.4	90.1±0.4	90.1±0.3	90.1±0.3	90.1±0.2	**90.2±0.4**	**90.2±0.4**	90.1±0.3	**90.2±0.4**	**90.2±0.3**
	3C	61.4±0.6	62.0±0.4	62.1±0.3	61.6±0.2	62.0±0.1	61.1±0.8	62.3±0.4	**62.4±0.5**	62.2±0.5	62.3±0.4
	4C	65.4±0.8	68.2±0.5	68.5±0.4	68.6±0.3	**70.1±0.5**	66.4±1.3	68.7±0.4	68.8±0.2	68.7±0.7	69.6±0.4
	5C	50.1±0.5	52.3±0.4	52.8±0.4	51.5±0.7	**53.4±0.2**	46.0±4.7	52.6±0.5	52.6±0.2	51.3±0.8	53.2±0.1
2018	SP	86.9±0.4	86.8±0.5	86.9±0.4	86.8±0.4	**87.0±0.4**	86.6±0.7	86.5±0.3	86.5±0.4	86.5±0.5	86.6±0.8
	DP	86.9±0.5	86.8±0.4	86.8±0.4	86.7±0.4	86.9±0.4	86.7±0.3	86.9±0.4	86.9±0.5	86.8±0.5	**87.0±0.4**
	3C	59.7±1.3	60.9±0.5	61.0±0.6	60.3±0.5	**61.5±0.4**	60.0±0.5	61.3±0.3	**61.5±0.5**	60.8±0.3	61.4±0.3
	4C	75.8±1.2	78.4±0.5	78.5±0.6	78.4±1.2	**79.7±0.4**	75.4±3.1	78.2±0.4	78.1±0.6	78.0±0.9	79.5±0.5
	5C	57.5±0.8	59.2±0.4	59.3±0.2	58.7±1.9	**60.9±0.4**	56.2±1.4	59.4±0.2	59.3±0.8	59.1±0.5	60.5±0.3
2019	SP	90.8±0.3	90.8±0.4	90.8±0.3	**90.9±0.3**	90.8±0.3	90.6±0.5	90.6±0.4	90.6±0.4	90.4±0.1	90.6±0.4
	DP	90.8±0.3	90.8±0.3	90.8±0.2	90.7±0.3	**90.9±0.3**	90.6±0.3	90.8±0.3	90.7±0.3	**90.9±0.3**	**90.9±0.3**
	3C	63.1±0.7	63.9±0.3	64.0±0.2	63.4±1.2	**64.6±0.2**	63.5±0.3	64.2±0.2	64.2±0.2	63.4±0.6	64.3±0.2
	4C	74.3±0.8	76.1±0.3	76.1±0.3	77.1±0.6	**77.9±0.3**	75.0±0.4	76.3±0.5	76.5±0.4	77.1±0.3	77.5±0.4
	5C	55.2±1.8	58.0±0.3	57.9±0.5	57.8±0.8	**59.0±0.2**	54.5±2.9	57.8±0.4	57.6±0.5	55.6±1.9	58.7±0.2
2020	SP	97.1±0.2	97.1±0.2	97.2±0.1	**97.3±0.2**	**97.3±0.2**	96.4±0.2	96.4±0.1	96.4±0.2	96.7±0.1	96.6±0.2
	DP	**97.3±0.1**	97.2±0.1	97.1±0.1	**97.3±0.2**	**97.3±0.2**	97.2±0.1	97.1±0.2	97.1±0.2	97.1±0.1	97.2±0.1
	3C	81.8±0.6	**83.2±0.1**	83.1±0.2	81.6±1.5	82.8±1.3	80.1±1.4	82.8±0.6	82.6±0.6	82.2±0.4	82.9±0.5
	4C	96.2±0.2	96.2±0.2	**96.5±0.2**	96.4±0.5	96.4±0.2	96.0±0.2	96.1±0.6	96.1±0.2	96.2±0.2	**96.5±0.1**
	5C	80.3±1.6	82.9±0.3	82.9±0.3	82.1±0.5	**83.2±0.4**	78.8±3.8	82.6±0.6	82.7±0.2	82.1±0.5	82.8±0.4

Table C.23: Link prediction test performance (accuracy in percentage) comparison for variants of MSGNN for individual years 2000-2010 of the *FiLL-OPCL* data set. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The best is marked in **bold red** and the second best is marked in underline blue .

q value		0					$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Year	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
2000	SP	87.5±0.5	87.6±0.5	87.7±0.5	87.7±0.6	**87.9±0.5**	87.2±0.5	87.6±0.6	87.6±0.5	87.4±0.5	87.6±0.5
	DP	87.6±0.6	87.6±0.6	87.5±0.5	87.7±0.5	**87.9±0.5**	87.6±0.6	87.8±0.5	**87.9±0.6**	87.7±0.4	**87.9±0.6**
	3C	59.5±0.9	60.0±0.4	60.2±0.5	59.6±0.8	**60.7±0.3**	59.4±1.2	60.6±0.6	60.6±0.4	59.7±0.5	**60.7±0.4**
	4C	68.4±1.2	69.9±0.4	70.1±0.3	70.3±0.8	**71.3±0.3**	68.1±0.8	70.0±0.6	70.0±0.5	70.5±0.5	**71.3±0.3**
	5C	49.8±2.1	52.3±0.4	52.4±0.5	51.1±1.0	**53.5±0.4**	49.0±1.6	52.4±0.6	52.1±0.3	51.6±1.0	53.4±0.5
2001	SP	89.7±0.4	89.9±0.4	89.9±0.3	89.9±0.3	**90.0±0.5**	89.7±0.5	89.8±0.4	89.8±0.3	89.4±0.4	89.9±0.4
	DP	89.9±0.3	89.9±0.4	90.0±0.3	89.9±0.3	90.0±0.3	89.8±0.4	90.1±0.3	90.0±0.4	89.9±0.3	**90.2±0.3**
	3C	60.4±0.5	61.6±0.4	61.5±0.4	61.2±0.6	61.9±0.3	61.1±0.3	61.8±0.5	62.0±0.4	61.2±0.5	**62.2±0.4**
	4C	71.9±0.6	74.3±0.7	74.1±0.6	74.5±0.9	**75.5±0.3**	71.9±1.4	74.3±0.7	74.0±0.9	73.6±1.4	75.4±0.6
	5C	52.1±0.7	54.7±0.5	54.6±0.4	53.7±0.6	**55.8±0.4**	49.4±2.0	54.8±0.7	54.6±0.6	54.7±0.8	**55.8±0.4**
2002	SP	90.3±0.2	90.4±0.3	90.4±0.2	89.9±0.3	**90.5±0.2**	89.7±0.7	90.2±0.3	90.2±0.3	89.6±0.3	90.0±0.4
	DP	90.2±0.3	90.3±0.3	90.3±0.4	90.3±0.2	90.3±0.2	90.4±0.3	90.4±0.3	90.4±0.2	90.3±0.2	**90.5±0.3**
	3C	63.0±1.1	64.7±0.3	64.5±0.5	64.4±0.4	**65.1±0.7**	62.8±1.0	64.6±0.6	64.6±0.7	64.0±0.4	65.0±0.4
	4C	82.3±0.7	83.0±0.5	83.3±0.6	83.5±0.6	**84.5±0.5**	82.4±0.8	83.4±0.5	83.2±0.7	83.4±0.5	**84.5±0.5**
	5C	61.8±1.1	64.2±0.3	64.4±0.4	64.4±0.7	65.4±0.3	60.4±3.0	64.5±0.3	64.7±0.2	64.4±1.1	**65.5±0.2**
2003	SP	88.8±0.2	**89.1±0.4**	88.9±0.2	88.9±0.2	**89.1±0.3**	88.7±0.1	88.9±0.2	88.9±0.2	88.6±0.3	88.9±0.3
	DP	89.1±0.2	89.1±0.4	89.0±0.2	88.8±0.4	**89.2±0.5**	88.9±0.1	89.0±0.5	89.1±0.4	89.0±0.4	**89.2±0.5**
	3C	60.9±0.3	61.9±0.2	62.0±0.2	61.3±0.5	62.5±0.5	59.7±2.3	62.4±0.4	62.4±0.4	61.2±0.7	**62.7±0.4**
	4C	80.1±0.8	81.8±0.3	81.6±0.4	81.4±0.7	82.2±0.4	80.6±0.5	81.3±0.6	81.6±0.6	81.5±0.7	**82.5±0.3**
	5C	57.6±1.8	61.5±0.5	61.6±0.4	60.4±1.5	**62.4±0.3**	55.8±4.4	61.2±0.5	61.4±0.3	61.1±0.7	62.3±0.4
2004	SP	**87.5±0.4**	87.4±0.3	87.4±0.3	**87.5±0.5**	87.4±0.3	87.4±0.4	**87.5±0.4**	87.4±0.3	87.2±0.3	87.4±0.4
	DP	87.4±0.2	87.2±0.5	**87.5±0.3**	87.3±0.3	87.4±0.3	87.4±0.3	87.4±0.3	87.4±0.3	87.3±0.4	87.4±0.2
	3C	59.4±0.3	59.9±0.4	60.0±0.3	59.2±0.8	**60.3±0.4**	59.0±0.5	**60.3±0.6**	60.2±0.3	59.6±0.5	60.1±0.7
	4C	76.3±0.6	77.7±0.4	77.7±0.5	77.3±1.1	**78.3±0.3**	75.9±0.6	77.6±0.3	76.9±0.4	77.7±0.5	**78.3±0.3**
	5C	55.2±0.8	57.1±0.3	57.4±0.2	56.6±0.7	**57.9±0.4**	55.3±0.6	56.7±0.8	56.7±0.5	55.9±0.4	**57.9±0.3**
2005	SP	86.2±0.4	86.3±0.4	86.3±0.3	86.1±0.7	**86.4±0.3**	85.9±0.6	86.2±0.4	86.3±0.4	85.7±0.4	86.1±0.4
	DP	86.3±0.4	**86.5±0.3**	**86.5±0.4**	86.1±0.4	86.4±0.3	**86.5±0.4**	86.3±0.4	86.4±0.4	86.3±0.4	86.4±0.3
	3C	58.3±0.8	59.0±0.4	59.2±0.5	58.7±0.6	59.5±0.6	58.5±0.5	**59.7±0.4**	59.3±0.4	58.6±0.4	59.5±0.6
	4C	76.2±1.1	78.6±0.2	78.5±0.2	78.2±0.6	**79.6±0.2**	77.3±0.9	78.5±0.4	78.4±0.6	78.8±0.4	79.5±0.3
	5C	54.3±1.8	58.1±0.4	58.0±0.4	57.0±0.6	**59.0±0.4**	53.8±2.1	57.8±0.5	57.8±0.3	57.1±0.8	58.8±0.5
2006	SP	89.8±0.4	89.6±0.4	**89.9±0.4**	89.8±0.4	**89.9±0.5**	89.5±0.4	89.6±0.4	89.7±0.4	89.3±0.6	89.7±0.4
	DP	89.7±0.4	89.6±0.5	89.8±0.4	89.6±0.7	89.9±0.4	89.8±0.5	89.9±0.4	**90.1±0.5**	89.8±0.3	**90.1±0.4**
	3C	61.5±0.5	62.1±0.3	62.2±0.4	61.9±0.5	62.8±0.6	61.6±0.4	62.5±0.6	62.6±0.4	60.7±2.6	**63.0±0.5**
	4C	78.1±0.9	80.7±0.3	80.6±0.6	80.9±0.5	**81.6±0.2**	78.5±0.9	80.0±0.4	80.0±0.5	80.1±0.4	**81.6±0.3**
	5C	57.5±0.7	59.9±0.5	60.0±0.4	59.3±1.4	**60.9±0.4**	56.4±2.2	59.3±0.5	59.7±0.4	59.5±0.5	60.8±0.6
2007	SP	87.8±0.3	**88.0±0.3**	87.9±0.3	87.8±0.1	**88.0±0.2**	87.7±0.3	87.8±0.3	87.9±0.3	87.4±0.2	87.8±0.2
	DP	87.7±0.4	87.8±0.2	87.9±0.2	87.8±0.2	**88.0±0.2**	**88.0±0.1**	**88.0±0.2**	87.8±0.2	87.9±0.3	**88.0±0.3**
	3C	61.6±0.6	63.2±0.6	62.9±0.9	62.3±0.4	63.4±0.6	61.4±0.4	63.2±0.1	63.4±0.6	61.9±0.7	**63.7±0.5**
	4C	79.6±0.6	81.7±0.6	81.8±0.7	81.7±0.8	**83.1±0.2**	79.1±1.0	82.0±0.6	82.0±0.3	82.1±0.5	83.0±0.2
	5C	60.4±1.0	62.8±0.3	62.6±0.3	61.8±0.6	63.7±0.4	60.2±1.5	62.9±0.5	62.9±0.6	62.5±0.5	**64.1±0.4**
2008	SP	96.1±0.3	96.4±0.3	96.2±0.4	96.4±0.3	**96.5±0.3**	95.6±0.2	95.5±0.3	95.5±0.4	95.5±0.5	95.7±0.2
	DP	96.4±0.3	96.2±0.4	96.3±0.3	96.5±0.3	**96.6±0.3**	96.2±0.3	96.5±0.3	96.4±0.3	96.3±0.4	**96.6±0.2**
	3C	75.4±1.0	76.8±0.3	77.0±0.2	75.4±1.5	**77.6±0.1**	73.2±1.0	75.9±0.8	76.0±0.2	74.5±0.9	76.7±0.2
	4C	93.9±0.3	94.8±0.4	95.1±0.2	95.0±0.5	**95.7±0.2**	92.7±0.6	94.0±0.4	94.3±0.2	94.5±0.4	95.4±0.5
	5C	74.7±4.0	78.8±0.3	78.7±0.4	77.8±0.8	**79.8±0.3**	74.8±0.9	77.4±0.8	76.6±2.2	76.7±1.2	78.9±0.7
2009	SP	94.9±0.2	95.1±0.1	95.0±0.2	95.0±0.2	**95.2±0.2**	94.7±0.1	94.7±0.1	94.7±0.3	94.5±0.3	94.5±0.2
	DP	94.9±0.1	95.1±0.1	95.0±0.2	95.0±0.2	**95.3±0.1**	95.1±0.1	95.1±0.1	95.1±0.2	95.1±0.2	95.2±0.2
	3C	68.1±0.6	70.4±0.1	**70.8±0.3**	69.9±0.6	**70.8±0.4**	69.2±0.5	70.3±0.4	70.5±0.4	70.0±0.4	70.5±0.2
	4C	88.9±1.2	90.5±0.4	90.8±0.3	91.2±0.4	**91.6±0.3**	89.9±0.2	90.8±0.4	90.6±0.4	90.9±0.3	91.4±0.3
	5C	69.7±0.8	72.5±0.3	72.4±0.3	72.2±0.3	**73.5±0.4**	69.9±0.6	72.1±0.2	72.4±0.3	71.9±0.4	73.2±0.2
2010	SP	92.3±0.4	92.4±0.4	92.4±0.3	92.3±0.3	**92.5±0.3**	92.0±0.5	92.2±0.4	92.1±0.4	91.9±0.3	92.2±0.2
	DP	92.1±0.5	92.4±0.4	92.4±0.3	92.3±0.3	92.4±0.4	92.4±0.3	**92.5±0.3**	**92.5±0.3**	92.3±0.4	**92.5±0.4**
	3C	63.7±1.8	66.5±0.7	66.4±0.4	66.0±0.4	**67.2±0.3**	64.1±0.8	66.7±0.7	66.6±0.5	65.5±0.4	67.1±0.5
	4C	87.5±1.0	88.3±0.5	88.6±0.4	88.4±0.6	**89.4±0.4**	86.9±1.1	88.1±0.3	88.1±0.4	88.7±0.4	89.3±0.3
	5C	67.1±0.7	68.8±0.5	68.9±0.1	67.8±1.0	69.8±0.2	66.1±1.4	68.6±0.6	68.6±0.3	67.7±0.8	**69.9±0.3**

Table C.24: Link prediction test performance (accuracy in percentage) comparison for variants of MSGNN for individual years 2011-2020 of the *FiLL-OPCL* data set. Each variant is denoted by a q value and a 2-tuple: (whether to include signed features, whether to include weighted features), where "T" and "F" stand for "True" and "False", respectively. "T" for weighted features means simply summing up entries in the adjacency matrix while "T'" means summing the absolute values of the entries. The best is marked in **bold red** and the second best is marked in underline blue .

q value		0					$q_0 := 1/[2\max_{i,j}(\mathbf{A}_{i,j} - \mathbf{A}_{j,i})]$				
Year	Link Task	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)	(F, F)	(F, T)	(F, T')	(T, F)	(T, T)
2011	SP	96.0±0.4	96.1±0.3	96.1±0.3	**96.2±0.2**	**96.2±0.3**	95.4±0.4	95.5±0.5	95.5±0.3	95.7±0.3	95.9±0.2
	DP	95.5±0.9	96.0±0.4	96.0±0.2	**96.3±0.4**	**96.3±0.2**	96.0±0.2	96.1±0.3	96.2±0.2	96.2±0.3	**96.3±0.3**
	3C	77.2±0.4	78.3±0.6	78.7±0.1	77.9±0.4	**79.0±0.2**	76.8±0.8	78.5±0.6	78.7±0.4	76.7±1.8	78.6±0.4
	4C	94.3±0.4	94.9±0.3	95.0±0.2	95.0±0.3	95.4±0.2	93.9±0.6	94.8±0.2	94.7±0.2	95.1±0.2	**95.5±0.3**
	5C	77.5±1.3	80.2±0.6	80.2±0.4	79.3±0.5	**80.9±0.4**	77.6±1.7	79.7±0.6	79.8±0.5	79.8±0.4	80.6±0.3
2012	SP	91.0±0.5	**91.1±0.3**	**91.1±0.2**	90.9±0.4	**91.1±0.2**	90.6±0.4	90.7±0.6	90.8±0.4	90.6±0.2	90.8±0.5
	DP	90.9±0.4	91.1±0.3	91.0±0.2	91.0±0.3	**91.2±0.3**	91.0±0.3	**91.2±0.3**	91.1±0.4	91.0±0.2	**91.2±0.2**
	3C	63.1±0.8	64.1±0.4	63.6±0.4	63.4±0.4	64.2±0.4	63.4±0.3	64.0±0.7	64.2±0.6	63.3±1.2	**64.4±0.5**
	4C	78.6±0.9	80.6±0.2	80.6±0.3	80.9±0.6	**81.7±0.4**	78.0±1.5	80.6±0.4	80.4±0.3	80.5±0.5	81.5±0.4
	5C	57.2±1.3	60.6±0.0	60.7±0.5	60.0±0.7	**61.6±0.5**	55.8±3.1	60.5±0.5	60.3±0.3	60.2±0.5	61.5±0.3
2013	SP	89.1±0.2	89.3±0.2	89.3±0.3	89.3±0.3	**89.5±0.3**	88.9±0.3	88.9±0.3	89.0±0.2	88.9±0.2	89.4±0.2
	DP	89.0±0.3	89.3±0.2	89.2±0.3	89.1±0.3	**89.4±0.3**	89.1±0.2	89.3±0.2	89.2±0.3	89.2±0.2	89.3±0.2
	3C	63.3±0.2	63.8±0.4	63.9±0.2	63.2±0.4	**64.3±0.4**	62.8±0.4	63.9±0.3	64.1±0.3	63.0±0.5	**64.3±0.3**
	4C	79.8±1.6	82.0±0.4	81.8±0.4	82.6±0.6	**83.3±0.3**	80.8±1.4	82.3±0.3	82.0±0.5	81.7±1.0	82.8±0.8
	5C	61.5±0.5	63.1±0.4	63.0±0.5	61.9±1.6	**64.2±0.3**	60.7±0.8	62.5±0.7	63.1±0.6	62.4±0.8	63.9±0.4
2014	SP	87.7±0.3	**87.8±0.4**	87.6±0.3	87.6±0.3	87.7±0.4	87.5±0.4	87.3±0.7	87.5±0.3	87.3±0.5	87.5±0.3
	DP	87.6±0.4	87.6±0.3	87.7±0.3	87.7±0.5	87.7±0.5	87.6±0.3	87.6±0.3	87.6±0.4	87.7±0.5	**87.9±0.4**
	3C	60.5±0.6	61.2±0.5	61.2±0.7	60.9±0.3	61.8±0.7	60.6±0.5	61.7±0.4	61.6±0.2	61.1±0.3	**61.9±0.2**
	4C	78.3±1.6	80.1±0.2	80.4±0.3	80.3±0.4	**81.3±0.3**	78.5±0.3	80.6±0.2	80.4±0.3	80.9±0.4	81.2±0.2
	5C	58.4±1.1	61.0±0.4	61.2±0.4	60.5±0.5	**61.9±0.3**	58.2±1.0	60.5±0.3	60.9±0.3	59.0±1.6	61.8±0.3
2015	SP	89.5±0.4	89.6±0.3	89.7±0.3	89.5±0.4	**89.8±0.3**	89.1±0.2	89.1±0.4	89.2±0.5	89.0±0.5	89.6±0.4
	DP	89.5±0.2	89.7±0.3	89.5±0.3	89.7±0.3	**89.9±0.3**	89.6±0.2	89.6±0.3	89.7±0.3	89.6±0.2	89.8±0.3
	3C	62.2±0.6	63.5±0.6	63.7±0.5	62.7±0.8	**64.1±0.4**	62.0±1.0	63.8±0.3	63.7±0.5	62.8±0.2	**64.1±0.2**
	4C	82.1±0.5	83.6±0.4	83.5±0.2	83.7±0.6	84.7±0.3	82.8±0.4	83.6±0.4	83.6±0.3	83.8±0.5	**84.8±0.2**
	5C	61.9±0.5	63.8±0.5	63.9±0.6	63.5±0.7	**64.9±0.4**	61.5±0.5	63.7±0.5	63.7±0.8	63.4±1.1	64.8±0.5
2016	SP	88.9±0.4	88.9±0.3	88.9±0.3	88.8±0.4	**89.0±0.2**	88.7±0.2	88.8±0.3	88.5±0.3	88.7±0.4	88.7±0.3
	DP	**88.9±0.3**	88.7±0.3	88.6±0.4	88.8±0.4	**88.9±0.2**	88.6±0.6	**88.9±0.2**	88.8±0.3	88.7±0.3	**88.9±0.3**
	3C	61.4±0.6	62.4±0.6	61.9±0.3	61.6±0.5	**62.5±0.2**	61.5±0.6	61.8±0.5	62.0±0.4	61.4±0.5	62.2±0.5
	4C	72.6±1.7	75.2±0.4	74.4±0.2	76.2±0.6	**76.8±0.2**	73.6±1.0	75.0±0.7	74.9±0.1	75.6±0.6	76.7±0.5
	5C	54.7±1.0	56.8±0.7	56.6±0.5	56.4±1.0	57.9±0.3	51.4±1.6	56.9±0.8	57.1±0.7	56.2±0.6	**58.1±0.3**
2017	SP	89.2±0.2	**89.4±0.2**	89.2±0.2	89.2±0.2	89.3±0.3	89.1±0.2	89.3±0.2	89.3±0.3	88.9±0.3	**89.4±0.3**
	DP	89.3±0.2	89.2±0.3	89.2±0.2	89.4±0.1	89.3±0.3	89.4±0.1	89.4±0.2	89.3±0.3	89.3±0.2	**89.5±0.2**
	3C	60.4±0.6	61.0±0.3	61.1±0.2	60.7±0.3	61.3±0.3	60.9±0.5	**61.4±0.3**	61.3±0.2	60.7±0.3	61.3±0.2
	4C	65.8±0.4	68.4±0.6	68.3±0.3	68.9±0.8	**69.5±0.5**	64.9±2.9	68.4±0.8	68.6±0.5	68.1±0.7	69.4±0.4
	5C	49.2±0.5	51.0±0.3	51.1±0.6	49.9±1.3	**52.0±0.4**	47.9±1.2	51.1±0.6	51.4±0.4	49.7±0.8	51.8±0.3
2018	SP	88.0±0.5	88.0±0.4	88.0±0.4	88.0±0.3	**88.2±0.4**	87.3±0.5	87.8±0.5	87.8±0.6	87.4±0.6	87.7±0.6
	DP	87.7±0.7	88.0±0.6	88.0±0.4	**88.1±0.4**	**88.1±0.4**	88.0±0.6	**88.1±0.5**	88.0±0.4	87.8±0.6	**88.1±0.4**
	3C	61.1±0.4	63.1±0.5	63.1±0.5	62.5±0.6	63.7±0.3	61.3±1.3	63.8±0.4	63.8±0.5	62.0±0.7	**64.0±0.6**
	4C	80.6±0.5	82.1±0.3	82.0±0.4	82.1±0.6	**83.0±0.4**	79.6±0.6	81.1±0.5	81.3±0.3	81.6±1.2	**83.0±0.7**
	5C	61.1±0.9	62.8±0.5	62.7±0.5	62.2±0.5	**63.8±0.5**	58.3±5.1	62.3±0.4	62.4±0.5	61.9±0.9	63.7±0.5
2019	SP	89.2±0.2	89.1±0.3	89.2±0.5	89.1±0.4	**89.3±0.4**	88.6±1.1	89.1±0.2	89.0±0.2	88.6±0.3	89.1±0.5
	DP	89.0±0.4	89.2±0.2	89.3±0.2	89.1±0.2	**89.4±0.2**	89.3±0.3	89.3±0.2	89.3±0.4	89.2±0.4	89.3±0.2
	3C	61.3±0.3	62.0±0.4	61.9±0.2	61.6±0.3	62.2±0.2	60.4±0.8	62.3±0.5	62.2±0.5	62.1±0.5	**62.4±0.5**
	4C	70.8±1.0	72.4±0.4	72.8±0.3	73.2±0.5	**74.5±0.2**	71.6±0.8	72.9±0.3	72.9±0.2	73.1±0.6	74.3±0.4
	5C	51.4±2.0	55.0±0.3	55.2±0.2	53.7±0.9	**56.0±0.3**	52.4±1.3	55.1±0.3	55.2±0.2	53.3±2.3	**56.0±0.2**
2020	SP	92.0±0.1	91.8±0.4	91.8±0.2	92.2±0.2	**92.3±0.1**	91.5±0.2	91.4±0.2	91.3±0.2	91.6±0.2	91.5±0.1
	DP	91.8±0.2	91.7±0.3	92.0±0.3	**92.2±0.1**	**92.2±0.1**	91.7±0.3	92.0±0.2	91.9±0.1	92.0±0.2	92.1±0.1
	3C	66.5±0.7	69.0±0.3	68.6±0.5	68.2±1.0	**69.2±0.3**	66.0±1.2	**69.2±0.3**	69.0±0.4	67.3±0.7	**69.2±0.4**
	4C	83.0±0.9	84.0±0.4	84.0±0.3	84.8±0.3	**85.4±0.2**	81.4±1.4	83.1±0.5	83.1±0.7	84.1±1.0	84.8±0.4
	5C	64.2±1.7	66.5±0.5	66.6±0.7	66.3±0.6	**67.8±0.2**	62.5±0.5	65.1±1.0	65.6±1.4	65.3±1.2	67.3±0.6

Table C.25: Link prediction test performance (accuracy in percentage) comparison for MSGNN with different q values for individual years 2000-2010 of the *FiLL-pvCLCL* data set. The best is marked in **bold red** and the second best is marked in underline blue .

Year	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
2000	SP	89.0±0.4	**89.1±0.3**	88.9±0.5	88.9±0.4	88.7±0.6	88.8±0.5
	DP	**89.1±0.4**	**89.1±0.4**	**89.1±0.3**	**89.1±0.4**	**89.1±0.4**	**89.1±0.5**
	3C	61.3±0.4	61.3±0.4	61.3±0.5	61.3±0.5	**61.5±0.5**	**61.5±0.7**
	4C	**72.3±0.5**	72.1±0.5	72.1±0.7	72.1±0.5	72.2±0.6	71.9±0.5
	5C	**54.0±0.4**	53.9±0.4	**54.0±0.5**	53.9±0.6	53.8±0.5	53.9±0.4
2001	SP	**90.7±0.2**	90.5±0.3	90.4±0.2	90.6±0.3	90.4±0.3	**90.7±0.2**
	DP	**90.7±0.2**	**90.7±0.1**	**90.7±0.2**	**90.7±0.2**	**90.7±0.2**	**90.7±0.1**
	3C	62.6±0.2	62.6±0.2	62.6±0.3	62.7±0.2	62.6±0.2	**63.1±0.5**
	4C	75.8±0.3	75.8±0.4	75.7±0.3	**76.1±0.3**	75.9±0.4	75.7±0.3
	5C	56.7±0.4	**56.8±0.2**	56.7±0.3	**56.8±0.3**	**56.8±0.3**	**56.8±0.2**
2002	SP	**91.4±0.2**	91.0±0.3	91.1±0.2	90.9±0.4	91.2±0.3	91.2±0.3
	DP	**91.5±0.1**	**91.5±0.1**	91.4±0.3	**91.5±0.2**	91.4±0.1	**91.5±0.2**
	3C	**66.2±0.4**	66.0±0.2	66.0±0.4	66.1±0.2	65.9±0.4	**66.2±0.2**
	4C	**85.7±0.5**	**85.7±0.5**	85.5±0.3	85.5±0.3	85.6±0.4	**85.7±0.4**
	5C	**66.8±0.4**	**66.8±0.4**	66.7±0.5	66.7±0.3	**66.8±0.3**	66.7±0.4
2003	SP	**89.5±0.4**	89.0±0.4	89.4±0.3	89.2±0.4	89.2±0.3	89.3±0.4
	DP	**89.6±0.4**	89.5±0.4	89.5±0.4	89.5±0.4	**89.6±0.4**	**89.6±0.4**
	3C	63.2±0.5	62.9±0.5	**63.3±0.3**	63.2±0.4	63.2±0.5	63.1±0.5
	4C	82.6±0.4	82.8±0.3	**82.9±0.2**	82.8±0.2	**82.9±0.2**	82.7±0.4
	5C	62.6±0.2	62.7±0.3	**62.8±0.3**	**62.8±0.4**	62.4±0.4	62.7±0.4
2004	SP	88.7±0.3	88.2±0.8	88.7±0.3	88.7±0.3	88.6±0.2	**88.8±0.3**
	DP	**88.8±0.3**	88.7±0.3	88.7±0.4	**88.8±0.4**	88.7±0.2	**88.8±0.3**
	3C	61.6±0.5	61.7±0.5	**61.8±0.5**	61.7±0.2	**61.8±0.3**	61.6±0.4
	4C	78.8±0.6	78.7±0.4	**78.9±0.5**	78.6±0.4	78.7±0.5	78.7±0.4
	5C	58.7±0.3	58.7±0.4	**58.8±0.5**	**58.8±0.5**	**58.8±0.3**	58.7±0.6
2005	SP	**87.8±0.4**	87.4±0.6	87.4±0.4	87.7±0.5	87.7±0.3	87.7±0.4
	DP	**87.9±0.4**	87.7±0.5	87.8±0.5	87.8±0.4	87.8±0.4	87.8±0.4
	3C	**61.2±0.2**	**61.2±0.2**	61.0±0.4	60.9±0.3	61.1±0.2	60.9±0.6
	4C	**80.8±0.2**	80.7±0.4	**80.8±0.2**	**80.8±0.2**	80.6±0.2	80.5±0.2
	5C	**61.3±0.4**	**61.3±0.3**	61.2±0.3	**61.3±0.2**	61.1±0.2	61.0±0.2
2006	SP	**91.0±0.2**	90.7±0.4	90.7±0.2	**91.0±0.3**	90.6±0.1	90.6±0.1
	DP	**91.1±0.2**	91.0±0.2	**91.1±0.1**	91.0±0.1	90.9±0.2	**91.1±0.1**
	3C	64.0±0.3	**64.3±0.4**	64.2±0.4	64.0±0.4	64.0±0.2	64.1±0.3
	4C	82.9±0.2	82.8±0.2	83.0±0.3	82.9±0.2	82.9±0.3	**83.1±0.4**
	5C	62.6±0.3	62.6±0.2	62.6±0.4	62.6±0.2	**63.0±0.3**	62.8±0.3
2007	SP	**90.4±0.5**	90.3±0.4	90.0±0.3	90.3±0.3	90.1±0.3	90.0±0.4
	DP	90.4±0.3	90.5±0.4	90.4±0.4	90.4±0.3	90.5±0.4	**90.6±0.4**
	3C	69.0±0.4	69.0±0.7	**69.2±0.3**	69.1±0.3	69.0±0.2	69.0±0.3
	4C	88.1±0.2	88.3±0.4	88.2±0.4	88.1±0.3	**88.4±0.2**	**88.4±0.2**
	5C	69.9±0.5	69.9±0.6	**70.0±0.5**	69.7±0.4	69.7±0.5	69.7±0.2
2008	SP	**96.4±0.2**	95.8±0.2	95.9±0.1	95.7±0.3	95.5±0.4	95.7±0.3
	DP	96.4±0.1	**96.5±0.2**	**96.5±0.1**	**96.5±0.1**	96.3±0.4	96.3±0.2
	3C	**79.3±0.7**	79.0±0.2	79.2±0.3	79.1±0.1	78.5±0.3	78.9±0.2
	4C	96.1±0.2	**96.5±0.2**	96.3±0.3	96.2±0.2	96.1±0.5	96.2±0.3
	5C	**82.4±0.4**	82.2±0.6	82.2±0.7	82.2±0.6	82.1±0.4	82.0±0.4
2009	SP	**97.8±0.2**	97.2±0.1	97.3±0.2	97.3±0.2	97.1±0.2	97.1±0.3
	DP	**97.8±0.1**	97.7±0.1	**97.8±0.2**	**97.8±0.2**	**97.8±0.2**	97.6±0.1
	3C	78.4±0.4	**78.6±0.4**	78.5±0.2	78.5±0.5	78.4±0.4	**78.6±0.3**
	4C	95.2±0.3	**95.3±0.3**	95.1±0.4	95.1±0.3	94.8±0.3	95.0±0.2
	5C	**79.9±0.2**	**79.9±0.3**	79.8±0.2	**79.9±0.4**	79.6±0.6	79.8±0.2
2010	SP	**92.8±0.3**	92.0±0.2	92.5±0.2	92.4±0.3	92.4±0.3	92.3±0.3
	DP	92.6±0.3	92.7±0.3	92.7±0.3	92.6±0.4	**92.8±0.4**	92.7±0.4
	3C	68.3±0.2	68.4±0.2	68.4±0.4	68.4±0.3	**68.6±0.3**	68.5±0.4
	4C	90.9±0.4	90.9±0.3	91.0±0.4	90.9±0.3	91.0±0.4	**91.1±0.4**
	5C	**72.5±0.4**	**72.5±0.3**	72.4±0.3	**72.5±0.3**	72.4±0.3	72.4±0.1

Table C.26: Link prediction test performance (accuracy in percentage) comparison for MSGNN with different q values for individual years 2011-2020 of the *FiLL-pvCLCL* data set. The best is marked in **bold red** and the second best is marked in underline blue .

Year	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
2011	SP	**98.7±0.2**	98.4±0.3	98.3±0.2	98.5±0.2	98.2±0.5	98.3±0.3
	DP	**98.7±0.2**	**98.7±0.1**	**98.7±0.1**	**98.7±0.1**	**98.7±0.1**	**98.7±0.1**
	3C	86.5±0.2	**86.6±0.1**	86.1±0.5	86.4±0.2	86.2±0.3	86.2±0.3
	4C	98.3±0.3	**98.4±0.1**	**98.4±0.2**	98.3±0.4	98.3±0.2	98.3±0.2
	5C	87.4±0.3	87.3±0.3	**87.5±0.3**	87.3±0.4	87.2±0.6	87.2±0.4
2012	SP	**92.7±0.3**	92.1±0.8	92.4±0.3	92.4±0.4	92.5±0.3	92.4±0.3
	DP	92.6±0.2	92.6±0.2	**92.7±0.3**	**92.7±0.4**	**92.7±0.3**	92.6±0.3
	3C	67.1±0.2	**67.5±0.4**	67.1±0.2	67.2±0.4	67.1±0.3	67.4±0.3
	4C	87.0±0.5	86.9±0.4	**87.1±0.5**	87.0±0.5	87.0±0.5	86.9±0.4
	5C	66.8±0.1	66.9±0.1	66.8±0.3	66.8±0.3	66.9±0.1	**67.0±0.2**
2013	SP	**90.5±0.3**	90.0±0.2	90.2±0.4	90.1±0.2	90.3±0.3	90.3±0.1
	DP	90.3±0.3	**90.5±0.2**	90.4±0.1	90.3±0.3	90.4±0.2	90.4±0.3
	3C	**66.9±0.2**	66.3±0.4	66.3±0.4	66.4±0.4	66.3±0.4	66.1±0.3
	4C	85.9±0.3	**86.1±0.2**	**86.1±0.3**	**86.1±0.2**	86.0±0.3	**86.1±0.3**
	5C	67.0±0.3	66.9±0.2	67.0±0.4	**67.1±0.2**	67.0±0.1	66.8±0.3
2014	SP	**87.3±0.2**	86.8±0.2	87.2±0.2	87.0±0.3	86.9±0.3	87.1±0.2
	DP	**87.3±0.2**	87.2±0.2	87.2±0.2	87.2±0.3	87.2±0.2	87.2±0.3
	3C	**60.8±0.3**	60.5±0.2	**60.8±0.1**	**60.8±0.2**	**60.8±0.2**	60.6±0.2
	4C	80.2±0.3	80.1±0.2	80.2±0.3	**80.3±0.3**	80.1±0.4	80.2±0.2
	5C	60.3±0.4	60.4±0.2	**60.5±0.2**	60.4±0.4	60.3±0.3	60.3±0.4
2015	SP	**89.1±0.4**	87.7±1.0	88.9±0.1	88.8±0.4	88.8±0.1	88.8±0.2
	DP	89.2±0.4	89.2±0.4	89.2±0.3	89.2±0.4	**89.4±0.4**	89.3±0.3
	3C	63.5±0.5	63.9±0.3	**64.1±0.2**	63.7±0.3	63.7±0.4	63.8±0.4
	4C	84.6±0.5	84.5±0.5	84.6±0.3	84.5±0.3	**84.7±0.4**	84.4±0.5
	5C	65.6±0.4	**65.9±0.3**	65.8±0.2	65.5±0.4	65.4±0.3	65.7±0.4
2016	SP	**90.2±0.5**	89.7±0.4	89.6±0.4	89.8±0.5	89.7±0.5	89.7±0.5
	DP	90.1±0.5	**90.2±0.4**	**90.2±0.4**	**90.2±0.4**	90.0±0.4	90.0±0.5
	3C	64.0±0.3	64.0±0.3	63.9±0.2	63.9±0.3	**64.1±0.3**	63.9±0.2
	4C	82.0±0.5	81.8±0.5	81.7±0.6	**82.2±0.6**	81.8±0.6	82.1±0.5
	5C	**62.2±0.3**	62.1±0.3	62.0±0.4	62.1±0.5	61.9±0.4	62.0±0.4
2017	SP	**90.2±0.3**	89.8±0.4	89.8±0.9	89.9±0.4	90.0±0.3	90.0±0.3
	DP	90.1±0.2	**90.2±0.3**	**90.2±0.3**	90.0±0.4	90.1±0.3	**90.2±0.3**
	3C	62.0±0.1	62.4±0.4	62.4±0.3	**62.5±0.4**	62.4±0.2	62.3±0.4
	4C	**70.1±0.5**	70.0±0.5	**70.1±0.5**	70.0±0.2	69.9±0.5	69.6±0.4
	5C	**53.4±0.2**	53.2±0.2	**53.4±0.1**	53.3±0.2	53.3±0.3	53.2±0.1
2018	SP	**87.0±0.4**	86.7±0.4	85.8±1.8	86.8±0.6	86.7±0.3	86.6±0.8
	DP	86.9±0.4	**87.0±0.5**	86.8±0.4	**87.0±0.5**	**87.0±0.5**	**87.0±0.4**
	3C	61.5±0.4	61.5±0.6	**61.8±0.3**	**61.8±0.5**	61.7±0.3	61.4±0.3
	4C	79.7±0.4	**79.8±0.4**	79.5±0.4	79.7±0.4	79.7±0.5	79.5±0.5
	5C	**60.9±0.4**	60.8±0.5	60.7±0.5	60.7±0.7	60.6±0.5	60.5±0.3
2019	SP	**90.8±0.3**	90.4±0.2	90.3±0.2	90.6±0.3	90.4±0.3	90.6±0.4
	DP	**90.9±0.3**	**90.9±0.2**	90.8±0.3	90.8±0.2	90.8±0.2	**90.9±0.3**
	3C	**64.6±0.2**	64.5±0.3	64.3±0.3	64.3±0.2	64.4±0.2	64.3±0.2
	4C	**77.9±0.3**	**77.9±0.3**	**77.9±0.3**	77.8±0.5	77.6±0.7	77.5±0.4
	5C	**59.0±0.2**	58.8±0.2	**59.0±0.4**	58.9±0.4	**59.0±0.3**	58.7±0.2
2020	SP	**97.3±0.2**	96.8±0.2	96.7±0.2	96.5±0.2	96.6±0.1	96.6±0.2
	DP	97.3±0.2	**97.4±0.2**	97.3±0.1	97.3±0.2	97.3±0.1	97.2±0.1
	3C	82.8±1.3	82.7±1.0	**83.2±0.5**	82.7±0.8	82.8±0.5	82.9±0.5
	4C	96.4±0.2	**96.6±0.2**	96.5±0.1	96.4±0.1	96.4±0.3	96.5±0.1
	5C	**83.2±0.4**	83.0±0.3	82.9±0.5	83.1±0.2	83.0±0.4	82.8±0.4

Table C.27: Link prediction test performance (accuracy in percentage) comparison for MSGNN with different q values for individual years 2000-2010 of the *FiLL-OPCL* data set. The best is marked in **bold red** and the second best is marked in underline blue .

Year	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
2000	SP	**87.9±0.5**	87.4±0.6	87.6±0.5	87.5±0.5	87.6±0.4	87.6±0.5
	DP	87.9±0.5	87.9±0.4	87.7±0.7	**88.0±0.5**	87.9±0.6	87.9±0.6
	3C	60.7±0.3	**60.8±0.4**	60.7±0.3	60.2±0.4	60.7±0.5	60.7±0.4
	4C	71.3±0.3	71.3±0.4	71.3±0.4	**71.5±0.3**	71.4±0.4	71.3±0.3
	5C	**53.5±0.4**	53.2±0.5	53.4±0.4	53.4±0.4	53.4±0.5	53.4±0.5
2001	SP	**90.0±0.5**	89.9±0.3	89.9±0.4	89.6±0.5	89.8±0.5	89.9±0.4
	DP	90.0±0.3	90.1±0.4	**90.2±0.2**	90.1±0.4	90.1±0.3	**90.2±0.3**
	3C	61.9±0.3	62.1±0.7	61.9±0.5	**62.2±0.4**	62.0±0.5	**62.2±0.4**
	4C	75.5±0.3	75.2±0.3	**75.6±0.5**	75.5±0.3	**75.6±0.6**	75.4±0.6
	5C	**55.8±0.4**	**55.8±0.4**	55.6±0.4	55.7±0.3	**55.8±0.4**	**55.8±0.4**
2002	SP	**90.5±0.2**	90.1±0.1	90.2±0.3	90.0±0.2	90.0±0.3	90.0±0.4
	DP	90.3±0.2	**90.5±0.3**	90.4±0.3	90.4±0.2	**90.5±0.3**	**90.5±0.3**
	3C	65.1±0.7	65.1±0.5	65.3±0.3	65.4±0.5	**65.5±0.3**	65.0±0.4
	4C	**84.5±0.5**	84.3±0.4	84.3±0.5	**84.5±0.4**	84.4±0.3	**84.5±0.5**
	5C	65.4±0.3	**65.6±0.5**	65.4±0.7	65.5±0.5	65.4±0.4	65.5±0.2
2003	SP	**89.1±0.3**	89.0±0.3	**89.1±0.3**	**89.1±0.3**	88.8±0.3	88.9±0.3
	DP	89.2±0.5	89.2±0.3	89.0±0.2	**89.3±0.4**	89.1±0.4	89.2±0.5
	3C	62.5±0.5	**62.7±0.3**	62.5±0.5	62.5±0.5	62.5±0.5	**62.7±0.4**
	4C	82.2±0.4	82.3±0.3	82.4±0.5	82.3±0.4	82.3±0.4	**82.5±0.3**
	5C	62.4±0.3	62.3±0.4	62.4±0.4	62.3±0.4	**62.6±0.2**	62.3±0.4
2004	SP	87.4±0.3	87.0±0.4	87.3±0.2	87.4±0.4	**87.5±0.3**	87.4±0.4
	DP	87.4±0.3	**87.5±0.3**	87.3±0.3	**87.5±0.4**	**87.5±0.3**	87.4±0.2
	3C	60.3±0.4	60.6±0.5	60.5±0.5	60.0±0.4	**60.7±0.3**	60.1±0.7
	4C	78.3±0.3	**78.6±0.2**	78.3±0.3	78.4±0.4	78.2±0.2	78.3±0.3
	5C	57.9±0.4	57.9±0.4	**58.1±0.3**	57.8±0.1	57.9±0.5	57.9±0.3
2005	SP	**86.4±0.3**	86.3±0.3	86.0±0.4	86.2±0.3	86.2±0.5	86.1±0.4
	DP	86.4±0.3	**86.5±0.3**	86.4±0.3	**86.5±0.3**	86.4±0.3	86.4±0.3
	3C	59.5±0.6	**59.6±0.3**	**59.6±0.6**	**59.6±0.3**	59.5±0.4	59.5±0.6
	4C	79.6±0.2	79.5±0.4	79.5±0.3	79.6±0.3	**79.7±0.4**	79.5±0.3
	5C	59.0±0.4	59.0±0.2	59.0±0.4	**59.1±0.3**	59.0±0.4	58.8±0.5
2006	SP	**89.9±0.5**	89.5±0.4	89.2±1.2	89.6±0.6	89.7±0.5	89.7±0.4
	DP	89.9±0.4	89.9±0.4	**90.1±0.4**	90.0±0.4	**90.1±0.4**	**90.1±0.4**
	3C	62.8±0.6	62.7±0.4	62.7±0.6	62.7±0.4	**63.0±0.5**	**63.0±0.5**
	4C	**81.6±0.2**	81.4±0.4	81.5±0.3	81.2±0.4	81.4±0.3	**81.6±0.3**
	5C	60.9±0.4	60.9±0.6	60.9±0.5	**61.0±0.4**	60.9±0.4	60.8±0.6
2007	SP	**88.0±0.2**	87.7±0.2	87.8±0.3	87.9±0.2	87.8±0.2	87.8±0.2
	DP	88.0±0.2	87.9±0.3	88.0±0.3	88.0±0.3	**88.1±0.3**	88.0±0.3
	3C	63.4±0.6	63.8±0.4	63.7±0.6	**63.9±0.5**	63.8±0.4	63.7±0.5
	4C	83.1±0.2	**83.4±0.4**	83.1±0.3	83.3±0.6	83.1±0.4	83.0±0.2
	5C	63.7±0.4	64.0±0.2	64.0±0.4	64.0±0.4	63.8±0.4	**64.1±0.4**
2008	SP	**96.5±0.3**	96.0±0.4	95.9±0.3	95.8±0.4	95.7±0.4	95.7±0.2
	DP	**96.6±0.3**	96.5±0.2	96.5±0.2	96.4±0.2	96.4±0.3	**96.6±0.2**
	3C	**77.6±0.1**	76.6±0.6	76.7±0.4	77.0±0.3	76.6±0.5	76.7±0.2
	4C	**95.7±0.2**	95.6±0.2	**95.7±0.2**	95.6±0.2	**95.7±0.3**	95.4±0.5
	5C	**79.8±0.3**	79.7±0.4	**79.8±0.3**	79.5±0.2	79.4±0.1	78.9±0.7
2009	SP	**95.2±0.2**	94.8±0.2	94.6±0.2	94.7±0.2	94.5±0.4	94.5±0.2
	DP	**95.3±0.1**	95.1±0.1	95.2±0.1	95.2±0.2	95.1±0.2	95.2±0.2
	3C	70.8±0.4	**70.9±0.4**	70.5±0.3	70.6±0.4	70.7±0.2	70.5±0.2
	4C	91.6±0.3	91.5±0.3	91.6±0.2	91.6±0.1	**91.7±0.3**	91.4±0.3
	5C	**73.5±0.4**	73.4±0.3	**73.5±0.2**	**73.5±0.5**	73.3±0.4	73.2±0.2
2010	SP	**92.5±0.3**	92.2±0.4	92.1±0.3	91.3±1.0	92.2±0.3	92.2±0.2
	DP	92.4±0.4	**92.5±0.3**	**92.5±0.3**	**92.5±0.3**	92.4±0.4	**92.5±0.4**
	3C	**67.2±0.3**	67.0±0.4	67.0±0.5	67.0±0.4	67.0±0.3	67.1±0.5
	4C	**89.4±0.4**	89.2±0.4	89.0±0.2	88.9±0.4	89.2±0.4	89.3±0.3
	5C	69.8±0.2	69.8±0.3	**70.0±0.4**	69.9±0.3	69.7±0.4	69.9±0.3

Table C.28: Link prediction test performance (accuracy in percentage) comparison for MSGNN with different q values for individual years 2011-2020 of the *FiLL-OPCL* data set. The best is marked in **bold red** and the second best is marked in underline blue .

Year	Link Task	$q = 0$	$q = 0.2q_0$	$q = 0.4q_0$	$q = 0.6q_0$	$q = 0.8q_0$	$q = q_0$
2011	SP	**96.2±0.3**	96.0±0.2	96.0±0.2	95.9±0.3	96.0±0.2	95.9±0.2
	DP	96.3±0.2	**96.4±0.3**	96.3±0.2	96.3±0.3	96.3±0.2	96.3±0.3
	3C	79.0±0.2	78.8±0.4	**79.1±0.3**	79.0±0.3	79.0±0.2	78.6±0.4
	4C	95.4±0.2	95.4±0.3	**95.5±0.2**	**95.5±0.3**	95.4±0.2	**95.5±0.3**
	5C	80.9±0.4	**81.0±0.6**	80.8±0.3	80.8±0.3	80.7±0.6	80.6±0.3
2012	SP	**91.1±0.2**	91.0±0.4	90.8±0.3	90.8±0.4	90.8±0.5	90.8±0.5
	DP	**91.2±0.3**	**91.2±0.3**	**91.2±0.3**	**91.2±0.2**	**91.2±0.4**	**91.2±0.2**
	3C	64.2±0.4	**64.6±0.5**	**64.6±0.5**	64.5±0.4	64.5±0.3	64.4±0.5
	4C	**81.7±0.4**	81.6±0.5	**81.7±0.5**	81.4±0.4	81.5±0.5	81.5±0.4
	5C	61.6±0.5	**61.7±0.4**	61.6±0.4	**61.7±0.3**	61.6±0.3	61.5±0.3
2013	SP	**89.5±0.3**	88.8±0.4	89.2±0.4	89.3±0.2	89.3±0.2	89.4±0.2
	DP	**89.4±0.3**	**89.4±0.1**	89.3±0.2	**89.4±0.2**	**89.4±0.2**	89.3±0.2
	3C	64.3±0.4	**64.4±0.2**	**64.4±0.1**	64.2±0.2	**64.4±0.3**	64.3±0.3
	4C	83.3±0.3	83.4±0.1	83.4±0.3	83.4±0.4	**83.5±0.1**	82.8±0.8
	5C	**64.2±0.3**	64.1±0.2	64.0±0.3	64.1±0.4	64.0±0.3	63.9±0.4
2014	SP	**87.7±0.4**	87.0±0.6	87.5±0.5	**87.7±0.4**	87.4±0.5	87.5±0.3
	DP	87.7±0.5	**87.9±0.5**	**87.9±0.4**	87.8±0.3	87.8±0.4	**87.9±0.4**
	3C	61.8±0.7	**62.0±0.5**	61.8±0.3	61.8±0.3	61.7±0.3	61.9±0.2
	4C	81.3±0.3	**81.5±0.2**	81.4±0.2	81.4±0.3	81.3±0.2	81.2±0.2
	5C	61.9±0.3	**62.0±0.3**	61.9±0.3	61.8±0.4	61.9±0.4	61.8±0.3
2015	SP	**89.8±0.3**	89.3±0.4	89.4±0.5	89.4±0.3	89.4±0.2	89.6±0.4
	DP	**89.9±0.3**	89.7±0.3	89.6±0.5	89.7±0.4	89.7±0.4	89.8±0.3
	3C	64.1±0.4	64.0±0.4	**64.3±0.2**	64.2±0.2	64.2±0.3	64.1±0.2
	4C	84.7±0.3	84.6±0.4	**84.8±0.2**	**84.8±0.1**	84.7±0.2	**84.8±0.2**
	5C	**64.9±0.4**	64.8±0.6	64.8±0.5	64.6±0.7	64.8±0.5	64.8±0.5
2016	SP	**89.0±0.2**	88.7±0.2	88.7±0.3	88.8±0.5	88.5±0.8	88.7±0.3
	DP	88.9±0.2	**89.0±0.3**	88.9±0.4	88.8±0.3	**89.0±0.3**	88.9±0.3
	3C	**62.5±0.2**	62.2±0.5	62.4±0.5	62.4±0.3	62.4±0.3	62.2±0.5
	4C	76.8±0.2	**76.9±0.4**	76.8±0.4	76.6±0.4	76.7±0.5	76.7±0.5
	5C	57.9±0.3	58.0±0.4	58.0±0.6	58.0±0.7	57.9±0.5	**58.1±0.3**
2017	SP	89.3±0.3	89.1±0.3	89.3±0.2	89.2±0.3	**89.4±0.2**	**89.4±0.3**
	DP	89.3±0.3	89.4±0.2	89.4±0.4	89.4±0.2	89.4±0.1	**89.5±0.2**
	3C	61.3±0.3	61.3±0.3	61.2±0.4	**61.4±0.4**	61.3±0.3	61.3±0.2
	4C	**69.5±0.5**	**69.5±0.5**	69.4±0.5	69.3±0.7	69.3±0.6	69.4±0.4
	5C	**52.0±0.4**	51.8±0.5	51.8±0.4	51.6±0.5	51.9±0.6	51.8±0.3
2018	SP	**88.2±0.4**	87.6±0.6	87.8±0.6	87.8±0.6	87.9±0.4	87.7±0.6
	DP	88.1±0.4	88.0±0.4	88.1±0.4	**88.2±0.4**	88.1±0.4	88.1±0.4
	3C	63.7±0.3	64.1±0.6	**64.2±0.7**	63.7±0.7	64.0±0.5	64.0±0.6
	4C	83.0±0.4	82.7±0.4	83.0±0.4	**83.1±0.4**	83.0±0.5	83.0±0.7
	5C	**63.8±0.5**	63.7±0.4	63.7±0.6	63.6±0.5	63.6±0.5	63.7±0.5
2019	SP	**89.3±0.4**	88.7±0.4	89.0±0.3	88.7±0.6	89.1±0.3	89.1±0.5
	DP	**89.4±0.2**	89.3±0.4	**89.4±0.2**	89.2±0.3	89.2±0.1	89.3±0.2
	3C	62.2±0.2	**62.4±0.5**	62.3±0.7	62.2±0.6	**62.4±0.3**	**62.4±0.5**
	4C	**74.5±0.2**	**74.5±0.4**	74.1±0.3	74.4±0.2	74.2±0.3	74.3±0.4
	5C	56.0±0.3	55.9±0.3	**56.1±0.4**	**56.1±0.4**	55.8±0.3	56.0±0.2
2020	SP	**92.3±0.1**	91.8±0.1	91.5±0.3	91.7±0.2	91.6±0.1	91.5±0.1
	DP	**92.2±0.1**	**92.2±0.1**	**92.2±0.1**	92.1±0.2	**92.2±0.1**	92.1±0.1
	3C	69.2±0.3	69.6±0.6	**69.8±0.3**	68.8±0.4	69.0±0.7	69.2±0.4
	4C	85.4±0.2	**85.5±0.2**	85.2±0.8	85.2±0.5	85.1±0.4	84.8±0.4
	5C	**67.8±0.2**	**67.8±0.6**	67.5±0.7	67.4±0.5	67.3±0.6	67.3±0.6

D

GNNRank: Learning Global Rankings from Pairwise Comparisons via Directed Graph Neural Networks Supplementary Information

D.1 Implementation Details

D.1.1 Setup

We use all data for training for at most 1000 epochs, and stop early if the loss does not decrease for 200 epochs. For proximal variants, we use 50 epochs for pretraining. We use Adam [226] and Stochastic Gradient Descend (SGD) as the optimizers and ℓ_2 regularization with weight decay $5 \cdot 10^{-4}$ to avoid overfitting. We use as learning rate 0.01 throughout for GNNRank-N methods as well as pretraining with Adam, and 10 times that of pretraining learning rate for GNNRank-P methods with SGD. We run a grid search on hyperparameters.

For real-world data sets, we conduct 10 repeated runs, while for synthetic data, we generate 5 synthetic networks under the same setting, each with 2 repeated runs.

Note that we do not evaluate our method by the loss which is used to devise the method, as that would not be fair; instead we employ $\mathcal{L}_{upset,naive}$ or $\mathcal{L}_{upset,simple}$ which are never used in training. Thus, the comparison is fair. When ground truths are given, we follow [165] to use Kendall Tau values for comparison.

D.1.2 Codes, Data and Hardware

To fully reproduce our results, codes and preprocessed data are available at `https://github.com/SherylHYX/GNNRank`. Experiments were conducted on a compute node with 8 Nvidia Tesla T4, 96 Intel Xeon Platinum 8259CL CPUs @ 2.50GHz and 378GB RAM. Most experiments can be completed within a week, including all variants, hyperparameter searches and ablation studies, except for those on MVR.

The data sets considered here are relatively small and the same applies to GNNRank's competitive papers. However, even a network with 100 nodes has more than 10^{157} possible rankings; this large scale task needs efficient methods. Although each individual task does not require much resource (often $<$ 5min/run), for the paper we have 78 real-world and 18 synthetic data sets, each requires 10 runs for each of the 36 (proximal innerproduct, proximal dist) + 126 (proximal baseline) + 12 (non-proximal) = 174 variants, thus 167,040 runs, plus an extra ablation study.

D.2 Details on Finding Q for Proximal Gradient Steps

There are infinitely many choices of valid $\mathbf{Q}$; here we construct one of the special $\mathbf{Q}$'s for which we can compute $\mathbf{QL}$ efficiently. Since $\mathbf{Q}^\top \mathbf{e}_1 = \mathbf{1}/\sqrt{n}$, we can construct $\mathbf{R} = \mathbf{Q}^\top$ as a series of matrix multiplication of $(n-1)$ rotation matrices on two adjacent axes

$$\mathbf{R} = \mathbf{R}_{n-1}\mathbf{R}_{n-2}\cdots\mathbf{R}_2\mathbf{R}_1$$

where $\mathbf{R}_k$ is defined by:

$$[\mathbf{R}_k]_{ij} = \begin{cases} \sqrt{\frac{1}{n-k}} & i=k, j=k \text{ or } i=k+1, j=k+1 \\ -\sqrt{\frac{n-k-1}{n-k}} & i=k, j=k+1 \\ \sqrt{\frac{n-k-1}{n-k}} & i=k+1, j=k \\ 1 & i=j, \text{ and } i,j \neq k \text{ or } k+1 \\ 0 & \text{otherwise.} \end{cases}$$

To explain this construction, starting from $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^\top$, we carry out a rotation on the first and second axis to make the first element $\sqrt{1/n}$. The rotation matrix is

$$\mathbf{R}_1 = \begin{bmatrix} \sqrt{\frac{1}{n}} & -\sqrt{\frac{n-1}{n}} & & & \cdots & \\ \sqrt{\frac{n-1}{n}} & \sqrt{\frac{1}{n}} & & & \cdots & \\ & & 1 & & \cdots & \\ & & & 1 & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & & 1 \end{bmatrix}$$

We observe that

$$\mathbf{R}_1\mathbf{e}_1 = \begin{bmatrix} \sqrt{\frac{1}{n}} & \sqrt{\frac{n-1}{n}} & 0 & 0 & \cdots & 0 \end{bmatrix}^\top.$$

The second rotation occurs on the second and third axis to render $\mathbf{R}_2\mathbf{R}_1\mathbf{e}_1 =$

$\begin{bmatrix}\sqrt{\frac{1}{n}} & \sqrt{\frac{1}{n}} & \sqrt{\frac{n-2}{n}} & 0 & \cdots & 0\end{bmatrix}^\top$; this $\mathbf{R}_2$ is

$$\mathbf{R}_2 = \begin{bmatrix} 1 & & & & \cdots & \\ & \sqrt{\frac{1}{n-1}} & -\sqrt{\frac{n-2}{n-1}} & & \cdots & \\ & \sqrt{\frac{n-2}{n-1}} & \sqrt{\frac{1}{n-1}} & & \cdots & \\ & & & 1 & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & & 1 \end{bmatrix}$$

The general matrices R_k are obtained by continuing this construction.

The matrix $\mathbf{R}$ is

$$\begin{bmatrix} \sqrt{\frac{1}{n}} & -\sqrt{\frac{n-1}{n}} & & & \cdots & \\ \sqrt{\frac{1}{n}} & \sqrt{\frac{1}{n(n-1)}} & -\sqrt{\frac{n-2}{n-1}} & & \cdots & \\ \sqrt{\frac{1}{n}} & \sqrt{\frac{1}{n(n-1)}} & \sqrt{\frac{1}{(n-1)(n-2)}} & -\sqrt{\frac{n-3}{n-2}} & \cdots & \\ \sqrt{\frac{1}{n}} & \sqrt{\frac{1}{n(n-1)}} & \sqrt{\frac{1}{(n-1)(n-2)}} & \sqrt{\frac{1}{(n-2)(n-3)}} & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sqrt{\frac{1}{n}} & \sqrt{\frac{1}{n(n-1)}} & \sqrt{\frac{1}{(n-1)(n-2)}} & \sqrt{\frac{1}{(n-2)(n-3)}} & \cdots & \sqrt{\frac{1}{2}} \end{bmatrix}$$

We then put $\mathbf{Q} = \mathbf{R}^\top$.

The resulting $\mathbf{Q}$ is the following upper Hessenberg matrix, which is independent of model parameters and can be efficiently precomputed

$$\mathbf{Q}_{ij} = \begin{cases} \sqrt{\frac{1}{n}} & i = 1 \\ -\sqrt{\frac{n-i+1}{n-i+2}} & i \geq 2, j = i-1 \\ \sqrt{\frac{1}{(n-i+1)(n-i+2)}} & i \geq 2, j \geq i \\ 0 & \text{otherwise.} \end{cases}$$

The computation of $\mathbf{QL}$ takes $O(n^2)$ time since $\mathbf{Q}$ is a summation of (1) an upper-triangular matrix $\mathbf{U}$ with the same non-zero value on the same row, and (2) a shift matrix $\mathbf{V}$ (with negative subdiagonal). Then $\mathbf{UL}$ is essentially a cumulative sum from the bottom row to the top row, followed by a row-wise multiplication of the diagonal of $\mathbf{U}$, and $\mathbf{VL}$ is essentially a combination of row-wise multiplication and indexing.

D.3 Theoretical Analysis and Practical Considerations on Convergence of the Proximal Gradient Steps

First we prove a theorem useful for the main result.

Theorem 5. *Let $\{\alpha_\gamma > 0\}_{\gamma=1}^{\Gamma}$ in Algo. 1 be fixed and equal to α. Let ρ be the Fiedler eigenvalue of $\mathbf{S}$, i.e., the second smallest eigenvalue of $\mathbf{L}$. Let $\mathbf{r}^*$ be a Fiedler eigenvector corresponding to ρ, let $\mathbf{y}^* = [\mathbf{Q}\mathbf{r}^*]_{2:n}$ and $\widetilde{L} = \left[\mathbf{Q}\mathbf{L}\mathbf{Q}^\top\right]_{2:n,2:n}$. Let $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_{n-1} = 0$ be the eigenvalues of $\mathbf{P}\widetilde{\mathbf{L}}\mathbf{P}$, where $\mathbf{P} = \mathbf{I} - \mathbf{y}^*\mathbf{y}^{*\top}$. Let*

$\rho_\alpha = \max_{1\leq i\leq n-2} \frac{|1-2\alpha\lambda_i|}{1-2\alpha\rho}$. *If* $\mathbf{r}^*$ *is a strict local minimum of problem (5.5), and if* $\rho_\alpha < 1$ *and* $\alpha(\lambda_1+\rho) < 1$, *then Algo. 1 converges locally uniformly to* $\mathbf{r}^*$.

Proof. Our problem (5.7) is a special case of the problem (2) in [230] (henceforth equation D.1 further below, of minimizing a quadratic over a sphere), where our fixed point $\mathbf{y}^*$ is determined by $\mathbf{f}_\alpha(\overline{\mathbf{x}}) = \mathcal{P}_{\mathcal{S}^{n-2}}(-\alpha(2\widetilde{\mathbf{L}}\overline{\mathbf{x}})) = \overline{x}$. Recall that the spherical projection operator $\mathcal{P}_{\mathcal{S}^{n-2}}(\cdot) : \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}$ is defined to be $\mathcal{P}_{\mathcal{S}^{n-2}}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ if $\mathbf{x} \neq \mathbf{0}$, and $\mathcal{P}_{\mathcal{S}^{n-2}}(\mathbf{0}) = [1, 0, \ldots, 0]^\top$, where $\mathbf{0} = [0, \ldots, 0]^\top$. It is shown in Lemma 1 of [230] (henceforth Lemma 7 further below) that $\mathbf{y}^*$ with $||\mathbf{y}^*|| = 1$ being a solution of (5.7) is equivalent to the existence of a constant $\rho(\overline{\mathbf{x}})$ such that

$$\widetilde{\mathbf{L}}\mathbf{y}^* = \rho(\mathbf{y}^*) \cdot \mathbf{y}^*,$$

rendering $\rho(\mathbf{y}^*)$ to be an eigenvalue of $\widetilde{\mathbf{L}}$. Theorem 1 in [230] (henceforth Theorem 9 below) states that with an initial step size $\alpha > 0$, such that $2\alpha(\rho(\mathbf{y}^*) + \lambda_1) < 2$, as long as $\rho_\alpha = \max_{1\leq i\leq n-2} \frac{|1-2\alpha\lambda_i|}{1-2\alpha\gamma(\mathbf{y}^*)}) < 1$, the proximal gradients steps will converge to a local minimum.

Note that problem (5.5) and problem (5.7) are equivalent, and there is a one-to-one correspondence between $\mathbf{r}^*$ and $\mathbf{y}^*$, namely, $\mathbf{y}^* = [\mathbf{Q}\mathbf{r}^*]_{2:n} \in \mathbb{R}^{n-1}$, and $\mathbf{r}^* = \text{CONCAT}(0, \left[\mathbf{Q}^\top\mathbf{r}^*\right]) \in \mathbb{R}^n$. Hence, since $\mathbf{r}^*$ is a strict local minimum of problem (5.5), we have that $\mathbf{y}^*$ is a strict local minimum of problem (5.7).

By Lemma 7 from [230], we have $\rho < \lambda_{n-2}$, since $\mathbf{y}^*$ is a strict local minimum. In particular, by Theorem 9 in [230], when $\mathbf{y}^*$ is a strict local minimum to problem equation D.1, if $2\alpha(\rho + \lambda_1) < 2$, i.e., $\alpha(\rho + \lambda_1) < 1$, there is a constant M such that

$$||\mathbf{y}^\gamma - \mathbf{y}^*|| \leq M||\mathbf{y}^0 - \mathbf{y}^*||(\rho_\alpha + o(1))^\gamma,$$

where $\mathbf{y}^0$ is the initial guess at proximal gradient step 0, and γ denotes the step number. As $\rho_\alpha < 1$ is assumed, convergence follows. □

For convergence, it suffices to ensure that $\rho_\alpha < 1$, i.e. to have $2\alpha\gamma(\overline{\mathbf{x}}) < 1$, $|1-2\alpha\lambda_1| < 1-\alpha\gamma(\overline{\mathbf{x}})$ and $|1-2\alpha\lambda_{n-2}| < 1-2\alpha\gamma(\overline{\mathbf{x}})$. In particular, in our problem we take $\rho(\overline{\mathbf{x}}) = \rho$, which is also the smallest eigenvalue of $\widetilde{\mathbf{L}}$, so $\rho < \lambda_{n-2} \leq \lambda_1$. To apply Theorem 5 in order to obtain a convergence guarantee in Algo. 1, as $\lambda_{n-2} > \rho$, (which is equivalent to $\mathbf{r}^*$ being a strict local minimum) it remains to use an α value such that $2\alpha\lambda_1 < 1$. Lemma 8 of [230] also suggests that if $2\alpha\lambda_1 < 1$, then we indeed have $\mathbf{r}^*$ as a global minimizer to problem (5.5).

Finally we have the ingredients to prove the main result.

Theorem 6. *Let* $\{\alpha_\gamma > 0\}_{\gamma=1}^{\Gamma}$ *in Algo. 1 be fixed (equal to* α*) and let* ρ *be the Fiedler eigenvalue of* $\mathbf{S}$. *Denote a Fiedler eigenvector by* $\mathbf{r}^*$. *Assume that* $\mathbf{r}^*$ *is a strict local minimizer of problem (5.5). If* $0 < \alpha < \frac{1}{4(n-1)}$, *then with our definition of the similarity matrix, Algo. 1 converges locally uniformly to* $\mathbf{r}^*$.

Proof. Via our similarity matrix construction process, entries in $\mathbf{S}$ are upper-bounded by 1, so the degree matrix $\mathbf{D}$ has entries bounded above by n, the number of nodes, and lower bounded by 0, and that $\mathbf{L}$ is positive semi-definite. We thus have that the eigenvalues of $\mathbf{L}$ are in $[0, (n-1) + (n-1) \times 1]$ by the Gershgorin disc theorem. Since $\mathbf{Q}$ is orthogonal, we have that eigenvalues of $\widetilde{\mathbf{L}}$ are also within the range $[0, 2(n-1)]$. To see this, suppose $\mathbf{x}$ is an eigenvector of $\mathbf{L}$ with eigenvalue λ, i.e., $\mathbf{L}\mathbf{x} = \lambda\mathbf{x}$, so $\mathbf{Q}\mathbf{L}\mathbf{Q}\top(\mathbf{Q}\mathbf{x}) = \mathbf{Q}\mathbf{L}(\mathbf{Q}\top\mathbf{Q})\mathbf{x} = \mathbf{Q}\mathbf{L}\mathbf{x} = \mathbf{Q}(\mathbf{L}\mathbf{x}) = \mathbf{Q}\lambda\mathbf{x} = \lambda(\mathbf{Q}\mathbf{x})$.

Therefore, the values $\{\lambda_i\}_{i=1}^{n-2}$ in Theorem 5 are bounded above by $2(n-1)$. Thus, we have that $\rho + \lambda_1 \leq 2(n-1) + 2(n-1) = 4(n-1)$.

For convergence, it suffices to ensure that $\rho_\alpha = \max_{1\leq i\leq n-2} \frac{|1-2\alpha\lambda_i|}{1-2\alpha\rho} < 1$, i.e. to have $2\alpha\rho < 1$, $|1-2\alpha\lambda_1| < 1-2\alpha\rho$, and $|1-\alpha\lambda_{n-2}| < 1-2\alpha\rho$.

Thus, since we have $\lambda_{n-2} > \rho$ by Lemma 8 from [230] and the proof of Theorem 5, it remains to use an α value such that $2\alpha\lambda_1 < 1$, then $2\alpha\rho < 2\alpha\lambda_1 < 1$, and $0 < 1-2\alpha\lambda_1 = |1-2\alpha\lambda_1| \leq 1-2\alpha\lambda_{n-2} = |1-2\alpha\lambda_{n-2}| < 1-2\alpha\rho$. With $\alpha < \frac{1}{4(n-1)}$, we indeed have $2\alpha\lambda_1 < 2 \times \frac{1}{4(n-1)} \times 2(n-1) = 1$. □

Our current setting is to initialize all α values to be $\frac{1}{n-1}$. Note that within Algo. 1, the α values are fixed. Although not guaranteed to converge with the initial α, they could be adapted by our outer training loop, i.e. by optimization over the ranking loss function. [230] also suggests the value of the optimal step size to be $\frac{2}{\lambda_1+\lambda_{n-2}}$, which during training our method could in principle reproduce if the optimal step size is actually important for making accurate final rankings.

We remark that real-world weighted networks have the corresponding Fiedler eigenvalue of multiplicity 1; this multiplicity is larger than 1, only for special classes of graphs that exhibit certain symmetries. Our pragmatic assumption that the Fiedler eigenvalue has multiplicity 1, which thus means $\mathbf{r}^*$ being a strict local minimum of problem 5.5, is not a very restrictive one.

For completeness, the remainder of this section recalls the setup and main results from [230]. The problem considered therein is that of minimizing a quadratic form over the sphere

$$\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{B}\mathbf{x} - \mathbf{b}^\top \mathbf{x} \quad \text{s.t.} \quad \|\mathbf{x}\|_2^2 = 1 \tag{D.1}$$

with the matrix $\mathbf{B} \in \mathbb{R}^{m\times m}$ assumed symmetric but not positive semi-definite, hence a non-convex objective function. Upon considering the corresponding Lagrangian function, with ν denoting the Lagrange multiplier, the authors prove the following results, which also constitute building blocks in our analysis. The eigenvalues of $\mathbf{A}$ are given by $\lambda_{min}(\mathbf{B}) \leq \lambda_{m-1} \leq \ldots \leq \lambda_1 \leq \lambda_{max}(\mathbf{B})$.

Lemma 7 (Lemma 1 from [230])**.** *(Stationary conditions). The vector $\mathbf{x}_*$ is a stationary point of problem equation D.1 if and only if $\mathbf{x}_* \in \mathcal{S}^{m-1} = \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x}\| = 1\}$ and there exists a constant $\nu(\mathbf{x}_*)$ such that $\mathbf{r}_* = A\mathbf{x}_* - b = \nu(\mathbf{x}_*) \cdot \mathbf{x}_*$.*

Lemma 8 (Lemma 2 from [230])**.** *A stationary point $\mathbf{x}_*$ of problem equation D.1 is a strict local minimum if and only if $\nu(\mathbf{x}_*) < \lambda_{m-1}(\mathbf{x}_*)$. Furthermore, $\mathbf{x}_*$ is a **global minimizer** of problem equation D.1 if and only if $\nu(\mathbf{x}_*) \leq \lambda_{min}(\mathbf{B})$.*

The following result establishes the result that Projected Gradient Descent (PGD) converges to a local minimum at an asymptotic linear rate ρ_β.

Theorem 9 (Theorem 1 from [230]))**.** *The vector $\mathbf{x}_*$ is a strict local minimum of problem equation D.1, i.e. $\nu < \lambda_{m-1}$, if and only if there exists $\beta > 0$ such that Algorithm 1 [Projected Gradient Descent (PGD)] with step size α converges locally uniformly to $\mathbf{x}_*$. Furthermore, for any step size $\beta > 0$ such that $\beta(\lambda_1 + \nu) < 2$, the sequence $\{\mathbf{x}^{(t)}\}$ satisfies*

$$\|\mathbf{x}^{(t)} - \mathbf{x}_*\| \leq M\|\mathbf{x}^{(0)} - \mathbf{x}_*\| \left(\rho_\alpha + o(1)\right)^t,$$

for some constant $M > 0$ and $\rho_\beta = \max_{1\leq i\leq m-1} \frac{|1-\beta\lambda_i|}{1-\beta\nu}$.

D.4 Detailed Summary Statistics

Table D.1 gives the number of nodes (n), the number of directed edges ($|\mathcal{E}|$), the number of reciprocal edges ($|\mathcal{E}^r|$) (self-loops are counted once and for $u \neq v$, a reciprocal edge $u \to v, v \to u$ is counted twice) as well as their percentage among all edges, for the real-world networks, illustrating the variability in network size and density (defined as $|\mathcal{E}|/[n(n-1)]$). As we do not have input features available, we use the eigengap of the Hermitian matrix $(\mathbf{A} - \mathbf{A}^\top) \cdot i$, where $\mathbf{A}$ is the adjacency matrix and i the imaginary unit, introduced in [97], to determine the value K, which is assumed to be the number of clusters if we are solving a clustering problem. We then stack the real and imaginary parts of the top K eigenvectors of $(\mathbf{A}-\mathbf{A}^\top)\cdot i$ into $2K$-dimensional input features for GNNs. We also report the embedding dimension d used in the experiments, for each of the data sets.

D.5 Full Result Tables

D.5.1 Results on Individual Digraphs and $\mathcal{L}_{\text{upset, ratio}}$

Tables D.2, D.3 and D.4 provide detailed comparison on both $\mathcal{L}_{\text{upset, simple}}$, $\mathcal{L}_{\text{upset, naive}}$ and $\mathcal{L}_{\text{upset, ratio}}$ for all time-series match data, where Table D.4 additionally provides $\mathcal{L}_{\text{upset, ratio}}$ results on other real-world data sets. Again the best-performing variants of the non-proximal and of the proximal method are reported. The proximal method does not perform as well as the non-proximal method in terms of $\mathcal{L}_{\text{upset, ratio}}$, perhaps partially due to the fact that outputs from SerialRank [25] do not align well with skill level but rather relate to relative ordering. It is also natural for GNNRank-N methods that are not motivated by SerialRank to perform well on this metric as $\mathcal{L}_{\text{upset, ratio}}$ is in the training loss function. Therefore, here we only present the results on $\mathcal{L}_{\text{upset, ratio}}$ but do not draw conclusions on which method outperforms others.

D.5.2 Full Results on Synthetic Data

Table D.5 shows Kandall Tau values and corresponding $\mathcal{L}_{\text{upset, naive}}$ on ERO models, extending the results of Table 5.3 in the main text.

D.5.3 Results on Different Variants

Tables D.6, D.7 and D.8 compare different variants in the proposed GNNRank framework with themselves, together with the best and the worst baseline method, across all real-world data sets. Again, $\mathcal{L}_{\text{upset, ratio}}$ results are only shown to illustrate the performance in terms of minimizing the loss function $\mathcal{L}_{\text{upset, ratio}}$, but are not used to compare methods. We conclude that "proximal baseline" with IB as aggregation GNN usually performs the best among the variants. Compared with baselines, our different variants can attain comparable and often superior performance, and are never strongly outperformed.

D.5.4 Ablation Study Full Tables

Tables D.9, D.10 and D.11 extend results of the ablation study to all seasons, and additionally report $\mathcal{L}_{\text{upset, ratio}}$ and $\mathcal{L}_{\text{upset, naive}}$ results. Again, $\mathcal{L}_{\text{upset, ratio}}$ results are only shown to illustrate the performance in terms of minimizing the loss function $\mathcal{L}_{\text{upset, ratio}}$, but are not used to compare methods.

D.5.5 Inductive Learning Full Tables

Tables D.12 and D.14 contain results on the performance of the "IB proximal baseline" variant, trained with the "emb baseline" on the *Basketball finer* data set. The second to last column contains results of directly applying the model trained for season 1985 without further training, while the last column is the result for the model specifically trained for that season. For each year we have 10 runs, and we average over the total of 30 years for each run. On average, directly applying the original trained model gives $\mathcal{L}_{\text{upset, simple}} = 0.75 \pm 0.02$, $\mathcal{L}_{\text{upset, naive}} = 0.19 \pm 0.01$ and $\mathcal{L}_{\text{upset, ratio}} = 0.01 \pm 0.00$, while training specifically for the that season gives $\mathcal{L}_{\text{upset, simple}} = 0.74 \pm 0.00$, $\mathcal{L}_{\text{upset, naive}} = 0.19 \pm 0.00$ and $\mathcal{L}_{\text{upset, ratio}} = 0.01 \pm 0.00$. Thus, in this example, applying the general model produces almost the same superior performance.

D.6 Improvement on Baselines when Employed as Initial Guess for "Proximal Baseline" Variant

Table D.15 shows improvements on $\mathcal{L}_{upset,simple}$ by "proximal baseline" when setting a certain baseline as $\mathbf{r}'$. Across all data sets, "proximal baseline" improves the most, by 1.02, with SyncRank as initial guess, while the average improvement for SpringRank, SerialRank, BTL, Eig.Cent., PageRank and SVD_NRS are 0.07, 0.82, 0.22, 0.19, 0.21, and 0.12, respectively.

Table D.16 shows improvements on $\mathcal{L}_{upset,naive}$ by "proximal baseline" when setting a certain baseline as $\mathbf{r}'$. Across all real data sets, "proximal baseline" improves the most, by 0.24, again with SyncRank as initial guess, while the average improvement for SpringRank, SerialRank, BTL, Eig.Cent., PageRank and SVD_NRS are 0.00, 0.18, 0.04, 0.03, 0.03, and 0.01, respectively.

D.7 Variant and Hyperparameter Selection

The results reported in the main text are selected within either non-proximal or proximal categories depending on whether they have proximal gradient steps within the architecture. All selections are carried out via the lowest $\mathcal{L}_{\text{upset, simple}}$ or the lowest $\mathcal{L}_{\text{upset, naive}}$ or the lowest $\mathcal{L}_{\text{upset, ratio}}$ depending on our objective. "–" in the tables in this section means "not applicable". There are a total of two variants for non-proximal variants and three for proximal variants for score generation, and each can be coupled with one of DIMPA [5] and the inception block model (ib) [114] (or others like MagNet [6], not tested here) for digraph embedding learning. We fix the learning rate to be 0.01, and vary the the method of pretraining, the coefficient for $\mathcal{L}_{\text{upset, margin}}$ in training, the coefficient for $\mathcal{L}_{\text{upset, ratio}}$ in training, and the baseline selected for "proximal baseline" variant. Choices are SyncRank, SpringRank, SerialRank, BTL, Eig.Cent., PageRank and SVD_NRS for real-world data sets, and SpringRank, BTL, SerialRank for synthetic data. We consider these choices to demonstrate that our "proximal baseline" has the ability to improve over initial guess vectors coming from different types of baselines. See Sec. D.6 for more details on improvements over baselines.

Table D.1: Summary statistics for the real-world networks.

Data	n	$\lvert\mathcal{E}\rvert$	density	$\lvert\mathcal{E}^r\rvert$	$\frac{\lvert\mathcal{E}^r\rvert}{\lvert\mathcal{E}\rvert}$(%)	K	d
HeadToHead	602	5010	1.38e-02	464	9.26	48	32
Basketball (1985)	282	2904	3.66e-02	998	34.37	20	16
Basketball finer (1985)	282	4814	6.08e-02	4814	100.00	20	16
Basketball (1986)	283	2937	3.68e-02	1014	34.53	20	16
Basketball finer (1986)	283	4862	6.09e-02	4862	100.00	20	16
Basketball (1987)	290	3045	3.63e-02	1012	33.23	20	16
Basketball finer (1987)	290	5088	6.07e-02	5088	100.00	20	16
Basketball (1988)	290	3099	3.70e-02	1034	33.37	20	16
Basketball finer (1988)	290	5170	6.17e-02	5170	100.00	20	16
Basketball (1989)	293	3162	3.70e-02	1014	32.07	20	16
Basketball finer (1989)	293	5318	6.22e-02	5318	100.00	20	16
Basketball (1990)	292	3192	3.76e-02	1042	32.64	20	16
Basketball finer (1990)	292	5350	6.30e-02	5350	100.00	20	16
Basketball (1991)	295	3218	3.71e-02	1018	31.63	20	16
Basketball finer (1991)	295	5420	6.25e-02	5420	100.00	20	16
Basketball (1992)	298	3238	3.66e-02	1036	32.00	20	16
Basketball finer (1992)	298	5444	6.15e-02	5444	100.00	20	16
Basketball (1993)	298	3088	3.49e-02	1024	33.16	20	16
Basketball finer (1993)	298	5160	5.83e-02	5160	100.00	20	16
Basketball (1994)	301	3144	3.48e-02	1044	33.21	20	16
Basketball finer (1994)	301	5252	5.82e-02	5252	100.00	20	16
Basketball (1995)	302	3182	3.50e-02	1034	32.50	20	16
Basketball finer (1995)	302	5336	5.87e-02	5336	100.00	20	16
Basketball (1996)	305	3256	3.51e-02	1026	31.51	20	16
Basketball finer (1996)	305	5498	5.93e-02	5498	100.00	20	16
Basketball (1997)	305	3333	3.59e-02	1044	31.32	20	16
Basketball finer (1997)	305	5628	6.07e-02	5628	100.00	20	16
Basketball (1998)	306	3321	3.56e-02	966	29.09	20	16
Basketball finer (1998)	306	5684	6.09e-02	5684	100.00	20	16
Basketball (1999)	310	3385	3.53e-02	998	29.48	20	16
Basketball finer (1999)	310	5788	6.04e-02	5788	100.00	20	16
Basketball (2000)	318	3475	3.45e-02	852	24.52	20	16
Basketball finer (2000)	318	6274	6.22e-02	6274	100.00	20	16
Basketball (2001)	318	3405	3.38e-02	904	26.55	20	16
Basketball finer (2001)	318	6116	6.07e-02	6116	100.00	20	16
Basketball (2002)	321	3505	3.41e-02	976	27.85	20	16
Basketball finer (2002)	321	6192	6.03e-02	6192	100.00	20	16
Basketball (2003)	327	3560	3.34e-02	954	26.80	20	16
Basketball finer (2003)	327	6356	5.96e-02	6356	100.00	20	16
Basketball (2004)	326	3527	3.33e-02	952	26.99	20	16
Basketball finer (2004)	326	6316	5.96e-02	6316	100.00	20	16
Basketball (2005)	330	3622	3.34e-02	946	26.12	20	16
Basketball finer (2005)	330	6476	5.96e-02	6476	100.00	20	16
Basketball (2006)	334	3695	3.32e-02	924	25.01	20	16
Basketball finer (2006)	334	6680	6.01e-02	6680	100.00	20	16
Basketball (2007)	336	3974	3.53e-02	976	24.56	20	16
Basketball finer (2007)	336	7186	6.38e-02	7186	100.00	20	16
Basketball (2008)	342	4051	3.47e-02	972	23.99	20	16
Basketball finer (2008)	342	7386	6.33e-02	7386	100.00	20	16
Basketball (2009)	347	4155	3.46e-02	1046	25.17	20	16
Basketball finer (2009)	347	7478	6.23e-02	7478	100.00	20	16
Basketball (2010)	347	4133	3.44e-02	916	22.16	20	16
Basketball finer (2010)	347	7538	6.28e-02	7538	100.00	20	16
Basketball (2011)	345	4086	3.44e-02	950	23.25	20	16
Basketball finer (2011)	345	7504	6.32e-02	7504	100.00	20	16
Basketball (2012)	345	4126	3.48e-02	950	23.02	20	16
Basketball finer (2012)	345	7580	6.39e-02	7580	100.00	20	16
Basketball (2013)	347	4153	3.46e-02	960	23.12	20	16
Basketball finer (2013)	347	7616	6.34e-02	7616	100.00	20	16
Basketball (2014)	351	4196	3.42e-02	1008	24.02	20	16
Basketball finer (2014)	351	7650	6.23e-02	7650	100.00	20	16
Football (2009)	20	215	5.66e-01	78	36.28	9	8
Football finer (2009)	20	380	1.00e+00	380	100.00	9	8
Football (2010)	20	219	5.76e-01	86	39.27	9	8
Football finer (2010)	20	380	1.00e+00	380	100.00	9	8
Football (2011)	20	226	5.95e-01	92	40.71	9	8
Football finer (2011)	20	380	1.00e+00	380	100.00	9	8
Football (2012)	20	216	5.68e-01	86	39.81	9	8
Football finer (2012)	20	380	1.00e+00	380	100.00	9	8
Football (2013)	20	222	5.84e-01	82	36.94	9	8
Football finer (2013)	20	380	1.00e+00	380	100.00	9	8
Football (2014)	20	107	2.82e-01	0	0.00	9	8
Football finer (2014)	20	300	7.89e-01	300	100.00	9	8
Football(avg)	20	201	5.29e-01	71	32.17	9	8
Basketball(avg)	316	3506	3.51e-02	986	28.57	20	16
Football finer(avg)	20	367	9.65e-01	367	100	9	8
Basketball finer(avg)	316	6139	6.12e-02	6139	100	20	16
Animal	21	193	4.60e-01	64	33.16	3	8
Finance	1315	1729225	1.00e+00	1729225	100	20	64
Faculty:Business	113	1787	1.41e-01	0	0.00	5	16
Faculty:CS	206	1407	3.33e-02	0	0.00	9	16
Faculty:History	145	1204	5.77e-02	0	0.00	12	16

Table D.2: Result table on $\mathcal{L}_{\text{upset, simple}}$ for individual directed graphs, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue . When MVR could not generate results after a week, we omit the results and fill in "NAN" here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	MVR	GNNRank-N	GNNRank-P
HeadToHead	1.00±0.00	1.94±0.00	2.01±0.00	1.12±0.01	1.16±0.00	1.47±0.00	1.36±0.00	2.00±0.02	1.79±0.00	1.42±0.00	nan±nan	0.99±0.00	**0.96±0.00**
Finance	1.63±0.00	1.98±0.00	1.61±0.00	1.78±0.01	1.63±0.00	1.74±0.00	1.75±0.00	1.88±0.00	1.64±0.00	1.64±0.00	nan±nan	**1.00±0.00**	**1.00±0.00**
Animal	0.50±0.00	1.62±0.24	1.98±0.48	0.45±0.00	0.33±0.00	0.55±0.00	0.63±0.00	1.96±0.00	1.03±0.00	0.53±0.00	2.02±0.32	0.41±0.09	**0.25±0.00**
Faculty: Business	0.41±0.00	0.83±0.00	1.19±0.00	0.41±0.01	0.49±0.00	0.49±0.00	0.49±0.00	2.01±0.03	0.68±0.00	0.46±0.00	0.78±0.05	0.38±0.01	**0.36±0.00**
Faculty: CS	0.33±0.00	0.98±0.10	1.40±0.00	0.34±0.01	0.61±0.00	0.51±0.00	0.44±0.00	1.99±0.27	0.93±0.00	0.58±0.00	0.87±0.09	0.33±0.03	**0.32±0.00**
Faculty: History	0.32±0.00	0.57±0.00	2.16±0.80	0.30±0.01	0.57±0.00	0.40±0.00	0.37±0.00	2.13±0.30	0.95±0.00	0.38±0.00	0.84±0.17	**0.28±0.01**	0.30±0.01
Basketball(1985)	0.76±0.00	1.62±0.00	2.11±0.00	0.93±0.08	0.75±0.00	0.89±0.00	0.91±0.00	1.92±0.00	1.00±0.00	0.86±0.00	nan±nan	0.77±0.01	**0.71±0.00**
Basketball(1986)	0.78±0.00	1.79±0.00	1.99±0.00	0.96±0.02	0.78±0.00	0.89±0.00	0.88±0.00	1.96±0.00	1.09±0.00	0.86±0.00	nan±nan	0.78±0.01	**0.69±0.00**
Basketball(1987)	0.82±0.00	1.79±0.00	1.88±0.00	0.99±0.02	0.85±0.00	0.99±0.00	0.90±0.00	1.94±0.00	0.99±0.00	0.91±0.00	nan±nan	0.84±0.01	**0.77±0.01**
Basketball(1988)	0.74±0.00	1.66±0.00	1.91±0.00	0.92±0.03	0.77±0.00	0.87±0.00	0.85±0.00	1.93±0.00	0.97±0.00	0.83±0.00	nan±nan	0.76±0.01	**0.70±0.00**
Basketball(1989)	0.75±0.00	1.74±0.00	1.86±0.00	0.94±0.02	0.77±0.00	0.82±0.00	0.85±0.00	1.87±0.00	0.98±0.00	0.90±0.00	nan±nan	0.78±0.02	**0.70±0.00**
Basketball(1990)	0.75±0.00	1.68±0.00	1.94±0.00	0.89±0.08	0.78±0.00	0.90±0.00	0.88±0.00	1.95±0.00	0.91±0.00	0.85±0.00	nan±nan	0.77±0.02	**0.70±0.00**
Basketball(1991)	0.80±0.00	1.80±0.00	2.03±0.00	0.92±0.07	0.77±0.00	0.86±0.00	0.84±0.00	1.95±0.00	1.00±0.00	0.90±0.00	nan±nan	0.78±0.01	**0.70±0.00**
Basketball(1992)	0.75±0.00	1.74±0.00	1.89±0.00	0.86±0.10	0.72±0.00	0.84±0.00	0.80±0.00	2.00±0.00	0.96±0.00	0.84±0.00	nan±nan	0.74±0.01	**0.67±0.00**
Basketball(1993)	0.75±0.00	1.68±0.00	2.04±0.00	0.88±0.06	0.74±0.00	0.83±0.00	0.85±0.00	1.93±0.00	0.99±0.00	0.86±0.00	nan±nan	0.76±0.02	**0.68±0.01**
Basketball(1994)	0.74±0.00	1.67±0.00	2.00±0.00	0.77±0.09	0.74±0.00	0.82±0.00	0.83±0.00	2.04±0.00	0.90±0.00	0.84±0.00	nan±nan	0.75±0.01	**0.69±0.00**
Basketball(1995)	0.77±0.00	1.78±0.00	1.88±0.00	0.84±0.10	0.78±0.00	0.83±0.00	0.87±0.00	2.02±0.00	0.94±0.00	0.86±0.00	nan±nan	0.78±0.01	**0.72±0.01**
Basketball(1996)	0.82±0.00	1.67±0.00	2.05±0.00	0.96±0.07	0.81±0.00	0.90±0.00	0.92±0.00	2.05±0.00	1.08±0.00	0.95±0.00	nan±nan	0.84±0.01	**0.77±0.01**
Basketball(1997)	0.81±0.00	1.72±0.00	1.95±0.00	0.97±0.02	0.83±0.00	0.91±0.00	0.90±0.00	1.89±0.00	0.96±0.00	0.92±0.00	nan±nan	0.83±0.01	**0.76±0.01**
Basketball(1998)	0.77±0.00	1.69±0.00	1.91±0.00	0.96±0.03	0.79±0.00	0.90±0.00	0.85±0.00	1.92±0.00	0.96±0.00	0.89±0.00	nan±nan	0.80±0.01	**0.74±0.01**
Basketball(1999)	0.83±0.00	1.56±0.00	2.02±0.00	0.97±0.06	0.81±0.00	0.93±0.00	0.94±0.00	1.90±0.00	1.16±0.00	0.95±0.00	nan±nan	0.84±0.01	**0.74±0.00**
Basketball(2000)	0.84±0.00	1.77±0.00	1.97±0.00	0.98±0.02	0.86±0.00	0.91±0.00	0.91±0.00	1.91±0.00	1.11±0.00	0.94±0.00	nan±nan	0.82±0.01	**0.77±0.00**
Basketball(2001)	0.75±0.00	1.73±0.00	2.03±0.00	0.91±0.07	0.78±0.00	0.88±0.00	0.84±0.00	2.00±0.00	1.05±0.00	0.94±0.00	nan±nan	0.77±0.01	**0.71±0.00**
Basketball(2002)	0.80±0.00	1.69±0.00	1.91±0.00	0.92±0.05	0.80±0.00	0.85±0.00	0.91±0.00	1.94±0.00	1.06±0.00	0.88±0.00	nan±nan	0.80±0.01	**0.73±0.00**
Basketball(2003)	0.81±0.00	1.85±0.00	2.05±0.00	0.95±0.07	0.80±0.00	0.94±0.00	0.94±0.00	1.93±0.00	0.99±0.00	0.94±0.00	nan±nan	0.83±0.01	**0.78±0.00**
Basketball(2004)	0.75±0.00	1.67±0.00	2.00±0.11	0.89±0.07	0.75±0.00	0.85±0.00	0.87±0.00	1.95±0.00	0.92±0.00	0.86±0.00	nan±nan	0.77±0.01	**0.69±0.00**
Basketball(2005)	0.78±0.00	1.78±0.01	1.98±0.05	0.98±0.02	0.81±0.00	0.85±0.00	0.91±0.00	1.87±0.00	1.05±0.00	0.92±0.00	nan±nan	0.81±0.02	**0.73±0.00**
Basketball(2006)	0.81±0.00	1.78±0.00	2.00±0.00	0.84±0.08	0.77±0.00	0.86±0.00	0.89±0.00	1.99±0.00	1.03±0.00	0.91±0.00	nan±nan	0.82±0.01	**0.74±0.00**
Basketball(2007)	0.79±0.00	1.87±0.00	1.94±0.08	0.94±0.01	0.84±0.00	0.89±0.00	0.89±0.00	1.94±0.00	0.92±0.00	0.89±0.00	nan±nan	0.83±0.01	**0.77±0.00**
Basketball(2008)	0.82±0.00	1.75±0.00	2.01±0.00	0.90±0.07	0.80±0.00	0.90±0.00	0.93±0.00	1.92±0.00	0.98±0.00	0.90±0.00	nan±nan	0.83±0.01	**0.78±0.00**
Basketball(2009)	0.78±0.00	1.74±0.00	2.02±0.00	0.94±0.02	0.76±0.00	0.85±0.00	0.86±0.00	1.89±0.00	0.95±0.00	0.88±0.00	nan±nan	0.78±0.01	**0.72±0.01**
Basketball(2010)	0.79±0.00	1.55±0.00	1.97±0.00	0.85±0.10	0.77±0.00	0.92±0.00	0.91±0.00	2.03±0.00	0.93±0.00	0.91±0.00	nan±nan	0.82±0.01	**0.71±0.00**
Basketball(2011)	0.81±0.00	1.63±0.00	1.96±0.00	0.85±0.08	0.79±0.00	0.88±0.00	0.89±0.00	2.03±0.00	0.95±0.00	0.87±0.00	nan±nan	0.80±0.01	**0.74±0.00**
Basketball(2012)	0.76±0.00	1.81±0.00	1.97±0.00	0.88±0.08	0.77±0.00	0.85±0.00	0.84±0.00	1.96±0.00	0.90±0.00	0.85±0.00	nan±nan	0.78±0.01	**0.69±0.00**
Basketball(2013)	0.80±0.00	1.71±0.00	1.99±0.05	0.95±0.06	0.80±0.00	0.90±0.00	0.87±0.00	1.95±0.00	1.00±0.00	0.90±0.00	nan±nan	0.81±0.01	**0.76±0.00**
Basketball(2014)	0.80±0.00	1.72±0.00	2.06±0.00	0.87±0.09	0.79±0.00	0.88±0.00	0.91±0.00	1.94±0.00	1.01±0.00	0.87±0.00	nan±nan	0.83±0.01	**0.75±0.00**
Basketball finer(1985)	0.76±0.00	1.63±0.00	1.96±0.10	1.46±0.05	0.83±0.00	1.18±0.00	1.16±0.00	1.97±0.00	1.00±0.00	0.87±0.00	nan±nan	0.79±0.01	**0.71±0.00**
Basketball finer(1986)	0.77±0.00	1.81±0.00	1.99±0.00	1.42±0.06	0.84±0.00	1.16±0.00	1.15±0.00	1.99±0.00	1.09±0.00	0.86±0.00	nan±nan	0.81±0.01	**0.69±0.00**
Basketball finer(1987)	0.82±0.00	1.79±0.00	1.87±0.00	1.41±0.06	0.89±0.00	1.17±0.00	1.21±0.00	1.95±0.00	0.99±0.00	0.91±0.00	nan±nan	0.86±0.02	**0.77±0.01**
Basketball finer(1988)	0.78±0.00	1.79±0.00	1.90±0.00	1.43±0.10	0.84±0.00	1.23±0.00	1.19±0.00	1.97±0.00	0.97±0.00	0.83±0.00	nan±nan	0.80±0.01	**0.70±0.01**
Basketball finer(1989)	0.77±0.00	1.67±0.00	1.86±0.00	1.43±0.05	0.83±0.00	1.13±0.00	1.14±0.00	1.94±0.00	0.99±0.00	0.90±0.00	nan±nan	0.82±0.02	**0.70±0.00**
Basketball finer(1990)	0.79±0.00	1.67±0.00	1.93±0.00	1.45±0.05	0.82±0.00	1.28±0.00	1.17±0.00	1.98±0.00	0.91±0.00	0.84±0.00	nan±nan	0.80±0.02	**0.71±0.00**
Basketball finer(1991)	0.81±0.00	1.83±0.00	2.03±0.00	1.36±0.06	0.83±0.00	1.38±0.00	1.31±0.00	1.97±0.00	0.99±0.00	0.89±0.00	nan±nan	0.83±0.01	**0.71±0.00**
Basketball finer(1992)	0.73±0.00	1.72±0.00	1.88±0.00	1.33±0.06	0.77±0.00	1.26±0.00	1.21±0.00	1.87±0.00	0.95±0.00	0.84±0.00	nan±nan	0.75±0.02	**0.67±0.00**
Basketball finer(1993)	0.75±0.00	1.66±0.00	2.03±0.00	1.35±0.05	0.78±0.00	1.18±0.00	1.10±0.00	1.97±0.00	0.98±0.00	0.86±0.00	nan±nan	0.80±0.01	**0.68±0.01**
Basketball finer(1994)	0.74±0.00	1.69±0.00	2.01±0.00	1.35±0.08	0.78±0.00	1.23±0.00	1.10±0.00	1.94±0.00	0.90±0.00	0.83±0.00	nan±nan	0.77±0.01	**0.67±0.00**
Basketball finer(1995)	0.79±0.00	1.78±0.00	1.89±0.00	1.35±0.06	0.83±0.00	1.19±0.00	1.13±0.00	1.92±0.01	0.95±0.00	0.87±0.00	nan±nan	0.78±0.01	**0.72±0.01**
Basketball finer(1996)	0.81±0.00	1.67±0.00	1.95±0.00	1.44±0.06	0.88±0.00	1.22±0.00	1.20±0.00	1.94±0.00	1.08±0.00	0.95±0.00	nan±nan	0.86±0.01	**0.77±0.00**
Basketball finer(1997)	0.83±0.00	1.77±0.00	1.94±0.00	1.40±0.04	0.86±0.00	1.19±0.00	1.16±0.00	2.05±0.00	0.96±0.00	0.92±0.00	nan±nan	0.85±0.01	**0.75±0.01**
Basketball finer(1998)	0.78±0.00	1.70±0.00	1.92±0.00	1.36±0.07	0.83±0.00	1.14±0.00	1.13±0.00	1.91±0.00	0.97±0.00	0.90±0.00	nan±nan	0.82±0.02	**0.73±0.01**
Basketball finer(1999)	0.81±0.00	1.64±0.00	2.02±0.00	1.38±0.07	0.86±0.00	1.17±0.00	1.11±0.00	1.99±0.00	1.17±0.00	0.94±0.00	nan±nan	0.86±0.01	**0.73±0.00**
Basketball finer(2000)	0.84±0.00	1.75±0.00	1.97±0.00	1.39±0.05	0.90±0.00	1.26±0.00	1.18±0.00	1.92±0.00	1.12±0.00	0.95±0.00	nan±nan	0.88±0.01	**0.78±0.00**
Basketball finer(2001)	0.81±0.00	1.69±0.00	2.06±0.00	1.41±0.06	0.86±0.00	1.25±0.00	1.18±0.00	2.03±0.00	1.08±0.00	0.97±0.00	nan±nan	0.86±0.01	**0.73±0.00**
Basketball finer(2002)	0.87±0.00	1.75±0.00	1.86±0.00	1.43±0.08	0.89±0.00	1.20±0.00	1.13±0.00	2.03±0.00	1.07±0.00	0.92±0.00	nan±nan	0.83±0.01	**0.77±0.01**
Basketball finer(2003)	0.87±0.00	1.78±0.00	1.98±0.07	1.46±0.09	0.91±0.00	1.18±0.00	1.14±0.00	2.00±0.00	1.02±0.00	0.95±0.00	nan±nan	0.88±0.02	**0.78±0.00**
Basketball finer(2004)	0.77±0.00	1.71±0.06	1.87±0.00	1.41±0.06	0.80±0.00	1.17±0.00	1.13±0.00	1.98±0.02	0.95±0.00	0.88±0.00	nan±nan	0.84±0.01	**0.71±0.01**
Basketball finer(2005)	0.84±0.00	1.82±0.00	1.93±0.00	1.38±0.06	0.88±0.00	1.14±0.00	1.09±0.00	2.00±0.00	1.08±0.00	0.95±0.00	nan±nan	0.86±0.01	**0.75±0.01**
Basketball finer(2006)	0.86±0.00	1.76±0.00	1.97±0.00	1.40±0.07	0.85±0.00	1.21±0.00	1.11±0.00	1.96±0.00	1.06±0.00	0.94±0.00	nan±nan	0.88±0.02	**0.76±0.00**
Basketball finer(2007)	0.86±0.00	1.85±0.00	1.97±0.14	1.39±0.05	0.93±0.00	1.15±0.00	1.09±0.00	1.92±0.00	0.95±0.00	0.93±0.00	nan±nan	0.87±0.01	**0.80±0.01**
Basketball finer(2008)	0.85±0.00	1.72±0.00	1.98±0.00	1.36±0.08	0.88±0.00	1.20±0.00	1.13±0.00	1.96±0.00	0.99±0.00	0.91±0.00	nan±nan	0.86±0.02	**0.78±0.00**
Basketball finer(2009)	0.84±0.00	1.71±0.00	2.00±0.04	1.37±0.04	0.90±0.00	1.17±0.00	1.12±0.00	2.06±0.00	0.99±0.00	0.91±0.00	nan±nan	0.85±0.01	**0.75±0.00**
Basketball finer(2010)	0.82±0.00	1.68±0.00	1.98±0.00	1.34±0.06	0.84±0.00	1.10±0.00	1.13±0.00	1.97±0.00	0.94±0.00	0.92±0.00	nan±nan	0.85±0.01	**0.75±0.00**
Basketball finer(2011)	0.85±0.00	1.65±0.00	1.96±0.00	1.36±0.06	0.87±0.00	1.12±0.00	1.14±0.00	2.01±0.00	0.96±0.00	0.89±0.00	nan±nan	0.87±0.01	**0.77±0.00**
Basketball finer(2012)	0.80±0.00	1.68±0.00	1.97±0.00	1.37±0.07	0.83±0.00	1.17±0.00	1.10±0.00	1.91±0.00	0.93±0.00	0.89±0.00	nan±nan	0.83±0.01	**0.74±0.01**
Basketball finer(2013)	0.83±0.00	1.73±0.00	2.06±0.00	1.37±0.07	0.88±0.00	1.15±0.00	1.13±0.00	1.97±0.00	1.01±0.00	0.92±0.00	nan±nan	0.85±0.00	**0.78±0.01**
Basketball finer(2014)	0.84±0.00	1.77±0.19	2.07±0.00	1.42±0.09	0.86±0.00	1.18±0.00	1.13±0.00	1.97±0.00	1.02±0.00	0.88±0.00	nan±nan	0.87±0.01	**0.78±0.00**
Football(2009)	**0.61±0.00**	1.44±0.44	0.93±0.00	0.67±0.07	0.80±0.00	0.93±0.00	1.05±0.00	1.80±0.09	0.80±0.00	0.66±0.00	1.55±0.16	0.75±0.06	**0.61±0.00**
Football(2010)	1.19±0.00	1.81±0.15	1.54±0.00	1.18±0.04	1.29±0.00	1.34±0.00	1.37±0.00	1.84±0.13	1.24±0.00	1.17±0.00	1.94±0.20	**0.90±0.02**	0.95±0.05
Football(2011)	**0.80±0.00**	1.57±0.37	0.95±0.00	0.86±0.04	0.87±0.00	0.99±0.00	1.02±0.00	1.60±0.00	0.92±0.00	0.85±0.00	1.80±0.19	0.83±0.03	**0.80±0.01**
Football(2012)	0.97±0.00	1.50±0.03	1.41±0.00	0.99±0.05	0.89±0.00	0.94±0.00	0.99±0.00	1.65±0.00	1.02±0.00	0.86±0.00	1.73±0.18	**0.75±0.03**	0.80±0.08
Football(2013)	0.75±0.00	1.64±0.10	0.78±0.00	0.77±0.05	0.68±0.00	0.92±0.00	0.87±0.00	1.93±0.09	0.90±0.00	0.75±0.00	1.59±0.11	0.64±0.04	**0.56±0.00**
Football(2014)	1.12±0.00	1.81±0.18	1.57±0.00	1.16±0.01	1.09±0.00	1.27±0.00	1.20±0.00	1.83±0.11	1.12±0.00	1.08±0.00	1.72±0.20	**0.96±0.12**	**0.96±0.07**
Football finer(2009)	0.71±0.00	1.78±0.00	0.93±0.00	0.79±0.10	0.71±0.00	1.05±0.00	1.10±0.00	1.89±0.05	0.80±0.00	0.66±0.00	1.91±0.17	0.76±0.18	**0.65±0.02**
Football finer(2010)	1.29±0.00	1.63±0.02	1.44±0.00	1.31±0.09	1.29±0.00	1.37±0.00	1.49±0.00	1.96±0.00	1.24±0.00	1.17±0.00	2.11±0.01	1.00±0.01	**0.99±0.02**
Football finer(2011)	0.87±0.00	1.58±0.02	1.07±0.00	0.93±0.04	0.92±0.00	1.02±0.00	1.14±0.00	1.87±0.00	0.92±0.00	0.85±0.00	2.14±0.02	0.92±0.16	**0.84±0.02**
Football finer(2012)	0.97±0.00	1.76±0.18	1.10±0.00	0.99±0.05	0.92±0.00	1.10±0.00	1.20±0.00	2.03±0.06	1.02±0.00	**0.86±0.00**	2.15±0.01	0.93±0.07	**0.86±0.03**
Football finer(2013)	0.90±0.00	1.57±0.01	0.87±0.00	0.85±0.06	0.73±0.00	0.99±0.00	1.07±0.00	2.01±0.00	0.90±0.00	0.75±0.00	1.94±0.14	0.73±0.19	**0.57±0.01**
Football finer(2014)	1.16±0.00	1.74±0.04	1.57±0.00	1.19±0.04	1.05±0.00	1.23±0.00	1.23±0.00	1.68±0.00	1.12±0.00	1.08±0.00	2.09±0.17	**1.00±0.00**	**1.00±0.03**

Table D.3: Result table on $\mathcal{L}_{\text{upset, naive}}$ for individual directed graphs, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue . When MVR could not generate results after a week, we omit the results and fill in "NAN" here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	MVR	GNNRank-N	GNNRank-P
HeadToHead	0.25±0.00	0.48±0.00	0.50±0.00	0.28±0.00	0.29±0.00	0.37±0.00	0.34±0.00	0.50±0.01	0.45±0.00	0.36±0.00	nan±nan	0.27±0.00	**0.24±0.00**
Finance	0.41±0.00	0.50±0.00	**0.40±0.00**	0.45±0.00	0.41±0.00	0.44±0.00	0.44±0.00	0.47±0.00	0.41±0.00	0.41±0.00	nan±nan	0.41±0.00	**0.40±0.00**
Animal	0.13±0.00	0.40±0.06	0.58±0.11	0.11±0.00	0.08±0.00	0.14±0.00	0.16±0.00	0.49±0.00	0.26±0.00	0.13±0.00	0.50±0.08	0.10±0.02	**0.06±0.00**
Faculty: Business	0.10±0.00	0.21±0.00	0.30±0.00	0.10±0.00	0.12±0.00	0.12±0.00	0.12±0.00	0.50±0.01	0.17±0.00	0.12±0.00	0.19±0.01	0.10±0.00	**0.09±0.00**
Faculty: CS	**0.08±0.00**	0.24±0.02	0.35±0.00	**0.08±0.00**	0.15±0.00	0.13±0.00	0.11±0.00	0.50±0.07	0.23±0.00	0.15±0.00	0.22±0.02	**0.08±0.01**	**0.08±0.00**
Faculty: History	0.08±0.00	0.14±0.00	0.54±0.20	0.08±0.00	0.15±0.00	0.10±0.00	0.09±0.00	0.53±0.08	0.24±0.00	0.10±0.00	0.21±0.04	**0.07±0.00**	**0.07±0.00**
Basketball(1985)	0.19±0.00	0.40±0.00	0.53±0.00	0.23±0.02	0.19±0.00	0.22±0.00	0.23±0.00	0.48±0.00	0.25±0.00	0.22±0.00	nan±nan	0.19±0.00	**0.18±0.00**
Basketball(1986)	0.19±0.00	0.45±0.00	0.50±0.00	0.24±0.01	0.19±0.00	0.22±0.00	0.22±0.00	0.49±0.00	0.27±0.00	0.21±0.00	nan±nan	0.20±0.00	**0.17±0.00**
Basketball(1987)	0.21±0.00	0.45±0.00	0.47±0.00	0.25±0.00	0.21±0.00	0.25±0.00	0.22±0.00	0.48±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(1988)	0.19±0.00	0.42±0.00	0.48±0.00	0.23±0.01	0.19±0.00	0.22±0.00	0.21±0.00	0.48±0.00	0.24±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(1989)	0.19±0.00	0.44±0.00	0.47±0.00	0.24±0.01	0.19±0.00	0.20±0.00	0.21±0.00	0.47±0.00	0.25±0.00	0.23±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(1990)	0.19±0.00	0.42±0.00	0.48±0.00	0.22±0.02	0.20±0.00	0.23±0.00	0.22±0.00	0.49±0.00	0.23±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(1991)	0.20±0.00	0.45±0.00	0.51±0.00	0.23±0.02	0.19±0.00	0.22±0.00	0.21±0.00	0.49±0.00	0.25±0.00	0.22±0.00	nan±nan	0.19±0.00	**0.18±0.00**
Basketball(1992)	0.19±0.00	0.43±0.00	0.47±0.00	0.21±0.02	0.18±0.00	0.21±0.00	0.20±0.00	0.50±0.00	0.24±0.00	0.21±0.00	nan±nan	0.18±0.00	**0.17±0.00**
Basketball(1993)	0.19±0.00	0.42±0.00	0.51±0.00	0.22±0.02	0.19±0.00	0.21±0.00	0.21±0.00	0.48±0.00	0.25±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(1994)	0.19±0.00	0.42±0.00	0.50±0.00	0.19±0.02	0.18±0.00	0.20±0.00	0.21±0.00	0.51±0.00	0.23±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(1995)	0.19±0.00	0.44±0.00	0.47±0.00	0.21±0.02	0.19±0.00	0.21±0.00	0.22±0.00	0.50±0.00	0.24±0.00	0.22±0.00	nan±nan	0.19±0.00	**0.18±0.00**
Basketball(1996)	0.21±0.00	0.42±0.00	0.51±0.00	0.24±0.02	0.20±0.00	0.23±0.00	0.23±0.00	0.51±0.00	0.27±0.00	0.24±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(1997)	0.20±0.00	0.43±0.00	0.49±0.00	0.24±0.01	0.21±0.00	0.23±0.00	0.23±0.00	0.47±0.00	0.24±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(1998)	0.19±0.00	0.42±0.00	0.48±0.00	0.24±0.01	0.20±0.00	0.22±0.00	0.21±0.00	0.48±0.00	0.24±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball(1999)	0.21±0.00	0.39±0.00	0.50±0.00	0.24±0.02	0.20±0.00	0.23±0.00	0.24±0.00	0.47±0.00	0.29±0.00	0.24±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball(2000)	0.21±0.00	0.44±0.00	0.49±0.00	0.25±0.01	0.21±0.00	0.23±0.00	0.23±0.00	0.48±0.00	0.28±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(2001)	0.19±0.00	0.43±0.00	0.51±0.00	0.23±0.02	0.19±0.00	0.22±0.00	0.21±0.00	0.50±0.00	0.26±0.00	0.23±0.00	nan±nan	0.19±0.00	**0.18±0.00**
Basketball(2002)	0.20±0.00	0.42±0.00	0.48±0.00	0.23±0.01	0.20±0.00	0.21±0.00	0.23±0.00	0.49±0.00	0.26±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball(2003)	0.20±0.00	0.46±0.00	0.51±0.00	0.24±0.02	0.20±0.00	0.24±0.00	0.23±0.00	0.48±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(2004)	0.19±0.00	0.42±0.00	0.50±0.03	0.22±0.02	0.19±0.00	0.21±0.00	0.22±0.00	0.49±0.00	0.23±0.00	0.22±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(2005)	0.19±0.00	0.45±0.00	0.49±0.01	0.24±0.01	0.20±0.00	0.21±0.00	0.23±0.00	0.47±0.00	0.26±0.00	0.23±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball(2006)	0.20±0.00	0.44±0.00	0.50±0.00	0.21±0.02	**0.19±0.00**	0.22±0.00	0.22±0.00	0.50±0.00	0.26±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(2007)	0.20±0.00	0.47±0.00	0.48±0.02	0.24±0.00	0.21±0.00	0.22±0.00	0.22±0.00	0.49±0.00	0.23±0.00	0.22±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball(2008)	**0.20±0.00**	0.44±0.00	0.50±0.00	0.22±0.02	**0.20±0.00**	0.22±0.00	0.23±0.00	0.48±0.00	0.24±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.20±0.00**
Basketball(2009)	0.19±0.00	0.43±0.00	0.51±0.00	0.24±0.00	0.19±0.00	0.21±0.00	0.21±0.00	0.47±0.00	0.24±0.00	0.22±0.00	nan±nan	0.19±0.00	**0.18±0.00**
Basketball(2010)	0.20±0.00	0.39±0.00	0.49±0.00	0.21±0.02	0.19±0.00	0.23±0.00	0.23±0.00	0.51±0.00	0.23±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball(2011)	0.20±0.00	0.41±0.00	0.49±0.00	0.21±0.02	0.20±0.00	0.22±0.00	0.22±0.00	0.51±0.00	0.24±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball(2012)	0.19±0.00	0.45±0.00	0.49±0.00	0.22±0.02	0.19±0.00	0.21±0.00	0.21±0.00	0.49±0.00	0.22±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball(2013)	0.20±0.00	0.43±0.00	0.50±0.01	0.24±0.02	0.20±0.00	0.22±0.00	0.22±0.00	0.49±0.00	0.25±0.00	0.23±0.00	nan±nan	0.20±0.00	**0.19±0.00**
Basketball(2014)	0.20±0.00	0.43±0.00	0.52±0.00	0.22±0.02	0.20±0.00	0.22±0.00	0.23±0.00	0.48±0.00	0.25±0.00	0.22±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball finer(1985)	0.19±0.00	0.41±0.00	0.49±0.02	0.36±0.01	0.21±0.00	0.29±0.00	0.29±0.00	0.49±0.00	0.25±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball finer(1986)	0.19±0.00	0.45±0.00	0.50±0.00	0.36±0.02	0.21±0.00	0.29±0.00	0.29±0.00	0.50±0.00	0.27±0.00	0.21±0.00	nan±nan	0.20±0.00	**0.17±0.00**
Basketball finer(1987)	0.20±0.00	0.45±0.00	0.47±0.00	0.35±0.01	0.22±0.00	0.29±0.00	0.30±0.00	0.49±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.01	**0.19±0.00**
Basketball finer(1988)	0.19±0.00	0.45±0.00	0.48±0.00	0.36±0.03	0.21±0.00	0.31±0.00	0.30±0.00	0.49±0.00	0.24±0.00	0.21±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball finer(1989)	0.19±0.00	0.42±0.00	0.46±0.00	0.36±0.01	0.21±0.00	0.28±0.00	0.29±0.00	0.49±0.00	0.25±0.00	0.23±0.00	nan±nan	0.20±0.01	**0.18±0.00**
Basketball finer(1990)	0.20±0.00	0.42±0.00	0.48±0.00	0.36±0.01	0.21±0.00	0.32±0.00	0.29±0.00	0.50±0.00	0.23±0.00	0.21±0.00	nan±nan	0.20±0.01	**0.18±0.00**
Basketball finer(1991)	0.20±0.00	0.46±0.00	0.51±0.00	0.34±0.02	0.21±0.00	0.35±0.00	0.33±0.00	0.49±0.00	0.25±0.00	0.22±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball finer(1992)	0.18±0.00	0.43±0.00	0.47±0.00	0.33±0.01	0.19±0.00	0.31±0.00	0.30±0.00	0.47±0.00	0.24±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball finer(1993)	0.19±0.00	0.42±0.00	0.51±0.00	0.34±0.01	0.20±0.00	0.29±0.00	0.27±0.00	0.49±0.00	0.25±0.00	0.21±0.00	nan±nan	0.20±0.00	**0.17±0.00**
Basketball finer(1994)	0.18±0.00	0.42±0.00	0.50±0.00	0.34±0.02	0.19±0.00	0.31±0.00	0.27±0.00	0.49±0.00	0.22±0.00	0.21±0.00	nan±nan	0.19±0.00	**0.17±0.00**
Basketball finer(1995)	0.20±0.00	0.44±0.00	0.47±0.00	0.34±0.02	0.21±0.00	0.30±0.00	0.28±0.00	0.48±0.00	0.24±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball finer(1996)	0.20±0.00	0.42±0.00	0.49±0.00	0.36±0.02	0.22±0.00	0.30±0.00	0.30±0.00	0.49±0.00	0.27±0.00	0.24±0.00	nan±nan	0.22±0.00	**0.19±0.00**
Basketball finer(1997)	0.21±0.00	0.44±0.00	0.49±0.00	0.35±0.01	0.21±0.00	0.30±0.00	0.29±0.00	0.51±0.00	0.24±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball finer(1998)	0.20±0.00	0.42±0.00	0.48±0.00	0.34±0.02	0.21±0.00	0.29±0.00	0.28±0.00	0.48±0.00	0.24±0.00	0.22±0.00	nan±nan	0.20±0.00	**0.18±0.00**
Basketball finer(1999)	0.20±0.00	0.41±0.00	0.50±0.00	0.34±0.02	0.22±0.00	0.29±0.00	0.28±0.00	0.50±0.00	0.29±0.00	0.24±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball finer(2000)	0.21±0.00	0.44±0.00	0.49±0.00	0.35±0.01	0.23±0.00	0.32±0.00	0.30±0.00	0.48±0.00	0.28±0.00	0.24±0.00	nan±nan	0.22±0.00	**0.19±0.00**
Basketball finer(2001)	0.20±0.00	0.42±0.00	0.51±0.00	0.35±0.01	0.21±0.00	0.31±0.00	0.30±0.00	0.51±0.00	0.27±0.00	0.24±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball finer(2002)	0.22±0.00	0.44±0.00	0.47±0.00	0.36±0.02	0.22±0.00	0.30±0.00	0.28±0.00	0.51±0.00	0.27±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball finer(2003)	0.22±0.00	0.45±0.00	0.50±0.02	0.36±0.02	0.23±0.00	0.29±0.00	0.29±0.00	0.50±0.00	0.26±0.00	0.24±0.00	nan±nan	0.22±0.00	**0.19±0.00**
Basketball finer(2004)	0.19±0.00	0.43±0.01	0.47±0.00	0.35±0.01	0.20±0.00	0.29±0.00	0.28±0.00	0.49±0.00	0.24±0.00	0.22±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball finer(2005)	0.21±0.00	0.46±0.00	0.48±0.00	0.34±0.01	0.22±0.00	0.29±0.00	0.27±0.00	0.50±0.00	0.27±0.00	0.24±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball finer(2006)	0.21±0.00	0.44±0.00	0.49±0.00	0.35±0.02	0.21±0.00	0.30±0.00	0.28±0.00	0.49±0.00	0.27±0.00	0.23±0.00	nan±nan	0.22±0.01	**0.19±0.00**
Basketball finer(2007)	0.21±0.00	0.46±0.00	0.49±0.03	0.35±0.01	0.23±0.00	0.29±0.00	0.27±0.00	0.48±0.00	0.24±0.00	0.23±0.00	nan±nan	0.22±0.00	**0.20±0.00**
Basketball finer(2008)	0.21±0.00	0.43±0.00	0.49±0.00	0.34±0.02	0.22±0.00	0.30±0.00	0.28±0.00	0.49±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.20±0.00**
Basketball finer(2009)	0.21±0.00	0.43±0.00	0.50±0.01	0.34±0.01	0.22±0.00	0.29±0.00	0.28±0.00	0.52±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball finer(2010)	0.20±0.00	0.42±0.00	0.49±0.00	0.33±0.02	0.21±0.00	0.28±0.00	0.28±0.00	0.49±0.00	0.24±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.19±0.00**
Basketball finer(2011)	0.21±0.00	0.41±0.00	0.49±0.00	0.34±0.01	0.22±0.00	0.28±0.00	0.28±0.00	0.50±0.00	0.24±0.00	0.22±0.00	nan±nan	0.22±0.00	**0.19±0.00**
Basketball finer(2012)	0.20±0.00	0.42±0.00	0.49±0.00	0.34±0.02	0.21±0.00	0.29±0.00	0.28±0.00	0.48±0.00	0.23±0.00	0.22±0.00	nan±nan	0.21±0.00	**0.18±0.00**
Basketball finer(2013)	0.21±0.00	0.43±0.00	0.52±0.00	0.34±0.02	0.22±0.00	0.29±0.00	0.28±0.00	0.49±0.00	0.25±0.00	0.23±0.00	nan±nan	0.21±0.00	**0.20±0.00**
Basketball finer(2014)	0.21±0.00	0.44±0.05	0.52±0.00	0.36±0.02	0.22±0.00	0.30±0.00	0.28±0.00	0.49±0.00	0.26±0.00	0.22±0.00	nan±nan	0.22±0.00	**0.20±0.00**
Football(2009)	**0.15±0.00**	0.36±0.11	0.23±0.00	0.17±0.02	0.20±0.00	0.23±0.00	0.26±0.00	0.45±0.02	0.20±0.00	0.16±0.00	0.39±0.04	0.19±0.01	**0.15±0.00**
Football(2010)	0.30±0.00	0.45±0.04	0.39±0.00	0.30±0.01	0.32±0.00	0.34±0.00	0.34±0.00	0.46±0.03	0.31±0.00	**0.29±0.00**	0.49±0.05	0.30±0.01	**0.29±0.00**
Football(2011)	**0.20±0.00**	0.39±0.09	0.24±0.00	0.22±0.01	0.22±0.00	0.25±0.00	0.25±0.00	0.40±0.00	0.23±0.00	0.21±0.00	0.45±0.05	0.22±0.01	**0.20±0.00**
Football(2012)	0.24±0.00	0.38±0.01	0.35±0.00	0.25±0.01	0.22±0.00	0.24±0.00	0.25±0.00	0.41±0.00	0.25±0.00	0.22±0.00	0.43±0.04	**0.20±0.02**	0.21±0.00
Football(2013)	0.19±0.00	0.41±0.03	0.19±0.00	0.19±0.01	0.17±0.00	0.23±0.00	0.22±0.00	0.48±0.02	0.22±0.00	0.19±0.00	0.40±0.03	0.16±0.01	**0.14±0.00**
Football(2014)	0.28±0.00	0.45±0.05	0.39±0.00	0.29±0.00	0.28±0.00	0.32±0.00	0.30±0.00	0.46±0.03	0.28±0.00	0.27±0.00	0.43±0.05	**0.24±0.01**	0.26±0.01
Football finer(2009)	0.18±0.00	0.45±0.00	0.23±0.00	0.20±0.03	0.18±0.00	0.26±0.00	0.27±0.00	0.47±0.01	0.20±0.00	**0.16±0.00**	0.48±0.04	0.19±0.05	**0.16±0.01**
Football finer(2010)	0.32±0.00	0.41±0.01	0.36±0.00	0.33±0.02	0.32±0.00	0.34±0.00	0.37±0.00	0.49±0.00	0.31±0.00	**0.29±0.00**	0.53±0.00	0.31±0.02	**0.29±0.00**
Football finer(2011)	0.22±0.00	0.40±0.01	0.27±0.00	0.23±0.01	0.23±0.00	0.25±0.00	0.28±0.00	0.47±0.00	0.23±0.00	**0.21±0.00**	0.53±0.00	0.23±0.04	**0.21±0.01**
Football finer(2012)	0.24±0.00	0.44±0.04	0.27±0.00	0.25±0.01	0.23±0.00	0.27±0.00	0.30±0.00	0.51±0.01	0.25±0.00	0.22±0.00	0.54±0.00	0.24±0.02	**0.21±0.01**
Football finer(2013)	0.22±0.00	0.39±0.00	0.22±0.00	0.21±0.02	0.18±0.00	0.25±0.00	0.27±0.00	0.50±0.00	0.22±0.00	0.19±0.00	0.49±0.04	0.18±0.05	**0.14±0.00**
Football finer(2014)	0.29±0.00	0.44±0.01	0.39±0.00	0.30±0.01	**0.26±0.00**	0.31±0.00	0.31±0.00	0.42±0.00	0.28±0.00	0.27±0.00	0.52±0.04	0.27±0.01	0.27±0.02

Table D.4: Result table on $\mathcal{L}_{\text{upset, ratio}}$ for individual directed graphs as input, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue . MVR could not generate scores, so we omit the results.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	GNNRank-N	GNNRank-P
HeadToHead	0.86±0.00	0.97±0.00	0.97±0.00	0.71±0.01	0.89±0.00	0.98±0.04	0.83±0.00	1.19±0.09	1.14±0.00	0.95±0.00	**0.68±0.00**	0.94±0.00
Finance	**0.26±0.00**	0.27±0.00	0.27±0.00	0.28±0.00	0.64±0.00	0.27±0.00	**0.26±0.00**	0.59±0.00	**0.26±0.00**	**0.26±0.00**	**0.26±0.00**	0.27±0.00
Animal	0.42±0.00	0.80±0.02	0.88±0.00	**0.22±0.00**	0.41±0.00	0.43±0.14	0.50±0.00	1.34±0.00	0.62±0.00	0.42±0.00	0.24±0.03	0.66±0.00
Faculty: Business	0.67±0.00	0.98±0.00	0.99±0.00	0.34±0.00	0.52±0.00	0.74±0.25	0.47±0.00	1.40±0.13	0.64±0.00	0.62±0.00	**0.31±0.00**	0.89±0.00
Faculty: CS	0.65±0.00	0.99±0.00	1.00±0.00	0.28±0.00	0.51±0.00	0.63±0.24	0.42±0.00	1.33±0.16	0.73±0.00	0.72±0.00	**0.26±0.02**	0.86±0.00
Faculty: History	0.61±0.00	0.99±0.00	1.00±0.00	0.23±0.00	0.49±0.00	0.60±0.28	0.37±0.00	1.35±0.11	0.68±0.00	0.72±0.00	**0.21±0.00**	0.84±0.02
Basketball(1985)	0.67±0.00	0.86±0.00	0.86±0.00	0.54±0.04	0.62±0.00	0.67±0.17	0.55±0.00	1.32±0.00	0.67±0.00	0.63±0.00	**0.46±0.01**	0.82±0.00
Basketball(1986)	0.69±0.00	0.85±0.00	0.86±0.00	0.54±0.01	0.61±0.00	0.73±0.16	0.54±0.00	1.30±0.00	0.75±0.00	0.63±0.00	**0.42±0.00**	0.81±0.00
Basketball(1987)	0.71±0.00	0.86±0.00	0.86±0.00	0.57±0.01	0.65±0.00	0.67±0.15	0.56±0.00	1.37±0.00	0.70±0.00	0.68±0.00	**0.48±0.01**	0.82±0.00
Basketball(1988)	0.65±0.00	0.86±0.00	0.86±0.00	0.53±0.01	0.59±0.00	0.70±0.17	0.54±0.00	1.33±0.00	0.70±0.00	0.63±0.00	**0.45±0.01**	0.82±0.00
Basketball(1989)	0.71±0.00	0.87±0.00	0.87±0.00	0.55±0.01	0.63±0.00	0.77±0.15	0.55±0.00	1.34±0.00	0.70±0.00	0.66±0.00	**0.46±0.01**	0.83±0.00
Basketball(1990)	0.67±0.00	0.86±0.00	0.87±0.00	0.51±0.05	0.62±0.00	0.74±0.16	0.55±0.00	1.35±0.00	0.65±0.00	0.62±0.00	**0.44±0.01**	0.82±0.00
Basketball(1991)	0.67±0.00	0.86±0.00	0.87±0.00	0.53±0.03	0.64±0.00	0.74±0.16	0.55±0.00	1.34±0.00	0.68±0.00	0.63±0.00	**0.45±0.00**	0.82±0.00
Basketball(1992)	0.69±0.00	0.87±0.00	0.87±0.00	0.50±0.06	0.59±0.00	0.73±0.17	0.54±0.00	1.35±0.00	0.70±0.00	0.61±0.00	**0.43±0.00**	0.83±0.00
Basketball(1993)	0.67±0.00	0.86±0.00	0.86±0.00	0.51±0.03	0.60±0.00	0.79±0.11	0.52±0.00	1.28±0.00	0.68±0.00	0.62±0.00	**0.44±0.01**	0.82±0.00
Basketball(1994)	0.56±0.00	0.86±0.00	0.86±0.00	0.43±0.05	0.59±0.00	0.64±0.19	0.53±0.00	1.35±0.00	0.61±0.00	0.60±0.00	**0.42±0.00**	0.82±0.00
Basketball(1995)	0.67±0.00	0.85±0.00	0.86±0.00	0.47±0.06	0.59±0.00	0.58±0.16	0.53±0.00	1.36±0.00	0.64±0.00	0.61±0.00	**0.44±0.00**	0.81±0.00
Basketball(1996)	0.68±0.00	0.86±0.00	0.87±0.00	0.55±0.03	0.65±0.00	0.57±0.13	0.56±0.00	1.35±0.00	0.74±0.00	0.68±0.00	**0.48±0.00**	0.82±0.00
Basketball(1997)	0.71±0.00	0.86±0.00	0.86±0.00	0.56±0.00	0.63±0.00	0.71±0.16	0.56±0.00	1.30±0.00	0.69±0.00	0.68±0.00	**0.48±0.00**	0.82±0.00
Basketball(1998)	0.70±0.00	0.88±0.00	0.88±0.00	0.56±0.01	0.64±0.00	0.71±0.17	0.56±0.00	1.36±0.00	0.70±0.00	0.67±0.00	**0.46±0.00**	0.84±0.00
Basketball(1999)	0.60±0.00	0.87±0.00	0.88±0.00	0.55±0.03	0.65±0.00	0.68±0.17	0.57±0.00	1.32±0.00	0.80±0.00	0.66±0.00	**0.48±0.00**	0.83±0.00
Basketball(2000)	0.70±0.00	0.90±0.00	0.90±0.00	0.58±0.01	0.65±0.00	0.77±0.16	0.59±0.00	1.40±0.01	0.77±0.00	0.67±0.00	**0.50±0.00**	0.86±0.00
Basketball(2001)	0.70±0.00	0.89±0.00	0.89±0.00	0.55±0.04	0.64±0.00	0.72±0.18	0.56±0.00	1.40±0.00	0.74±0.00	0.67±0.00	**0.47±0.00**	0.85±0.00
Basketball(2002)	0.70±0.00	0.88±0.00	0.88±0.00	0.56±0.03	0.66±0.00	0.65±0.17	0.58±0.00	1.33±0.00	0.77±0.00	0.68±0.00	**0.48±0.01**	0.84±0.00
Basketball(2003)	0.72±0.00	0.88±0.00	0.89±0.00	0.56±0.04	0.68±0.00	0.67±0.16	0.59±0.00	1.40±0.00	0.73±0.00	0.71±0.00	**0.51±0.00**	0.85±0.00
Basketball(2004)	0.68±0.00	0.88±0.00	0.89±0.00	0.54±0.04	0.64±0.00	0.64±0.18	0.56±0.00	1.35±0.00	0.67±0.00	0.64±0.00	**0.46±0.00**	0.85±0.00
Basketball(2005)	0.73±0.00	0.89±0.00	0.89±0.00	0.57±0.01	0.65±0.00	0.69±0.18	0.58±0.00	1.29±0.01	0.73±0.00	0.68±0.00	**0.49±0.01**	0.85±0.00
Basketball(2006)	0.73±0.00	0.89±0.00	0.90±0.00	0.50±0.05	0.65±0.00	0.69±0.18	0.58±0.00	1.36±0.00	0.76±0.00	0.69±0.00	**0.49±0.01**	0.85±0.00
Basketball(2007)	0.72±0.00	0.90±0.00	0.90±0.00	0.57±0.00	0.65±0.00	0.62±0.16	0.57±0.00	1.41±0.00	0.67±0.00	0.67±0.00	**0.49±0.00**	0.86±0.00
Basketball(2008)	0.74±0.00	0.90±0.00	0.90±0.00	0.55±0.04	0.65±0.00	0.73±0.17	0.58±0.00	1.33±0.01	0.70±0.00	0.67±0.00	**0.51±0.01**	0.86±0.00
Basketball(2009)	0.70±0.00	0.90±0.00	0.90±0.00	0.57±0.01	0.65±0.00	0.65±0.18	0.56±0.00	1.24±0.00	0.69±0.00	0.66±0.00	**0.47±0.00**	0.86±0.00
Basketball(2010)	0.74±0.00	0.91±0.00	0.91±0.00	0.52±0.06	0.66±0.00	0.71±0.18	0.58±0.00	1.35±0.00	0.69±0.00	0.67±0.00	**0.49±0.00**	0.87±0.00
Basketball(2011)	0.73±0.00	0.90±0.00	0.90±0.00	0.52±0.05	0.66±0.00	0.63±0.16	0.58±0.00	1.34±0.00	0.70±0.00	0.68±0.00	**0.49±0.00**	0.86±0.00
Basketball(2012)	0.73±0.00	0.90±0.00	0.91±0.00	0.54±0.05	0.64±0.00	0.73±0.19	0.56±0.00	1.42±0.00	0.66±0.00	0.64±0.00	**0.48±0.01**	0.86±0.00
Basketball(2013)	0.74±0.00	0.90±0.00	0.91±0.00	0.58±0.03	0.67±0.00	0.70±0.17	0.59±0.00	1.34±0.00	0.73±0.00	0.70±0.00	**0.51±0.01**	0.86±0.00
Basketball(2014)	0.72±0.00	0.90±0.00	0.90±0.00	0.52±0.06	0.66±0.00	0.65±0.18	0.58±0.00	1.41±0.00	0.72±0.00	0.67±0.00	**0.49±0.01**	0.86±0.00
Basketball finer(1985)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.47±0.00	0.01±0.00	0.01±0.00	0.57±0.00	0.46±0.00	0.46±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1986)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.49±0.00	0.01±0.00	0.01±0.00	0.60±0.00	0.48±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1987)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.51±0.00	0.01±0.00	0.01±0.00	0.58±0.00	0.46±0.00	0.48±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1988)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.52±0.00	0.01±0.00	0.01±0.00	0.57±0.00	0.56±0.00	0.52±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1989)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.49±0.00	0.01±0.00	0.01±0.00	0.56±0.00	0.49±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1990)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.49±0.00	0.01±0.00	0.01±0.00	0.55±0.00	0.48±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1991)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.49±0.00	0.01±0.00	0.01±0.00	0.54±0.00	0.49±0.00	0.47±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1992)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.49±0.00	0.01±0.00	0.01±0.00	0.55±0.00	0.44±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1993)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.51±0.00	0.01±0.00	0.01±0.00	0.58±0.00	0.47±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1994)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.49±0.00	0.01±0.00	0.01±0.00	0.58±0.00	0.46±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1995)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.50±0.00	0.01±0.00	0.01±0.00	0.55±0.00	0.47±0.00	0.49±0.00	**0.00±0.00**	0.01±0.00
Basketball finer(1996)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.48±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.45±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1997)	0.04±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.48±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1998)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.47±0.00	0.45±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1999)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.48±0.00	0.46±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2000)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.55±0.00	0.50±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2001)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.59±0.00	0.51±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2002)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.51±0.00	0.50±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2003)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.47±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2004)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.45±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2005)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.60±0.00	0.43±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2006)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.46±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2007)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.53±0.00	**0.01±0.00**	**0.01±0.00**	0.53±0.00	0.49±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2008)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.59±0.00	0.47±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2009)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.50±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2010)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.48±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2011)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.59±0.00	0.46±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2012)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.48±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2013)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.46±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2014)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.53±0.00	0.01±0.00	0.01±0.00	0.56±0.00	0.48±0.00	0.47±0.00	**0.00±0.00**	0.01±0.00
Football(2009)	0.64±0.00	0.79±0.03	0.73±0.00	0.48±0.02	0.55±0.00	0.63±0.09	0.60±0.00	1.16±0.05	0.49±0.00	0.49±0.00	**0.46±0.01**	0.69±0.00
Football(2010)	0.73±0.00	0.80±0.04	0.81±0.01	0.66±0.02	0.76±0.00	0.75±0.04	0.74±0.00	1.25±0.01	0.74±0.00	0.73±0.00	**0.65±0.00**	0.73±0.01
Football(2011)	0.66±0.00	0.79±0.01	0.70±0.00	**0.53±0.02**	0.70±0.00	0.65±0.08	0.60±0.00	1.17±0.00	0.59±0.00	0.58±0.00	**0.53±0.01**	0.69±0.00
Football(2012)	0.58±0.00	0.81±0.00	0.83±0.00	0.53±0.03	0.60±0.00	0.68±0.09	0.63±0.00	1.18±0.00	0.58±0.00	0.58±0.00	**0.51±0.01**	0.71±0.00
Football(2013)	0.66±0.00	0.81±0.00	0.73±0.00	0.48±0.04	0.59±0.00	0.64±0.10	0.62±0.00	0.94±0.00	0.56±0.00	0.53±0.00	**0.46±0.01**	0.71±0.00
Football(2014)	0.85±0.00	0.95±0.01	0.97±0.01	0.74±0.01	0.98±0.00	0.83±0.07	0.81±0.00	1.49±0.00	0.84±0.00	0.82±0.00	**0.69±0.01**	0.85±0.00
Football finer(2009)	0.22±0.00	0.27±0.00	0.23±0.00	**0.17±0.01**	0.38±0.00	0.23±0.03	0.22±0.00	0.73±0.04	0.21±0.00	0.21±0.00	**0.17±0.00**	0.21±0.00
Football finer(2010)	0.20±0.00	0.23±0.00	0.21±0.00	0.18±0.01	0.56±0.00	0.22±0.01	0.22±0.00	0.67±0.00	0.24±0.00	0.22±0.00	**0.17±0.00**	0.18±0.00
Football finer(2011)	0.19±0.00	0.25±0.00	0.23±0.00	**0.17±0.00**	0.49±0.00	0.22±0.02	0.21±0.00	0.58±0.00	0.22±0.00	0.22±0.00	**0.17±0.00**	0.19±0.00
Football finer(2012)	0.19±0.00	0.23±0.04	0.21±0.01	0.16±0.00	0.48±0.00	0.19±0.02	0.19±0.00	0.66±0.04	0.18±0.00	0.17±0.00	**0.15±0.00**	0.17±0.00
Football finer(2013)	0.24±0.00	0.30±0.00	0.29±0.01	0.21±0.01	0.43±0.00	0.27±0.03	0.26±0.00	0.52±0.00	0.24±0.00	0.22±0.00	**0.19±0.04**	0.24±0.01
Football finer(2014)	0.36±0.00	0.45±0.00	0.45±0.00	0.35±0.00	0.62±0.00	0.40±0.03	0.39±0.00	0.98±0.00	0.37±0.00	0.37±0.00	**0.34±0.00**	0.38±0.00

Table D.5: Performance on Kendall Tau (top half) based on the lowest $\mathcal{L}_{\text{upset, naive}}$ (bottom half for corresponding values) on ERO models, averaged over 10 runs with one standard deviation. "avg" for time series first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best is marked in **bold red** while the second best is in underline blue . As MVR does not generate results after one week, we leave it out here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	GNNRank-N	GNNRank-P
ERO(p=0.05, style=uniform,η=0)	0.86±0.00	0.06±0.00	0.03±0.00	0.79±0.05	0.80±0.00	**0.91±0.00**	0.87±0.00	0.02±0.09	0.78±0.00	0.83±0.00	0.88±0.00	0.86±0.00
ERO(p=0.05, style=gamma,η=0)	0.81±0.00	0.06±0.00	0.00±0.00	0.87±0.01	0.81±0.00	**0.92±0.00**	0.90±0.00	-0.00±0.01	0.58±0.00	0.79±0.00	0.87±0.00	0.84±0.01
ERO(p=0.05, style=uniform,η=0.1)	0.75±0.00	0.04±0.00	0.03±0.00	0.70±0.01	0.77±0.00	0.56±0.00	0.58±0.00	0.01±0.05	0.74±0.00	0.77±0.00	0.76±0.01	**0.79±0.01**
ERO(p=0.05, style=gamma,η=0.1)	0.69±0.00	0.04±0.00	0.01±0.00	0.68±0.00	0.78±0.00	0.58±0.00	0.60±0.00	0.06±0.00	0.52±0.00	0.72±0.00	0.61±0.01	**0.81±0.01**
ERO(p=0.05, style=uniform,η=0.2)	0.68±0.00	0.07±0.00	0.03±0.00	0.64±0.01	0.73±0.00	0.50±0.00	0.48±0.00	0.02±0.04	0.67±0.00	0.70±0.00	0.68±0.01	**0.76±0.01**
ERO(p=0.05, style=gamma,η=0.2)	0.61±0.00	0.01±0.00	-0.01±0.00	0.61±0.00	0.74±0.00	0.52±0.00	0.51±0.00	-0.01±0.01	0.45±0.00	0.64±0.00	0.52±0.01	**0.77±0.00**
ERO(p=0.05, style=uniform,η=0.3)	0.61±0.00	0.05±0.00	0.01±0.00	0.59±0.01	0.68±0.00	0.44±0.00	0.41±0.00	0.05±0.00	0.60±0.00	0.62±0.00	0.62±0.00	**0.70±0.02**
ERO(p=0.05, style=gamma,η=0.3)	0.56±0.00	0.11±0.00	-0.00±0.00	0.56±0.01	0.71±0.00	0.46±0.00	0.44±0.00	0.11±0.00	0.34±0.00	0.56±0.00	0.43±0.03	**0.72±0.00**
ERO(p=0.05, style=uniform,η=0.4)	0.55±0.00	0.08±0.00	0.01±0.00	0.54±0.04	**0.63±0.00**	0.40±0.00	0.35±0.00	0.02±0.00	0.51±0.00	0.54±0.00	0.52±0.00	0.62±0.02
ERO(p=0.05, style=gamma,η=0.4)	0.51±0.00	0.08±0.00	-0.00±0.00	0.52±0.00	0.65±0.00	0.43±0.00	0.43±0.00	0.09±0.01	0.23±0.00	0.44±0.00	0.38±0.08	**0.66±0.01**
ERO(p=0.05, style=uniform,η=0.5)	0.47±0.00	0.08±0.00	0.01±0.00	0.47±0.05	**0.57±0.00**	0.37±0.00	0.32±0.00	0.04±0.00	0.25±0.00	0.36±0.00	0.42±0.02	0.56±0.00
ERO(p=0.05, style=gamma,η=0.5)	0.44±0.00	0.07±0.00	0.01±0.00	0.45±0.01	0.58±0.00	0.37±0.00	0.37±0.00	0.04±0.00	0.22±0.00	0.23±0.00	0.22±0.01	**0.59±0.02**
ERO(p=0.05, style=uniform,η=0.6)	0.39±0.00	0.03±0.00	-0.00±0.00	0.39±0.06	**0.48±0.00**	0.31±0.00	0.27±0.00	-0.03±0.00	0.15±0.00	0.15±0.00	0.33±0.00	0.46±0.00
ERO(p=0.05, style=gamma,η=0.6)	0.36±0.00	0.03±0.00	0.01±0.00	0.37±0.01	0.49±0.00	0.31±0.00	0.33±0.00	0.01±0.00	0.14±0.00	0.09±0.00	0.24±0.03	**0.50±0.01**
ERO(p=0.05, style=uniform,η=0.7)	0.31±0.00	0.08±0.00	-0.01±0.00	0.31±0.06	0.38±0.00	0.26±0.00	0.23±0.00	0.04±0.00	0.07±0.00	0.06±0.00	0.23±0.01	**0.39±0.03**
ERO(p=0.05, style=gamma,η=0.7)	0.29±0.00	-0.00±0.00	0.02±0.00	0.29±0.01	**0.39±0.00**	0.25±0.00	0.28±0.00	-0.00±0.00	0.02±0.00	0.03±0.00	0.05±0.02	**0.39±0.02**
ERO(p=0.05, style=uniform,η=0.8)	0.21±0.00	0.02±0.00	-0.02±0.00	0.22±0.07	0.25±0.00	0.20±0.00	0.15±0.00	-0.03±0.00	0.00±0.00	0.01±0.00	0.11±0.00	**0.27±0.04**
ERO(p=0.05, style=gamma,η=0.8)	0.18±0.00	0.07±0.00	0.00±0.00	0.18±0.01	0.24±0.00	0.16±0.00	0.19±0.00	-0.02±0.02	0.03±0.00	-0.03±0.00	0.05±0.02	**0.27±0.03**
ERO(p=1, style=uniform,η=0)	**1.00±0.00**	0.08±0.00	**1.00±0.00**	0.85±0.05	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	-0.00±0.06	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**
ERO(p=1, style=gamma,η=0)	**1.00±0.00**	0.08±0.00	**1.00±0.00**	0.95±0.00	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	0.01±0.09	**1.00±0.00**	0.99±0.00	**1.00±0.00**	**1.00±0.00**
ERO(p=1, style=uniform,η=0.1)	0.94±0.00	0.05±0.00	**0.98±0.00**	0.85±0.04	**0.98±0.00**	0.86±0.00	0.63±0.00	0.52±0.00	0.97±0.00	0.94±0.00	0.93±0.01	**0.98±0.00**
ERO(p=1, style=gamma,η=0.1)	0.88±0.00	0.10±0.00	**0.98±0.00**	0.85±0.00	**0.98±0.00**	0.72±0.00	0.57±0.00	0.38±0.00	0.92±0.00	0.90±0.00	0.89±0.01	**0.98±0.00**
ERO(p=1, style=uniform,η=0.2)	0.91±0.00	0.07±0.00	**0.97±0.00**	0.84±0.03	0.96±0.00	0.85±0.00	0.69±0.00	0.33±0.09	0.95±0.00	0.93±0.00	0.91±0.00	**0.97±0.00**
ERO(p=1, style=gamma,η=0.2)	0.85±0.00	0.05±0.00	**0.97±0.00**	0.83±0.00	0.96±0.00	0.71±0.00	0.60±0.00	0.28±0.00	0.89±0.00	0.87±0.00	0.86±0.00	**0.97±0.00**
ERO(p=1, style=uniform,η=0.3)	0.89±0.00	0.09±0.00	**0.96±0.00**	0.84±0.03	0.94±0.00	0.83±0.00	0.71±0.00	0.29±0.00	0.94±0.00	0.91±0.00	0.90±0.01	**0.96±0.00**
ERO(p=1, style=gamma,η=0.3)	0.82±0.00	0.06±0.00	**0.95±0.00**	0.80±0.00	**0.95±0.00**	0.70±0.00	0.64±0.00	0.32±0.00	0.85±0.00	0.84±0.00	0.85±0.00	**0.95±0.00**
ERO(p=1, style=uniform,η=0.4)	0.88±0.00	0.06±0.00	**0.94±0.00**	0.83±0.03	0.93±0.00	0.83±0.00	0.73±0.00	0.11±0.00	0.92±0.00	0.90±0.00	0.91±0.01	**0.94±0.00**
ERO(p=1, style=gamma,η=0.4)	0.79±0.00	0.04±0.00	**0.94±0.00**	0.78±0.00	0.93±0.00	0.69±0.00	0.65±0.00	0.14±0.00	0.83±0.00	0.82±0.00	0.83±0.01	**0.94±0.00**
ERO(p=1, style=uniform,η=0.5)	0.85±0.00	0.07±0.00	**0.92±0.00**	0.81±0.03	0.91±0.00	0.80±0.00	0.73±0.00	0.24±0.00	0.89±0.00	0.87±0.00	0.90±0.01	**0.92±0.00**
ERO(p=1, style=gamma,η=0.5)	0.76±0.00	0.03±0.00	**0.92±0.00**	0.74±0.01	0.90±0.00	0.68±0.00	0.66±0.00	0.16±0.01	0.79±0.00	0.78±0.00	0.81±0.01	**0.92±0.00**
ERO(p=1, style=uniform,η=0.6)	0.83±0.00	0.08±0.00	**0.89±0.00**	0.78±0.04	0.88±0.00	0.78±0.00	0.73±0.00	-0.00±0.00	0.86±0.00	0.85±0.00	0.87±0.01	**0.89±0.00**
ERO(p=1, style=gamma,η=0.6)	0.72±0.00	0.09±0.00	**0.89±0.00**	0.67±0.01	0.88±0.00	0.65±0.00	0.64±0.00	0.05±0.02	0.74±0.00	0.73±0.00	0.77±0.00	**0.89±0.00**
ERO(p=1, style=uniform,η=0.7)	0.77±0.00	0.08±0.00	0.80±0.00	0.72±0.05	**0.83±0.00**	0.74±0.00	0.70±0.00	0.18±0.00	0.80±0.00	0.79±0.00	**0.83±0.01**	**0.83±0.00**
ERO(p=1, style=gamma,η=0.7)	0.67±0.00	0.02±0.00	0.75±0.00	0.61±0.01	**0.83±0.00**	0.62±0.00	0.61±0.00	0.09±0.00	0.67±0.00	0.67±0.00	0.71±0.00	**0.83±0.00**
ERO(p=1, style=uniform,η=0.8)	0.68±0.00	0.06±0.00	0.11±0.00	0.62±0.05	**0.75±0.00**	0.65±0.00	0.62±0.00	0.15±0.00	0.67±0.00	0.66±0.00	0.70±0.01	0.74±0.01
ERO(p=1, style=gamma,η=0.8)	0.57±0.00	0.06±0.00	0.04±0.00	0.51±0.01	**0.75±0.00**	0.55±0.00	0.55±0.00	0.04±0.00	0.54±0.00	0.54±0.00	0.56±0.00	0.74±0.00
ERO(p=0.05, style=uniform,η=0)	0.06±0.00	0.47±0.00	0.47±0.00	0.09±0.03	0.06±0.00	**0.03±0.00**	0.05±0.00	0.49±0.03	0.10±0.00	0.07±0.00	0.04±0.00	0.05±0.00
ERO(p=0.05, style=gamma,η=0)	0.08±0.00	0.48±0.00	0.51±0.00	0.04±0.00	0.05±0.00	**0.02±0.00**	0.03±0.00	0.50±0.00	0.20±0.00	0.09±0.00	0.04±0.00	0.04±0.01
ERO(p=0.05, style=uniform,η=0.1)	0.13±0.00	0.46±0.00	0.47±0.00	0.15±0.01	0.11±0.00	0.22±0.00	0.21±0.00	0.50±0.01	0.16±0.00	0.14±0.00	0.13±0.01	**0.09±0.00**
ERO(p=0.05, style=gamma,η=0.1)	0.16±0.00	0.49±0.00	0.51±0.00	0.16±0.00	0.11±0.00	0.22±0.00	0.21±0.00	0.47±0.00	0.24±0.00	0.15±0.00	0.19±0.00	**0.09±0.00**
ERO(p=0.05, style=uniform,η=0.2)	0.18±0.00	0.48±0.00	0.50±0.00	0.20±0.01	0.15±0.00	0.25±0.00	0.27±0.00	0.49±0.01	0.21±0.00	0.19±0.00	0.19±0.00	**0.13±0.00**
ERO(p=0.05, style=gamma,η=0.2)	0.21±0.00	0.49±0.00	0.52±0.00	0.21±0.00	0.15±0.00	0.28±0.00	0.27±0.00	0.51±0.01	0.30±0.00	0.21±0.00	0.24±0.00	**0.14±0.00**
ERO(p=0.05, style=uniform,η=0.3)	0.22±0.00	0.46±0.00	0.51±0.00	0.23±0.01	0.20±0.00	0.28±0.00	0.31±0.00	0.47±0.00	0.26±0.00	0.24±0.00	0.23±0.00	**0.18±0.01**
ERO(p=0.05, style=gamma,η=0.3)	0.25±0.00	0.48±0.00	0.51±0.00	0.26±0.00	0.20±0.00	0.32±0.00	0.33±0.00	0.46±0.00	0.37±0.00	0.26±0.00	0.31±0.01	**0.19±0.00**
ERO(p=0.05, style=uniform,η=0.4)	0.26±0.00	0.48±0.00	0.51±0.00	0.27±0.01	**0.23±0.00**	0.31±0.00	0.34±0.00	0.50±0.00	0.31±0.00	0.28±0.00	0.28±0.00	**0.23±0.01**
ERO(p=0.05, style=gamma,η=0.4)	0.29±0.00	0.47±0.00	0.51±0.00	0.28±0.00	0.24±0.00	0.34±0.00	0.34±0.00	0.48±0.01	0.43±0.00	0.32±0.00	0.34±0.04	**0.23±0.01**
ERO(p=0.05, style=uniform,η=0.5)	0.29±0.00	0.48±0.00	0.51±0.00	0.30±0.01	**0.27±0.00**	0.34±0.00	0.36±0.00	0.49±0.00	0.40±0.00	0.35±0.00	0.32±0.00	**0.27±0.00**
ERO(p=0.05, style=gamma,η=0.5)	0.32±0.00	0.48±0.00	0.49±0.00	0.33±0.00	0.28±0.00	0.37±0.00	0.37±0.00	0.48±0.00	0.43±0.00	0.43±0.00	0.38±0.01	**0.27±0.01**
ERO(p=0.05, style=uniform,η=0.6)	0.32±0.00	0.47±0.00	0.51±0.00	0.33±0.01	0.31±0.00	0.37±0.00	0.39±0.00	0.51±0.00	0.44±0.00	0.42±0.00	0.35±0.00	**0.30±0.00**
ERO(p=0.05, style=gamma,η=0.6)	0.35±0.00	0.48±0.00	0.50±0.00	0.35±0.00	0.32±0.00	0.40±0.00	0.40±0.00	0.48±0.00	0.47±0.00	0.46±0.00	0.40±0.01	**0.30±0.01**
ERO(p=0.05, style=uniform,η=0.7)	0.34±0.00	0.48±0.00	0.49±0.00	0.35±0.01	0.34±0.00	0.39±0.00	0.40±0.00	0.49±0.00	0.48±0.00	0.47±0.00	0.38±0.01	**0.32±0.00**
ERO(p=0.05, style=gamma,η=0.7)	0.36±0.00	0.49±0.00	0.50±0.00	0.37±0.01	0.35±0.00	0.42±0.00	0.40±0.00	0.49±0.00	0.49±0.00	0.49±0.00	0.43±0.01	**0.33±0.01**
ERO(p=0.05, style=uniform,η=0.8)	0.36±0.00	0.48±0.00	0.49±0.00	0.37±0.00	0.36±0.00	0.40±0.00	0.41±0.00	0.51±0.00	0.50±0.00	0.49±0.00	0.42±0.00	**0.34±0.00**
ERO(p=0.05, style=gamma,η=0.8)	0.38±0.00	0.48±0.00	0.50±0.00	0.39±0.01	0.36±0.00	0.42±0.00	0.41±0.00	0.50±0.01	0.49±0.00	0.50±0.00	0.43±0.02	**0.34±0.01**
ERO(p=1, style=uniform,η=0)	**0.00±0.00**	0.46±0.00	**0.00±0.00**	0.07±0.02	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	0.50±0.03	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
ERO(p=1, style=gamma,η=0)	**0.00±0.00**	0.46±0.00	**0.00±0.00**	0.02±0.00	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	0.50±0.05	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
ERO(p=1, style=uniform,η=0.1)	0.08±0.00	0.48±0.00	**0.06±0.00**	0.12±0.02	**0.06±0.00**	0.11±0.00	0.21±0.00	0.27±0.00	**0.06±0.00**	0.08±0.00	0.08±0.00	**0.06±0.00**
ERO(p=1, style=gamma,η=0.1)	0.10±0.00	0.46±0.00	**0.06±0.00**	0.11±0.00	**0.06±0.00**	0.18±0.00	0.24±0.00	0.33±0.00	0.09±0.00	0.09±0.00	0.10±0.00	**0.06±0.00**
ERO(p=1, style=uniform,η=0.2)	0.13±0.00	0.47±0.00	**0.11±0.00**	0.16±0.01	0.12±0.00	0.16±0.00	0.22±0.00	0.37±0.04	0.12±0.00	0.13±0.00	0.14±0.00	**0.11±0.00**
ERO(p=1, style=gamma,η=0.2)	0.16±0.00	0.48±0.00	**0.11±0.00**	0.17±0.00	0.12±0.00	0.21±0.00	0.26±0.00	0.39±0.00	0.14±0.00	0.15±0.00	0.15±0.00	**0.11±0.00**
ERO(p=1, style=uniform,η=0.3)	0.19±0.00	0.47±0.00	**0.16±0.00**	0.21±0.01	0.17±0.00	0.21±0.00	0.25±0.00	0.40±0.00	0.17±0.00	0.18±0.00	0.18±0.00	**0.16±0.00**
ERO(p=1, style=gamma,η=0.3)	0.21±0.00	0.48±0.00	**0.16±0.00**	0.21±0.00	0.17±0.00	0.25±0.00	0.27±0.00	0.39±0.00	0.20±0.00	0.20±0.00	0.20±0.00	**0.16±0.00**
ERO(p=1, style=uniform,η=0.4)	0.23±0.00	0.48±0.00	**0.21±0.00**	0.25±0.01	0.22±0.00	0.25±0.00	0.28±0.00	0.47±0.00	0.22±0.00	0.23±0.00	0.23±0.00	**0.21±0.00**
ERO(p=1, style=gamma,η=0.4)	0.26±0.00	0.49±0.00	**0.22±0.00**	0.26±0.00	**0.22±0.00**	0.29±0.00	0.30±0.00	0.46±0.00	0.25±0.00	0.25±0.00	0.25±0.00	**0.22±0.00**
ERO(p=1, style=uniform,η=0.5)	0.28±0.00	0.49±0.00	**0.27±0.00**	0.30±0.01	**0.27±0.00**	0.30±0.00	0.31±0.00	0.44±0.00	**0.27±0.00**	0.28±0.00	**0.27±0.00**	**0.27±0.00**
ERO(p=1, style=gamma,η=0.5)	0.30±0.00	0.49±0.00	**0.27±0.00**	0.31±0.00	**0.27±0.00**	0.32±0.00	0.33±0.00	0.46±0.00	0.30±0.00	0.30±0.00	0.29±0.00	**0.27±0.00**
ERO(p=1, style=uniform,η=0.6)	0.33±0.00	0.48±0.00	**0.32±0.00**	0.34±0.01	**0.32±0.00**	0.34±0.00	0.35±0.00	0.50±0.00	**0.32±0.00**	0.33±0.00	**0.32±0.00**	**0.32±0.00**
ERO(p=1, style=gamma,η=0.6)	0.35±0.00	0.48±0.00	**0.32±0.00**	0.36±0.00	**0.32±0.00**	0.36±0.00	0.36±0.00	0.49±0.00	0.34±0.00	0.34±0.00	0.34±0.00	**0.32±0.00**
ERO(p=1, style=uniform,η=0.7)	0.38±0.00	0.49±0.00	**0.37±0.00**	0.39±0.01	**0.37±0.00**	0.38±0.00	0.39±0.00	0.47±0.00	**0.37±0.00**	0.38±0.00	**0.37±0.00**	**0.37±0.00**
ERO(p=1, style=gamma,η=0.7)	0.39±0.00	0.50±0.00	0.38±0.00	0.40±0.00	**0.37±0.00**	0.40±0.00	0.40±0.00	0.49±0.00	0.39±0.00	0.39±0.00	0.38±0.00	**0.37±0.00**
ERO(p=1, style=uniform,η=0.8)	0.42±0.00	0.49±0.00	0.49±0.00	0.43±0.00	0.42±0.00	0.43±0.00	0.43±0.00	0.48±0.00	0.42±0.00	0.42±0.00	0.42±0.00	**0.41±0.00**
ERO(p=1, style=gamma,η=0.8)	0.43±0.00	0.49±0.00	0.49±0.00	0.44±0.00	0.42±0.00	0.43±0.00	0.43±0.00	0.50±0.00	0.43±0.00	0.43±0.00	0.43±0.00	**0.41±0.00**

Table D.6: Performance on $\mathcal{L}_{\text{upset, simple}}$ for each variant in the proposed GNNRank framework, compared with the worst and the best baseline method, averaged over 10 runs, and plus/minus one standard deviation. "avg" for time series data sets first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best is marked in **bold red** while the second best is highlighted in underline blue .

Type	Baselines		DIMPA					IB				
Data	worst baseline	best baseline	dist	innerproduct	proximal dist	proximal innerproduct	proximal baseline	dist	innerproduct	proximal dist	proximal innerproduct	proximal baseline
HeadToHead	2.01±0.00	1.00±0.00	1.07±0.01	1.07±0.01	1.12±0.01	1.09±0.01	0.97±0.00	0.99±0.00	0.99±0.00	0.99±0.00	0.99±0.00	**0.96±0.00**
Finance	1.98±0.00	1.61±0.00	1.63±0.00	1.63±0.00	1.63±0.00	1.63±0.00	1.61±0.00	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	1.61±0.00
Animal	2.02±0.32	0.33±0.00	0.41±0.09	0.48±0.19	0.41±0.05	0.42±0.05	0.42±0.05	0.61±0.07	0.57±0.08	0.90±0.33	0.83±0.13	**0.25±0.00**
Faculty: Business	2.01±0.03	0.41±0.00	0.41±0.02	0.38±0.01	0.42±0.02	0.47±0.03	0.39±0.01	0.92±0.15	0.94±0.07	0.89±0.02	0.90±0.04	**0.36±0.00**
Faculty: CS	1.99±0.27	0.33±0.00	0.33±0.03	0.34±0.01	0.38±0.01	0.36±0.01	0.33±0.01	0.92±0.01	0.95±0.06	0.92±0.03	0.92±0.03	**0.32±0.00**
Faculty: History	2.16±0.80	0.30±0.01	0.30±0.01	**0.28±0.01**	0.30±0.01	0.31±0.01	0.30±0.01	0.92±0.01	0.94±0.08	0.93±0.06	0.90±0.06	0.30±0.01
Basketball(1985)	2.11±0.00	0.75±0.00	0.77±0.01	0.80±0.09	0.85±0.02	0.81±0.02	**0.71±0.01**	0.94±0.01	1.04±0.02	0.95±0.01	1.12±0.13	**0.71±0.00**
Basketball(1986)	1.99±0.00	0.78±0.00	0.79±0.01	0.78±0.01	0.83±0.02	0.82±0.01	0.70±0.00	0.90±0.02	1.01±0.02	0.92±0.02	1.02±0.03	**0.69±0.00**
Basketball(1987)	1.94±0.00	0.82±0.00	0.86±0.01	0.84±0.01	0.90±0.02	0.89±0.02	**0.77±0.01**	0.93±0.01	1.05±0.04	0.96±0.02	1.08±0.04	**0.77±0.00**
Basketball(1988)	1.93±0.00	0.74±0.00	0.76±0.01	0.78±0.08	0.81±0.02	0.80±0.01	**0.70±0.01**	0.91±0.01	1.01±0.01	0.95±0.03	1.02±0.02	**0.70±0.00**
Basketball(1989)	1.87±0.00	0.75±0.00	0.78±0.02	0.81±0.08	0.85±0.03	0.82±0.01	0.71±0.01	0.94±0.02	1.00±0.01	0.93±0.02	1.02±0.02	**0.70±0.00**
Basketball(1990)	1.95±0.00	0.75±0.00	0.77±0.02	0.81±0.01	0.84±0.02	0.81±0.02	**0.70±0.01**	0.93±0.02	1.01±0.03	0.95±0.02	1.02±0.03	**0.70±0.00**
Basketball(1991)	2.03±0.00	0.77±0.00	0.80±0.02	0.78±0.01	0.85±0.03	0.83±0.02	0.73±0.01	0.93±0.01	1.01±0.02	0.97±0.01	1.03±0.03	**0.70±0.00**
Basketball(1992)	2.00±0.00	0.72±0.00	0.76±0.01	0.74±0.01	0.80±0.01	0.78±0.02	0.69±0.01	0.93±0.01	1.02±0.02	0.97±0.02	1.05±0.04	**0.67±0.00**
Basketball(1993)	2.04±0.00	0.74±0.00	0.76±0.02	0.76±0.01	0.83±0.02	0.81±0.02	**0.68±0.01**	0.91±0.03	1.01±0.02	0.95±0.03	1.05±0.06	**0.68±0.00**
Basketball(1994)	2.04±0.00	0.74±0.00	0.76±0.02	0.75±0.01	0.82±0.02	0.80±0.02	0.71±0.01	0.90±0.02	1.02±0.03	0.94±0.01	1.08±0.08	**0.69±0.00**
Basketball(1995)	2.02±0.00	0.77±0.00	0.79±0.01	0.78±0.01	0.83±0.02	0.81±0.02	**0.72±0.01**	0.93±0.02	1.03±0.02	0.96±0.01	1.05±0.03	**0.72±0.00**
Basketball(1996)	2.05±0.00	0.81±0.00	0.84±0.01	0.84±0.01	0.92±0.01	0.89±0.01	**0.77±0.01**	0.92±0.02	1.03±0.05	0.94±0.01	1.09±0.08	**0.77±0.00**
Basketball(1997)	1.95±0.00	0.81±0.00	0.84±0.01	0.83±0.01	0.90±0.02	0.87±0.02	**0.76±0.01**	0.94±0.01	1.05±0.04	0.96±0.01	1.11±0.09	0.77±0.00
Basketball(1998)	1.92±0.00	0.77±0.00	0.81±0.02	0.80±0.01	0.88±0.02	0.84±0.02	**0.74±0.01**	0.90±0.02	1.01±0.01	0.94±0.02	1.04±0.06	0.75±0.00
Basketball(1999)	2.02±0.00	0.81±0.00	0.84±0.01	0.86±0.06	0.91±0.02	0.88±0.03	0.76±0.01	0.96±0.01	1.01±0.01	0.98±0.01	1.03±0.02	**0.74±0.00**
Basketball(2000)	1.97±0.00	0.84±0.00	0.84±0.01	0.82±0.01	0.92±0.01	0.90±0.02	**0.77±0.01**	0.94±0.01	1.01±0.01	0.95±0.02	1.04±0.04	**0.77±0.00**
Basketball(2001)	2.03±0.00	0.75±0.00	0.78±0.02	0.77±0.01	0.87±0.02	0.83±0.02	**0.71±0.01**	0.96±0.02	1.01±0.01	0.95±0.01	1.03±0.03	**0.71±0.00**
Basketball(2002)	1.94±0.00	0.80±0.00	0.81±0.01	0.80±0.01	0.86±0.02	0.84±0.03	**0.73±0.01**	0.91±0.01	1.02±0.02	0.93±0.03	1.06±0.11	**0.73±0.00**
Basketball(2003)	2.05±0.00	0.80±0.00	0.85±0.02	0.83±0.01	0.91±0.02	0.88±0.02	**0.78±0.01**	0.92±0.01	1.00±0.01	0.95±0.03	1.01±0.02	**0.78±0.00**
Basketball(2004)	2.00±0.11	0.75±0.00	0.77±0.01	0.78±0.01	0.84±0.02	0.81±0.02	0.70±0.01	0.93±0.01	0.99±0.01	0.96±0.01	1.05±0.15	**0.69±0.00**
Basketball(2005)	1.98±0.05	0.78±0.00	0.82±0.01	0.81±0.02	0.88±0.02	0.86±0.01	**0.73±0.00**	0.91±0.01	1.02±0.03	0.95±0.02	1.08±0.08	0.74±0.00
Basketball(2006)	2.00±0.00	0.77±0.00	0.82±0.01	0.83±0.01	0.90±0.02	0.88±0.02	0.75±0.01	0.94±0.02	1.01±0.02	0.95±0.01	1.04±0.05	**0.74±0.00**
Basketball(2007)	1.94±0.00	0.79±0.00	0.84±0.01	0.83±0.01	0.88±0.01	0.87±0.02	**0.77±0.01**	0.93±0.01	1.00±0.01	0.95±0.01	1.02±0.02	**0.77±0.00**
Basketball(2008)	2.01±0.00	0.80±0.00	0.83±0.01	0.84±0.04	0.88±0.02	0.86±0.01	0.79±0.01	0.92±0.02	1.00±0.01	0.95±0.02	1.01±0.04	**0.78±0.00**
Basketball(2009)	2.02±0.00	0.76±0.00	0.78±0.00	0.78±0.01	0.83±0.01	0.83±0.01	**0.72±0.01**	0.93±0.01	1.00±0.01	0.93±0.02	1.01±0.02	**0.72±0.00**
Basketball(2010)	2.03±0.00	0.77±0.00	0.83±0.01	0.82±0.01	0.89±0.01	0.86±0.01	0.73±0.01	0.89±0.01	1.01±0.02	0.92±0.02	1.01±0.02	**0.71±0.00**
Basketball(2011)	2.03±0.00	0.79±0.00	0.82±0.01	0.80±0.01	0.86±0.01	0.85±0.01	0.75±0.00	0.91±0.02	1.00±0.01	0.93±0.02	1.00±0.01	**0.74±0.00**
Basketball(2012)	1.97±0.00	0.76±0.00	0.78±0.02	0.78±0.01	0.84±0.02	0.83±0.02	0.70±0.01	0.90±0.03	1.00±0.01	0.94±0.03	1.01±0.01	**0.69±0.00**
Basketball(2013)	1.99±0.05	0.80±0.00	0.81±0.01	0.83±0.06	0.87±0.03	0.85±0.01	**0.76±0.00**	0.94±0.01	1.00±0.01	0.94±0.01	1.01±0.01	0.77±0.00
Basketball(2014)	2.06±0.00	0.79±0.00	0.84±0.01	0.83±0.01	0.89±0.02	0.88±0.01	**0.75±0.00**	0.94±0.02	1.01±0.01	0.97±0.01	1.02±0.02	0.76±0.00
Basketball finer(1985)	1.97±0.00	0.76±0.00	0.81±0.02	0.79±0.01	0.80±0.02	0.79±0.01	0.72±0.01	1.00±0.00	1.09±0.06	1.00±0.00	1.21±0.11	**0.71±0.00**
Basketball finer(1986)	1.99±0.00	0.77±0.00	0.84±0.02	0.81±0.01	0.91±0.03	0.81±0.01	**0.69±0.01**	1.00±0.00	1.11±0.06	1.00±0.00	1.19±0.10	**0.69±0.00**
Basketball finer(1987)	1.95±0.00	0.82±0.00	0.86±0.02	0.90±0.01	0.87±0.03	0.90±0.01	**0.77±0.01**	1.00±0.00	1.09±0.06	1.00±0.00	1.23±0.09	**0.77±0.00**
Basketball finer(1988)	1.97±0.00	0.78±0.00	0.81±0.01	0.80±0.01	0.81±0.01	0.80±0.01	**0.70±0.01**	1.00±0.00	1.10±0.11	1.00±0.00	1.22±0.14	**0.70±0.00**
Basketball finer(1989)	1.94±0.00	0.77±0.00	0.82±0.02	0.82±0.01	0.82±0.02	0.82±0.01	0.73±0.05	1.00±0.00	1.07±0.06	1.00±0.00	1.19±0.08	**0.70±0.00**
Basketball finer(1990)	1.98±0.00	0.79±0.00	0.80±0.02	0.80±0.01	0.81±0.02	0.81±0.01	0.72±0.01	1.00±0.00	1.10±0.06	1.00±0.00	1.20±0.12	**0.71±0.00**
Basketball finer(1991)	2.03±0.00	0.81±0.00	0.83±0.01	0.84±0.01	0.83±0.07	0.84±0.01	0.72±0.01	1.00±0.00	1.11±0.06	1.00±0.00	1.20±0.11	**0.71±0.00**
Basketball finer(1992)	1.88±0.00	0.73±0.00	0.75±0.02	0.76±0.01	0.75±0.02	0.76±0.01	0.68±0.01	1.00±0.00	1.08±0.06	1.00±0.00	1.21±0.07	**0.67±0.00**
Basketball finer(1993)	2.03±0.00	0.75±0.00	0.80±0.01	0.80±0.01	0.81±0.06	0.80±0.01	**0.68±0.01**	1.00±0.00	1.08±0.06	1.00±0.00	1.22±0.08	0.69±0.00
Basketball finer(1994)	2.01±0.00	0.74±0.00	0.77±0.01	0.78±0.01	0.77±0.01	0.78±0.01	0.69±0.01	1.00±0.00	1.10±0.06	1.00±0.00	1.21±0.06	**0.67±0.00**
Basketball finer(1995)	1.92±0.01	0.79±0.00	0.82±0.09	0.78±0.01	0.82±0.07	0.78±0.01	**0.72±0.01**	1.00±0.00	1.08±0.03	1.00±0.00	1.19±0.07	0.73±0.00
Basketball finer(1996)	1.95±0.00	0.81±0.00	0.86±0.01	0.90±0.01	0.87±0.01	0.90±0.01	**0.77±0.00**	1.00±0.00	1.07±0.04	1.00±0.00	1.15±0.08	**0.77±0.00**
Basketball finer(1997)	2.05±0.00	0.83±0.00	0.85±0.01	0.86±0.01	0.86±0.01	0.86±0.01	**0.75±0.01**	1.00±0.00	1.06±0.05	1.00±0.00	1.21±0.11	0.77±0.00
Basketball finer(1998)	1.92±0.00	0.78±0.00	0.82±0.02	0.86±0.01	0.86±0.02	0.86±0.01	**0.73±0.01**	1.00±0.00	1.08±0.05	1.00±0.00	1.23±0.07	0.74±0.00
Basketball finer(1999)	2.02±0.00	0.81±0.00	0.86±0.01	0.86±0.01	0.87±0.02	0.86±0.01	0.74±0.01	1.00±0.00	1.06±0.04	1.00±0.00	1.18±0.07	**0.73±0.00**
Basketball finer(2000)	1.97±0.00	0.84±0.00	0.88±0.01	0.90±0.01	0.88±0.01	0.90±0.01	**0.78±0.01**	1.00±0.00	1.09±0.06	1.00±0.00	1.24±0.03	**0.78±0.00**
Basketball finer(2001)	2.06±0.00	0.81±0.00	0.86±0.01	0.87±0.00	0.86±0.01	0.87±0.00	**0.73±0.01**	1.00±0.00	1.08±0.05	1.00±0.00	1.18±0.09	**0.73±0.00**
Basketball finer(2002)	2.03±0.00	0.87±0.00	0.83±0.01	0.84±0.01	0.83±0.02	0.84±0.01	**0.77±0.01**	1.00±0.00	1.06±0.04	1.00±0.00	1.18±0.07	0.78±0.00
Basketball finer(2003)	2.00±0.00	0.87±0.00	0.88±0.02	0.89±0.01	0.89±0.02	0.89±0.01	0.79±0.00	1.00±0.00	1.08±0.06	1.00±0.00	1.18±0.10	**0.78±0.00**
Basketball finer(2004)	1.98±0.02	0.77±0.00	0.84±0.01	0.84±0.01	0.93±0.01	0.84±0.01	**0.71±0.01**	1.00±0.00	1.04±0.04	1.00±0.00	1.19±0.09	0.72±0.00
Basketball finer(2005)	2.00±0.00	0.84±0.00	0.86±0.01	0.87±0.01	0.87±0.01	0.87±0.01	**0.75±0.01**	1.00±0.00	1.08±0.08	1.00±0.00	1.17±0.08	**0.75±0.00**
Basketball finer(2006)	1.97±0.00	0.85±0.00	0.88±0.02	0.90±0.01	0.88±0.02	0.90±0.00	0.77±0.01	1.00±0.00	1.08±0.05	1.00±0.00	1.21±0.08	**0.76±0.00**
Basketball finer(2007)	1.97±0.14	0.86±0.00	0.87±0.01	0.89±0.01	0.88±0.01	0.89±0.01	**0.80±0.01**	1.00±0.00	1.07±0.08	1.00±0.00	1.18±0.09	**0.80±0.00**
Basketball finer(2008)	1.98±0.00	0.85±0.00	0.86±0.02	0.87±0.02	0.86±0.02	0.88±0.02	0.79±0.01	1.00±0.00	1.07±0.07	1.00±0.00	1.20±0.11	**0.78±0.00**
Basketball finer(2009)	2.06±0.00	0.84±0.00	0.85±0.01	0.85±0.01	0.86±0.02	0.85±0.01	0.76±0.01	1.00±0.00	1.09±0.06	1.00±0.00	1.20±0.13	**0.75±0.00**
Basketball finer(2010)	1.98±0.00	0.82±0.00	0.85±0.01	0.85±0.01	0.86±0.02	0.85±0.01	**0.75±0.01**	1.00±0.00	1.02±0.03	1.00±0.00	1.12±0.05	**0.75±0.00**
Basketball finer(2011)	2.01±0.00	0.85±0.00	0.87±0.01	0.87±0.01	0.87±0.01	0.87±0.01	0.78±0.01	1.00±0.00	1.08±0.05	1.00±0.00	1.19±0.07	**0.77±0.00**
Basketball finer(2012)	1.97±0.00	0.80±0.00	0.83±0.01	0.85±0.01	0.84±0.01	0.85±0.01	**0.74±0.01**	1.00±0.00	1.10±0.07	1.00±0.00	1.19±0.07	0.75±0.00
Basketball finer(2013)	2.06±0.00	0.83±0.00	0.86±0.01	0.85±0.00	0.86±0.01	0.85±0.00	**0.78±0.01**	1.00±0.00	1.11±0.04	1.00±0.00	1.22±0.07	0.79±0.00
Basketball finer(2014)	2.07±0.00	0.84±0.00	0.88±0.03	0.87±0.01	0.89±0.03	0.87±0.00	**0.78±0.00**	1.00±0.00	1.11±0.10	1.00±0.00	1.22±0.09	0.79±0.00
Football(2009)	1.80±0.09	**0.61±0.00**	0.76±0.02	0.75±0.06	0.98±0.23	0.87±0.14	0.65±0.03	0.79±0.04	0.83±0.11	0.80±0.05	0.88±0.13	**0.61±0.00**
Football(2010)	1.94±0.20	1.17±0.00	1.20±0.05	1.21±0.04	1.15±0.14	1.26±0.07	1.17±0.04	**0.90±0.02**	0.96±0.05	0.96±0.04	0.95±0.05	1.17±0.01
Football(2011)	1.80±0.19	**0.80±0.00**	0.87±0.02	0.91±0.15	1.16±0.21	1.01±0.12	0.82±0.03	0.83±0.03	0.96±0.07	0.85±0.09	0.91±0.08	**0.80±0.01**
Football(2012)	1.73±0.18	0.86±0.00	0.81±0.09	0.83±0.17	0.98±0.11	0.97±0.10	0.85±0.04	**0.75±0.03**	0.86±0.12	0.80±0.08	0.82±0.11	0.83±0.01
Football(2013)	1.93±0.09	0.68±0.00	0.64±0.04	0.66±0.08	0.73±0.15	0.83±0.24	0.59±0.02	0.71±0.05	0.84±0.18	0.76±0.07	0.80±0.08	**0.56±0.00**
Football(2014)	1.83±0.11	1.08±0.00	0.97±0.06	0.98±0.17	1.22±0.12	1.18±0.12	1.06±0.06	**0.96±0.12**	1.08±0.27	**0.96±0.07**	1.03±0.03	1.08±0.00
Football finer(2009)	1.91±0.17	0.66±0.00	0.76±0.03	0.76±0.18	0.80±0.08	0.83±0.09	0.68±0.02	0.99±0.04	0.98±0.05	0.95±0.12	0.98±0.04	**0.65±0.02**
Football finer(2010)	2.11±0.01	1.17±0.00	1.23±0.09	1.25±0.16	1.23±0.04	1.27±0.06	1.20±0.04	1.00±0.01	1.00±0.01	1.00±0.00	**0.99±0.02**	1.17±0.01
Football finer(2011)	2.14±0.02	0.85±0.00	0.92±0.16	1.00±0.09	1.00±0.09	0.95±0.10	0.85±0.03	0.99±0.04	0.99±0.02	0.99±0.04	0.97±0.09	**0.84±0.02**
Football finer(2012)	2.15±0.01	**0.86±0.00**	0.93±0.07	0.99±0.05	0.95±0.05	0.97±0.06	0.89±0.07	1.00±0.00	0.99±0.03	0.99±0.03	0.98±0.05	**0.86±0.03**
Football finer(2013)	2.01±0.00	0.73±0.00	0.74±0.04	0.73±0.19	0.82±0.10	0.88±0.23	0.62±0.04	0.99±0.04	0.98±0.05	0.94±0.07	0.94±0.08	**0.57±0.01**
Football finer(2014)	2.09±0.17	1.05±0.00	1.07±0.05	1.10±0.15	1.22±0.11	1.20±0.26	1.07±0.06	**1.00±0.00**	1.03±0.10	**1.00±0.03**	1.03±0.08	1.07±0.02

Table D.7: Performance on $\mathcal{L}_{\text{upset, naive}}$ for each variant in the proposed GNNRank framework, compared with the worst and the best baseline method, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue .

Type	Baselines		DIMPA					IB				
Data	worst baseline	best baseline	dist	innerproduct	proximal dist	proximal innerproduct	proximal baseline	dist	innerproduct	proximal dist	proximal innerproduct	proximal baseline
HeadToHead	0.50±0.00	0.25±0.00	0.27±0.00	0.27±0.00	0.28±0.00	0.27±0.00	**0.24±0.00**	0.55±0.07	0.49±0.05	0.64±0.08	0.51±0.03	**0.24±0.00**
Finance	0.50±0.00	**0.40±0.00**	0.41±0.00	0.41±0.00	0.41±0.00	0.41±0.00	**0.40±0.00**	0.44±0.00	0.46±0.02	0.45±0.01	0.45±0.01	**0.40±0.00**
Animal	0.58±0.11	0.08±0.00	0.10±0.02	0.12±0.05	0.10±0.01	0.10±0.01	0.11±0.01	0.52±0.12	0.36±0.09	0.48±0.09	0.39±0.10	**0.06±0.00**
Faculty: Business	0.50±0.01	0.10±0.00	0.10±0.00	0.10±0.00	0.11±0.00	0.12±0.01	0.10±0.00	0.24±0.07	0.24±0.02	0.25±0.07	0.23±0.01	**0.09±0.00**
Faculty: CS	0.50±0.07	**0.08±0.00**	**0.08±0.01**	0.09±0.00	0.09±0.00	0.09±0.00	**0.08±0.00**	0.48±0.02	0.44±0.07	0.44±0.02	0.46±0.03	**0.08±0.00**
Faculty: History	0.54±0.20	0.08±0.00	**0.07±0.00**	**0.07±0.00**	**0.07±0.00**	0.08±0.00	0.08±0.00	0.50±0.08	0.43±0.16	0.51±0.24	0.46±0.21	0.08±0.00
Basketball(1985)	0.53±0.00	0.19±0.00	0.19±0.00	0.20±0.02	0.21±0.01	0.20±0.00	**0.18±0.00**	0.56±0.10	0.49±0.13	0.79±0.23	0.54±0.13	**0.18±0.00**
Basketball(1986)	0.50±0.00	0.19±0.00	0.20±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.18±0.00	0.61±0.15	0.46±0.11	0.77±0.18	0.54±0.07	**0.17±0.00**
Basketball(1987)	0.48±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.23±0.01	0.22±0.00	**0.19±0.00**	0.67±0.21	0.46±0.10	0.75±0.24	0.57±0.06	**0.19±0.00**
Basketball(1988)	0.48±0.00	0.19±0.00	0.19±0.00	0.19±0.02	0.20±0.01	0.20±0.00	0.18±0.00	0.72±0.06	0.46±0.14	0.82±0.09	0.56±0.09	**0.17±0.00**
Basketball(1989)	0.47±0.00	0.19±0.00	0.19±0.00	0.20±0.02	0.21±0.01	0.21±0.00	0.18±0.00	0.50±0.16	0.49±0.09	0.67±0.27	0.52±0.11	**0.17±0.00**
Basketball(1990)	0.49±0.00	0.19±0.00	0.19±0.00	0.20±0.00	0.21±0.00	0.20±0.01	0.18±0.00	0.61±0.12	0.46±0.09	0.80±0.21	0.53±0.09	**0.17±0.00**
Basketball(1991)	0.51±0.00	0.19±0.00	0.20±0.00	0.19±0.00	0.21±0.01	0.21±0.00	**0.18±0.00**	0.72±0.06	0.48±0.07	0.88±0.15	0.55±0.08	**0.18±0.00**
Basketball(1992)	0.50±0.00	0.18±0.00	0.19±0.00	0.18±0.00	0.20±0.00	0.20±0.00	**0.17±0.00**	0.56±0.20	0.43±0.08	0.70±0.27	0.54±0.08	**0.17±0.00**
Basketball(1993)	0.51±0.00	0.19±0.00	0.19±0.00	0.19±0.00	0.21±0.00	0.20±0.01	**0.17±0.00**	0.67±0.11	0.51±0.07	0.85±0.21	0.57±0.10	**0.17±0.00**
Basketball(1994)	0.51±0.00	0.18±0.00	0.19±0.00	0.19±0.00	0.20±0.01	0.20±0.00	0.18±0.00	0.52±0.22	0.43±0.08	0.63±0.29	0.52±0.08	**0.17±0.00**
Basketball(1995)	0.50±0.00	0.19±0.00	0.20±0.00	0.19±0.00	0.21±0.00	0.20±0.01	**0.18±0.00**	0.57±0.17	0.46±0.08	0.77±0.24	0.53±0.07	**0.18±0.00**
Basketball(1996)	0.51±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.23±0.00	0.22±0.00	**0.19±0.00**	0.65±0.15	0.49±0.08	0.84±0.20	0.57±0.04	**0.19±0.00**
Basketball(1997)	0.49±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	0.66±0.17	0.48±0.07	0.82±0.19	0.53±0.06	**0.19±0.00**
Basketball(1998)	0.48±0.00	0.19±0.00	0.20±0.00	0.20±0.00	0.22±0.01	0.21±0.00	**0.18±0.00**	0.65±0.20	0.45±0.08	0.78±0.26	0.56±0.08	0.19±0.00
Basketball(1999)	0.50±0.00	0.20±0.00	0.21±0.00	0.22±0.01	0.23±0.00	0.22±0.01	0.19±0.00	0.79±0.07	0.45±0.07	0.90±0.10	0.60±0.12	**0.18±0.00**
Basketball(2000)	0.49±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.23±0.00	0.22±0.01	**0.19±0.00**	0.62±0.14	0.45±0.06	0.80±0.17	0.54±0.09	**0.19±0.00**
Basketball(2001)	0.51±0.00	0.19±0.00	0.20±0.01	0.19±0.00	0.22±0.01	0.21±0.00	**0.18±0.00**	0.49±0.16	0.46±0.05	0.67±0.30	0.51±0.10	**0.18±0.00**
Basketball(2002)	0.49±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.22±0.00	0.21±0.01	**0.18±0.00**	0.48±0.21	0.49±0.06	0.61±0.28	0.50±0.08	**0.18±0.00**
Basketball(2003)	0.51±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.23±0.00	0.22±0.00	0.20±0.00	0.55±0.13	0.52±0.10	0.79±0.18	0.54±0.06	**0.19±0.00**
Basketball(2004)	0.50±0.03	0.19±0.00	0.19±0.00	0.19±0.00	0.21±0.00	0.20±0.00	**0.17±0.00**	0.74±0.12	0.45±0.06	0.86±0.07	0.60±0.09	**0.17±0.00**
Basketball(2005)	0.49±0.01	0.19±0.00	0.21±0.00	0.20±0.00	0.22±0.01	0.21±0.00	**0.18±0.00**	0.73±0.08	0.48±0.08	0.84±0.20	0.57±0.11	**0.18±0.00**
Basketball(2006)	0.50±0.00	**0.19±0.00**	0.21±0.00	0.21±0.00	0.23±0.00	0.22±0.01	**0.19±0.00**	0.60±0.18	0.46±0.06	0.76±0.26	0.56±0.08	**0.19±0.00**
Basketball(2007)	0.49±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	0.55±0.16	0.49±0.09	0.76±0.25	0.55±0.07	**0.19±0.00**
Basketball(2008)	0.50±0.00	**0.20±0.00**	0.21±0.00	0.21±0.01	0.22±0.00	0.21±0.00	**0.20±0.00**	0.72±0.17	0.47±0.09	0.86±0.20	0.57±0.10	**0.20±0.00**
Basketball(2009)	0.51±0.00	0.19±0.00	0.20±0.00	0.19±0.00	0.21±0.00	0.21±0.00	**0.18±0.00**	0.48±0.15	0.47±0.07	0.70±0.23	0.51±0.12	**0.18±0.00**
Basketball(2010)	0.51±0.00	0.19±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	**0.18±0.00**	0.54±0.20	0.41±0.04	0.68±0.23	0.52±0.15	**0.18±0.00**
Basketball(2011)	0.51±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.22±0.00	0.21±0.00	0.19±0.00	0.50±0.13	0.46±0.14	0.73±0.21	0.55±0.14	**0.18±0.00**
Basketball(2012)	0.49±0.00	0.19±0.00	0.19±0.00	0.19±0.00	0.21±0.00	0.21±0.00	0.18±0.00	0.64±0.10	0.48±0.13	0.81±0.21	0.56±0.11	**0.17±0.00**
Basketball(2013)	0.50±0.01	0.20±0.00	0.20±0.00	0.21±0.02	0.22±0.01	0.21±0.00	**0.19±0.00**	0.49±0.16	0.46±0.10	0.66±0.26	0.50±0.08	**0.19±0.00**
Basketball(2014)	0.52±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	0.76±0.05	0.48±0.12	0.89±0.10	0.59±0.12	**0.19±0.00**
Basketball finer(1985)	0.49±0.00	0.19±0.00	0.20±0.01	0.20±0.00	0.20±0.01	0.20±0.00	**0.18±0.00**	0.99±0.02	0.81±0.12	0.99±0.01	0.75±0.06	**0.18±0.00**
Basketball finer(1986)	0.50±0.00	0.19±0.00	0.21±0.00	0.20±0.00	0.23±0.01	0.20±0.00	**0.17±0.00**	0.99±0.01	0.82±0.10	0.98±0.06	0.76±0.06	**0.17±0.00**
Basketball finer(1987)	0.49±0.00	0.20±0.00	0.21±0.01	0.23±0.00	0.22±0.01	0.23±0.00	**0.19±0.00**	0.99±0.02	0.79±0.12	0.98±0.06	0.74±0.03	**0.19±0.00**
Basketball finer(1988)	0.49±0.00	0.19±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.20±0.00	**0.18±0.00**	0.99±0.01	0.76±0.15	0.97±0.09	0.73±0.09	**0.18±0.00**
Basketball finer(1989)	0.49±0.00	0.19±0.00	0.20±0.01	0.20±0.00	0.20±0.01	0.20±0.00	**0.18±0.01**	0.98±0.05	0.75±0.15	1.00±0.01	0.71±0.12	**0.18±0.00**
Basketball finer(1990)	0.50±0.00	0.20±0.00	0.20±0.01	0.20±0.00	0.20±0.01	0.20±0.00	**0.18±0.00**	0.98±0.05	0.79±0.15	1.00±0.01	0.74±0.04	**0.18±0.00**
Basketball finer(1991)	0.51±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.23±0.09	0.21±0.00	**0.18±0.00**	0.96±0.10	0.78±0.14	0.97±0.08	0.71±0.11	**0.18±0.00**
Basketball finer(1992)	0.47±0.00	0.18±0.00	0.19±0.00	0.19±0.00	0.19±0.01	0.19±0.00	**0.17±0.00**	0.94±0.10	0.77±0.18	0.99±0.02	0.72±0.14	**0.17±0.00**
Basketball finer(1993)	0.51±0.00	0.19±0.00	0.20±0.00	0.20±0.00	0.22±0.01	0.20±0.00	**0.17±0.00**	0.98±0.04	0.79±0.15	1.00±0.01	0.75±0.05	**0.17±0.00**
Basketball finer(1994)	0.50±0.00	0.18±0.00	0.19±0.00	0.20±0.00	0.19±0.00	0.20±0.00	**0.17±0.00**	0.98±0.04	0.80±0.14	0.98±0.06	0.75±0.10	**0.17±0.00**
Basketball finer(1995)	0.48±0.00	0.20±0.00	0.21±0.00	0.20±0.00	0.22±0.00	0.20±0.00	**0.18±0.00**	0.94±0.08	0.77±0.18	0.99±0.02	0.73±0.15	**0.18±0.00**
Basketball finer(1996)	0.49±0.00	0.20±0.00	0.22±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	0.97±0.06	0.79±0.15	0.97±0.07	0.74±0.13	**0.19±0.00**
Basketball finer(1997)	0.51±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	0.96±0.07	0.79±0.16	0.99±0.03	0.74±0.13	**0.19±0.00**
Basketball finer(1998)	0.48±0.00	0.20±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.18±0.00**	0.99±0.02	0.83±0.12	1.00±0.00	0.75±0.03	**0.18±0.00**
Basketball finer(1999)	0.50±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.21±0.00	**0.18±0.00**	0.99±0.03	0.84±0.10	0.99±0.01	0.77±0.04	**0.18±0.00**
Basketball finer(2000)	0.49±0.00	0.21±0.00	0.22±0.00	0.23±0.00	0.22±0.00	0.23±0.00	0.20±0.00	1.00±0.01	0.84±0.11	1.00±0.00	0.75±0.08	**0.19±0.00**
Basketball finer(2001)	0.51±0.00	0.20±0.00	0.21±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.18±0.00**	0.97±0.09	0.78±0.12	0.99±0.02	0.75±0.05	**0.18±0.00**
Basketball finer(2002)	0.51±0.00	0.22±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.19±0.00**	0.98±0.07	0.83±0.07	0.97±0.10	0.77±0.06	**0.19±0.00**
Basketball finer(2003)	0.50±0.00	0.22±0.00	0.22±0.00	0.22±0.00	0.22±0.00	0.22±0.00	0.20±0.00	0.98±0.05	0.84±0.02	1.00±0.01	0.77±0.04	**0.19±0.00**
Basketball finer(2004)	0.49±0.00	0.19±0.00	0.21±0.00	0.21±0.00	0.23±0.00	0.21±0.00	**0.18±0.00**	0.98±0.05	0.79±0.15	1.00±0.01	0.72±0.13	**0.18±0.00**
Basketball finer(2005)	0.50±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	0.99±0.02	0.83±0.11	0.98±0.06	0.74±0.05	**0.19±0.00**
Basketball finer(2006)	0.49±0.00	0.21±0.00	0.22±0.01	0.23±0.00	0.22±0.01	0.23±0.00	**0.19±0.00**	0.99±0.03	0.80±0.13	0.99±0.02	0.73±0.05	**0.19±0.00**
Basketball finer(2007)	0.49±0.03	0.21±0.00	0.22±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.20±0.00**	1.00±0.00	0.82±0.02	1.00±0.00	0.76±0.05	**0.20±0.00**
Basketball finer(2008)	0.49±0.00	0.21±0.00	0.21±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.20±0.00**	1.00±0.00	0.84±0.02	1.00±0.00	0.77±0.05	**0.20±0.00**
Basketball finer(2009)	0.52±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.19±0.00**	0.98±0.05	0.82±0.11	1.00±0.00	0.76±0.07	**0.19±0.00**
Basketball finer(2010)	0.49±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.19±0.00**	0.99±0.02	0.83±0.01	1.00±0.01	0.75±0.05	**0.19±0.00**
Basketball finer(2011)	0.50±0.00	0.21±0.00	0.22±0.00	0.22±0.00	0.22±0.00	0.22±0.00	**0.19±0.00**	1.00±0.00	0.84±0.04	1.00±0.00	0.75±0.04	**0.19±0.00**
Basketball finer(2012)	0.49±0.00	0.20±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.18±0.00**	1.00±0.01	0.83±0.10	0.99±0.04	0.77±0.05	0.19±0.00
Basketball finer(2013)	0.52±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.20±0.00**	0.99±0.03	0.82±0.11	0.98±0.06	0.76±0.04	**0.20±0.00**
Basketball finer(2014)	0.52±0.00	0.21±0.00	0.22±0.01	0.22±0.00	0.22±0.01	0.22±0.00	**0.20±0.00**	1.00±0.01	0.82±0.02	1.00±0.01	0.76±0.04	**0.20±0.00**
Football(2009)	0.45±0.02	**0.15±0.00**	0.19±0.01	0.19±0.01	0.24±0.06	0.22±0.04	0.16±0.01	0.29±0.07	0.24±0.05	0.36±0.09	0.27±0.04	**0.15±0.00**
Football(2010)	0.49±0.05	**0.29±0.00**	0.30±0.01	0.30±0.01	0.31±0.01	0.31±0.02	**0.29±0.01**	0.37±0.05	0.38±0.04	0.48±0.10	0.38±0.04	**0.29±0.00**
Football(2011)	0.45±0.05	**0.20±0.00**	0.22±0.01	0.23±0.04	0.29±0.05	0.25±0.03	0.21±0.01	0.43±0.23	0.34±0.07	0.46±0.22	0.34±0.08	**0.20±0.00**
Football(2012)	0.43±0.04	0.22±0.00	**0.20±0.02**	0.21±0.04	0.24±0.03	0.24±0.02	0.21±0.01	0.25±0.02	0.30±0.08	0.32±0.07	0.30±0.09	0.21±0.00
Football(2013)	0.48±0.02	0.17±0.00	0.16±0.01	0.16±0.02	0.18±0.04	0.21±0.06	0.15±0.01	0.34±0.06	0.24±0.02	0.46±0.12	0.34±0.06	**0.14±0.00**
Football(2014)	0.46±0.03	0.27±0.00	**0.24±0.01**	0.25±0.04	0.31±0.03	0.30±0.03	0.26±0.01	0.28±0.02	0.32±0.04	0.27±0.03	0.29±0.04	0.27±0.00
Football finer(2009)	0.48±0.04	**0.16±0.00**	0.19±0.01	0.19±0.05	0.21±0.03	0.21±0.02	0.17±0.00	0.58±0.33	0.29±0.07	0.59±0.28	0.32±0.10	**0.16±0.01**
Football finer(2010)	0.53±0.00	**0.29±0.00**	0.31±0.02	0.31±0.04	0.31±0.01	0.32±0.02	0.30±0.01	0.64±0.24	0.43±0.06	0.62±0.23	0.41±0.04	**0.29±0.00**
Football finer(2011)	0.53±0.00	**0.21±0.00**	0.23±0.04	0.25±0.02	0.25±0.02	0.24±0.03	**0.21±0.01**	0.52±0.25	0.32±0.07	0.48±0.22	0.33±0.07	**0.21±0.01**
Football finer(2012)	0.54±0.00	0.22±0.00	0.24±0.02	0.25±0.01	0.24±0.01	0.24±0.02	0.22±0.02	0.77±0.31	0.39±0.12	0.59±0.29	0.36±0.05	**0.21±0.01**
Football finer(2013)	0.50±0.00	0.18±0.00	0.19±0.01	0.18±0.05	0.20±0.03	0.22±0.06	0.15±0.01	0.67±0.29	0.28±0.04	0.52±0.25	0.32±0.06	**0.14±0.00**
Football finer(2014)	0.52±0.04	**0.26±0.00**	0.27±0.01	0.28±0.04	0.30±0.03	0.30±0.06	0.27±0.02	0.40±0.03	0.45±0.06	0.43±0.03	0.44±0.06	0.27±0.01
ERO(p=0.05, style=uniform,η=0)	0.49±0.03	**0.03±0.00**	0.04±0.00	0.06±0.03	0.05±0.01	0.06±0.04	0.05±0.01	0.06±0.00	0.05±0.00	0.05±0.00	0.05±0.00	0.05±0.00
ERO(p=0.05, style=gamma,η=0)	0.51±0.00	**0.02±0.00**	0.04±0.00	0.07±0.06	0.04±0.00	0.04±0.00	0.04±0.01	0.10±0.00	0.09±0.00	0.09±0.00	0.09±0.00	0.07±0.01
ERO(p=0.05, style=uniform,η=0.1)	0.50±0.01	0.11±0.00	0.14±0.01	0.14±0.02	0.17±0.06	0.14±0.01	**0.09±0.00**	0.13±0.00	0.13±0.01	0.13±0.01	0.13±0.00	0.13±0.00
ERO(p=0.05, style=gamma,η=0.1)	0.51±0.00	0.11±0.00	0.19±0.00	0.21±0.00	0.24±0.01	0.23±0.01	**0.09±0.00**	0.20±0.00	0.22±0.03	0.20±0.00	0.20±0.01	0.11±0.01
ERO(p=0.05, style=uniform,η=0.2)	0.50±0.00	0.15±0.00	0.19±0.00	0.20±0.02	0.21±0.05	0.20±0.00	**0.13±0.00**	0.19±0.00	0.19±0.00	0.19±0.01	0.20±0.00	0.19±0.01
ERO(p=0.05, style=gamma,η=0.2)	0.52±0.00	0.15±0.00	0.24±0.00	0.27±0.00	0.32±0.01	0.28±0.01	**0.14±0.00**	0.27±0.00	0.29±0.00	0.28±0.01	0.29±0.01	**0.14±0.01**
ERO(p=0.05, style=uniform,η=0.3)	0.51±0.00	0.20±0.00	0.24±0.00	0.24±0.02	0.26±0.00	0.25±0.00	**0.18±0.01**	0.23±0.00	0.23±0.00	0.26±0.03	0.25±0.01	0.24±0.01
ERO(p=0.05, style=gamma,η=0.3)	0.51±0.00	0.20±0.00	0.31±0.01	0.32±0.01	0.35±0.01	0.32±0.01	**0.19±0.00**	0.32±0.00	0.34±0.01	0.32±0.01	0.33±0.01	**0.19±0.00**
ERO(p=0.05, style=uniform,η=0.4)	0.51±0.00	**0.23±0.00**	0.29±0.00	0.29±0.02	0.36±0.00	0.30±0.03	**0.23±0.01**	0.28±0.00	0.28±0.01	0.32±0.00	0.31±0.02	0.27±0.02
ERO(p=0.05, style=gamma,η=0.4)	0.51±0.00	0.24±0.00	0.34±0.04	0.36±0.01	0.40±0.04	0.36±0.01	**0.23±0.01**	0.34±0.01	0.37±0.01	0.35±0.01	0.37±0.04	**0.23±0.01**
ERO(p=0.05, style=uniform,η=0.5)	0.51±0.00	**0.27±0.00**	0.32±0.00	0.35±0.00	0.40±0.00	0.35±0.00	**0.27±0.00**	0.33±0.01	0.32±0.00	0.34±0.01	0.34±0.01	0.29±0.01
ERO(p=0.05, style=gamma,η=0.5)	0.49±0.00	0.28±0.00	0.39±0.02	0.39±0.00	0.42±0.02	0.39±0.00	**0.27±0.01**	0.38±0.01	0.40±0.01	0.38±0.01	0.40±0.01	**0.27±0.01**
ERO(p=0.05, style=uniform,η=0.6)	0.51±0.00	0.31±0.00	0.36±0.01	0.38±0.00	0.42±0.01	0.38±0.00	**0.30±0.00**	0.35±0.00	0.35±0.00	0.37±0.00	0.37±0.00	0.31±0.00
ERO(p=0.05, style=gamma,η=0.6)	0.50±0.00	0.32±0.00	0.40±0.01	0.42±0.00	0.44±0.01	0.42±0.00	**0.30±0.01**	0.41±0.01	0.42±0.01	0.44±0.01	0.44±0.02	**0.30±0.01**
ERO(p=0.05, style=uniform,η=0.7)	0.49±0.00	0.34±0.00	0.39±0.01	0.40±0.00	0.43±0.00	0.40±0.00	**0.32±0.00**	0.39±0.00	0.38±0.01	0.40±0.00	0.39±0.01	0.33±0.01
ERO(p=0.05, style=gamma,η=0.7)	0.50±0.00	0.35±0.00	0.46±0.01	0.45±0.01	0.46±0.01	0.45±0.01	**0.33±0.01**	0.43±0.02	0.43±0.01	0.45±0.01	0.46±0.02	**0.33±0.01**
ERO(p=0.05, style=uniform,η=0.8)	0.51±0.00	0.36±0.00	0.43±0.01	0.42±0.00	0.43±0.01	0.42±0.00	**0.34±0.00**	0.42±0.00	0.42±0.00	0.42±0.00	0.42±0.00	**0.34±0.00**
ERO(p=0.05, style=gamma,η=0.8)	0.50±0.00	0.36±0.00	0.45±0.01	0.44±0.00	0.45±0.01	0.44±0.00	**0.34±0.01**	0.43±0.02	0.44±0.01	0.45±0.04	0.46±0.05	**0.34±0.01**
ERO(p=1, style=uniform,η=0)	0.50±0.03	**0.00±0.00**	0.01±0.02	0.04±0.07	0.01±0.01	0.04±0.09	**0.00±0.01**	**0.00±0.00**	0.04±0.11	0.01±0.01	0.05±0.06	**0.00±0.00**
ERO(p=1, style=gamma,η=0)	0.50±0.05	**0.00±0.00**	0.03±0.03	0.01±0.03	0.03±0.04	0.04±0.05	0.02±0.00	**0.00±0.00**	0.01±0.01	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
ERO(p=1, style=uniform,η=0.1)	0.48±0.00	**0.06±0.00**	0.09±0.01	0.08±0.00	0.08±0.00	0.08±0.00	0.08±0.01	0.11±0.01	0.13±0.09	0.10±0.01	0.11±0.03	**0.06±0.00**
ERO(p=1, style=gamma,η=0.1)	0.46±0.00	**0.06±0.00**	0.10±0.00	0.12±0.01	0.10±0.00	0.13±0.01	0.11±0.01	0.20±0.15	0.13±0.02	0.21±0.18	0.12±0.02	**0.06±0.00**
ERO(p=1, style=uniform,η=0.2)	0.47±0.00	**0.11±0.00**	0.14±0.00	0.15±0.02	0.14±0.00	0.14±0.00	0.13±0.00	0.21±0.03	0.23±0.03	0.22±0.04	0.21±0.08	**0.11±0.00**
ERO(p=1, style=gamma,η=0.2)	0.48±0.00	**0.11±0.00**	0.15±0.00	0.16±0.02	0.18±0.01	0.17±0.01	0.17±0.00	0.29±0.13	0.20±0.05	0.27±0.11	0.18±0.00	**0.11±0.00**
ERO(p=1, style=uniform,η=0.3)	0.47±0.00	**0.16±0.00**	0.18±0.00	0.19±0.00	0.20±0.01	0.19±0.00	0.19±0.00	0.31±0.19	0.25±0.02	0.27±0.08	0.23±0.02	**0.16±0.00**
ERO(p=1, style=gamma,η=0.3)	0.48±0.00	**0.16±0.00**	0.20±0.00	0.22±0.02	0.23±0.01	0.22±0.01	0.21±0.00	0.39±0.13	0.25±0.01	0.36±0.08	0.24±0.01	**0.16±0.00**
ERO(p=1, style=uniform,η=0.4)	0.48±0.00	**0.21±0.00**	0.23±0.00	0.23±0.00	0.26±0.03	0.26±0.06	0.23±0.00	0.29±0.03	0.30±0.02	0.28±0.03	0.27±0.02	**0.21±0.00**
ERO(p=1, style=gamma,η=0.4)	0.49±0.00	**0.22±0.00**	0.25±0.00	0.26±0.02	0.27±0.00	0.27±0.01	0.25±0.00	0.52±0.28	0.30±0.02	0.36±0.07	0.29±0.00	**0.22±0.00**
ERO(p=1, style=uniform,η=0.5)	0.49±0.00	**0.27±0.00**	**0.27±0.00**	0.28±0.02	0.32±0.03	0.28±0.00	0.28±0.00	0.33±0.01	0.35±0.04	0.32±0.02	0.33±0.02	**0.27±0.00**
ERO(p=1, style=gamma,η=0.5)	0.49±0.00	**0.27±0.00**	0.29±0.00	0.31±0.02	0.32±0.01	0.32±0.02	0.30±0.00	0.66±0.25	0.37±0.05	0.37±0.04	0.33±0.00	**0.27±0.00**
ERO(p=1, style=uniform,η=0.6)	0.50±0.00	**0.32±0.00**	**0.32±0.00**	0.33±0.00	0.37±0.02	0.33±0.00	0.33±0.00	0.37±0.01	0.40±0.04	0.37±0.01	0.37±0.01	**0.32±0.00**
ERO(p=1, style=gamma,η=0.6)	0.49±0.00	**0.32±0.00**	0.34±0.00	0.35±0.02	0.36±0.02	0.35±0.03	0.34±0.02	0.61±0.25	0.42±0.07	0.38±0.01	0.38±0.01	**0.32±0.00**
ERO(p=1, style=uniform,η=0.7)	0.49±0.00	**0.37±0.00**	**0.37±0.00**	**0.37±0.00**	0.40±0.00	**0.37±0.00**	0.38±0.00	0.41±0.01	0.43±0.03	0.41±0.01	0.42±0.02	**0.37±0.00**
ERO(p=1, style=gamma,η=0.7)	0.50±0.00	**0.37±0.00**	0.39±0.00	0.38±0.00	0.40±0.01	0.38±0.00	**0.37±0.00**	0.62±0.16	0.44±0.02	0.48±0.17	0.42±0.01	**0.37±0.00**
ERO(p=1, style=uniform,η=0.8)	0.49±0.00	0.42±0.00	0.42±0.00	0.42±0.00	0.43±0.00	0.42±0.00	0.42±0.00	0.44±0.01	0.47±0.01	0.45±0.01	0.46±0.03	**0.41±0.00**
ERO(p=1, style=gamma,η=0.8)	0.50±0.00	0.42±0.00	0.43±0.00	0.43±0.00	0.44±0.01	0.43±0.00	**0.41±0.00**	0.49±0.05	0.46±0.02	0.45±0.00	0.45±0.01	**0.41±0.00**

Table D.8: Performance on $\mathcal{L}_{\text{upset, ratio}}$ for each variant in the proposed GNNRank framework, compared with the worst and the best baseline method, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue .

Type	Baselines		DIMPA					IB				
Data	worst baseline	best baseline	dist	innerproduct	proximal dist	proximal innerproduct	proximal baseline	dist	innerproduct	proximal dist	proximal innerproduct	proximal baseline
HeadToHead	1.19±0.09	0.71±0.01	**0.68±0.00**	0.69±0.01	0.95±0.00	0.95±0.00	0.94±0.00	0.96±0.00	0.96±0.00	0.96±0.00	0.96±0.00	0.94±0.00
Finance	0.64±0.00	**0.26±0.00**	**0.26±0.00**	**0.26±0.00**	0.27±0.00	0.27±0.00	0.27±0.00	0.27±0.00	0.27±0.00	0.27±0.00	0.27±0.00	0.27±0.00
Animal	1.34±0.00	**0.22±0.00**	0.24±0.03	0.35±0.24	0.68±0.02	0.67±0.01	0.66±0.00	0.60±0.14	0.45±0.02	0.79±0.01	0.72±0.02	0.66±0.00
Faculty: Business	1.40±0.13	0.34±0.00	0.33±0.01	**0.31±0.00**	0.89±0.00	0.90±0.00	0.89±0.01	0.80±0.20	0.80±0.07	0.93±0.02	0.93±0.01	0.89±0.00
Faculty: CS	1.33±0.16	0.28±0.00	**0.26±0.02**	**0.26±0.01**	0.92±0.01	0.91±0.00	0.86±0.00	0.90±0.00	0.90±0.06	0.95±0.00	0.95±0.00	0.90±0.00
Faculty: History	1.35±0.11	0.23±0.00	0.23±0.01	**0.21±0.00**	0.85±0.00	0.88±0.00	0.84±0.02	0.89±0.04	0.83±0.11	0.96±0.00	0.93±0.01	0.87±0.00
Basketball(1985)	1.32±0.00	0.54±0.04	**0.46±0.01**	0.48±0.11	0.82±0.00	0.82±0.00	0.82±0.00	0.81±0.02	0.90±0.02	0.85±0.01	0.85±0.00	0.82±0.00
Basketball(1986)	1.30±0.00	0.54±0.01	0.43±0.00	**0.42±0.00**	0.82±0.00	0.82±0.00	0.81±0.00	0.78±0.02	0.87±0.02	0.85±0.01	0.85±0.00	0.81±0.00
Basketball(1987)	1.37±0.00	0.56±0.00	0.49±0.01	**0.48±0.01**	0.83±0.00	0.83±0.00	0.83±0.00	0.82±0.01	0.91±0.04	0.85±0.01	0.86±0.00	0.82±0.00
Basketball(1988)	1.33±0.00	0.53±0.01	**0.45±0.01**	0.47±0.01	0.83±0.00	0.83±0.00	0.82±0.00	0.82±0.01	0.87±0.01	0.86±0.01	0.86±0.00	0.82±0.00
Basketball(1989)	1.34±0.00	0.55±0.01	**0.46±0.01**	0.50±0.01	0.83±0.00	0.83±0.00	0.83±0.00	0.80±0.04	0.87±0.01	0.86±0.01	0.86±0.00	0.83±0.00
Basketball(1990)	1.35±0.00	0.51±0.05	**0.44±0.01**	0.47±0.01	0.83±0.00	0.83±0.00	0.82±0.00	0.82±0.01	0.88±0.03	0.86±0.01	0.86±0.00	0.82±0.00
Basketball(1991)	1.34±0.00	0.53±0.03	0.46±0.00	**0.45±0.00**	0.83±0.00	0.83±0.00	0.82±0.00	0.84±0.01	0.88±0.02	0.86±0.00	0.86±0.01	0.82±0.00
Basketball(1992)	1.35±0.00	0.50±0.06	0.44±0.01	**0.43±0.00**	0.83±0.00	0.83±0.00	0.83±0.00	0.81±0.04	0.90±0.02	0.86±0.01	0.86±0.00	0.82±0.00
Basketball(1993)	1.28±0.00	0.51±0.03	0.45±0.01	**0.44±0.01**	0.82±0.00	0.82±0.00	0.82±0.00	0.80±0.06	0.88±0.02	0.85±0.01	0.85±0.00	0.82±0.00
Basketball(1994)	1.35±0.00	0.43±0.05	0.43±0.01	**0.42±0.00**	0.82±0.00	0.82±0.00	0.82±0.00	0.77±0.07	0.89±0.03	0.85±0.01	0.85±0.00	0.82±0.00
Basketball(1995)	1.36±0.00	0.47±0.06	0.45±0.01	**0.44±0.00**	0.82±0.00	0.82±0.00	0.81±0.00	0.81±0.02	0.89±0.02	0.85±0.01	0.85±0.00	0.81±0.00
Basketball(1996)	1.35±0.00	0.55±0.03	**0.48±0.00**	**0.48±0.00**	0.83±0.00	0.83±0.00	0.82±0.00	0.80±0.03	0.90±0.05	0.85±0.01	0.86±0.00	0.82±0.00
Basketball(1997)	1.30±0.00	0.56±0.00	0.49±0.00	**0.48±0.00**	0.83±0.00	0.83±0.00	0.82±0.00	0.82±0.02	0.91±0.04	0.85±0.01	0.86±0.00	0.82±0.00
Basketball(1998)	1.36±0.00	0.56±0.00	0.47±0.01	**0.46±0.00**	0.84±0.00	0.84±0.00	0.84±0.00	0.82±0.04	0.90±0.01	0.87±0.01	0.87±0.00	0.84±0.00
Basketball(1999)	1.32±0.00	0.55±0.03	**0.48±0.00**	0.51±0.01	0.84±0.00	0.84±0.00	0.83±0.00	0.85±0.01	0.89±0.01	0.87±0.00	0.87±0.00	0.83±0.00
Basketball(2000)	1.40±0.01	0.58±0.01	0.51±0.01	**0.50±0.00**	0.86±0.00	0.86±0.00	0.86±0.00	0.85±0.02	0.91±0.01	0.89±0.01	0.89±0.00	0.86±0.00
Basketball(2001)	1.40±0.00	0.55±0.04	0.48±0.01	**0.47±0.00**	0.85±0.00	0.85±0.00	0.85±0.00	0.83±0.03	0.90±0.01	0.88±0.01	0.88±0.00	0.85±0.00
Basketball(2002)	1.33±0.00	0.56±0.03	0.49±0.01	**0.48±0.01**	0.85±0.00	0.85±0.00	0.84±0.00	0.80±0.04	0.91±0.02	0.87±0.01	0.88±0.00	0.84±0.00
Basketball(2003)	1.40±0.00	0.56±0.04	0.52±0.01	**0.51±0.00**	0.85±0.00	0.85±0.00	0.85±0.00	0.82±0.02	0.89±0.01	0.88±0.01	0.88±0.00	0.85±0.00
Basketball(2004)	1.35±0.00	0.54±0.04	0.47±0.01	**0.46±0.00**	0.85±0.00	0.85±0.00	0.85±0.00	0.85±0.02	0.88±0.01	0.88±0.00	0.88±0.00	0.85±0.00
Basketball(2005)	1.29±0.01	0.57±0.01	0.50±0.01	**0.49±0.01**	0.86±0.00	0.86±0.00	0.85±0.00	0.84±0.01	0.91±0.03	0.88±0.01	0.88±0.00	0.85±0.00
Basketball(2006)	1.36±0.00	0.50±0.05	0.50±0.01	**0.49±0.01**	0.86±0.00	0.86±0.00	0.86±0.00	0.83±0.04	0.91±0.01	0.88±0.01	0.89±0.00	0.85±0.00
Basketball(2007)	1.41±0.00	0.57±0.00	**0.49±0.00**	**0.49±0.00**	0.86±0.00	0.86±0.00	0.86±0.00	0.83±0.03	0.91±0.01	0.89±0.01	0.89±0.00	0.86±0.00
Basketball(2008)	1.33±0.01	0.55±0.04	**0.51±0.01**	0.53±0.10	0.86±0.00	0.86±0.00	0.86±0.00	0.85±0.03	0.90±0.01	0.89±0.01	0.89±0.00	0.86±0.00
Basketball(2009)	1.24±0.00	0.56±0.00	0.48±0.00	**0.47±0.00**	0.86±0.00	0.86±0.00	0.86±0.00	0.83±0.03	0.90±0.01	0.89±0.01	0.89±0.00	0.86±0.00
Basketball(2010)	1.35±0.00	0.52±0.06	**0.49±0.00**	**0.49±0.00**	0.88±0.00	0.87±0.00	0.87±0.00	0.83±0.05	0.92±0.02	0.90±0.01	0.91±0.00	0.87±0.00
Basketball(2011)	1.34±0.00	0.52±0.05	0.50±0.01	**0.49±0.00**	0.87±0.00	0.87±0.00	0.86±0.00	0.83±0.03	0.91±0.01	0.89±0.01	0.89±0.00	0.86±0.00
Basketball(2012)	1.42±0.00	0.54±0.05	**0.48±0.01**	**0.48±0.01**	0.87±0.00	0.87±0.00	0.86±0.00	0.85±0.04	0.91±0.01	0.90±0.01	0.90±0.00	0.86±0.00
Basketball(2013)	1.34±0.00	0.58±0.03	**0.51±0.01**	0.53±0.10	0.87±0.00	0.87±0.00	0.86±0.00	0.84±0.03	0.91±0.01	0.89±0.01	0.90±0.00	0.86±0.00
Basketball(2014)	1.41±0.00	0.52±0.06	**0.49±0.01**	**0.49±0.01**	0.86±0.00	0.86±0.00	0.86±0.00	0.86±0.01	0.91±0.01	0.89±0.00	0.89±0.00	0.86±0.00
Basketball finer(1985)	0.57±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.19±0.02	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1986)	0.60±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.19±0.02	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1987)	0.58±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.21±0.02	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1988)	0.57±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.22±0.02	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1989)	0.56±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.22±0.03	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1990)	0.55±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.22±0.01	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1991)	0.54±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.23±0.03	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1992)	0.55±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.22±0.03	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1993)	0.58±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.20±0.02	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1994)	0.58±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.21±0.03	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1995)	0.55±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.21±0.02	0.01±0.00	0.01±0.00	0.01±0.00
Basketball finer(1996)	0.56±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(1997)	0.56±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(1998)	0.56±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.21±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(1999)	0.57±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.18±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2000)	0.55±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.21±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2001)	0.59±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2002)	0.57±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2003)	0.58±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.21±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2004)	0.57±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2005)	0.60±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2006)	0.57±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2007)	0.53±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.22±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2008)	0.59±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2009)	0.57±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2010)	0.58±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.20±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2011)	0.59±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.21±0.02	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2012)	0.57±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.22±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2013)	0.56±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.22±0.03	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**
Basketball finer(2014)	0.56±0.00	0.01±0.00	0.01±0.00	**0.00±0.00**	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.20±0.04	0.01±0.00	0.01±0.00	0.01±0.00
Football(2009)	1.16±0.05	0.48±0.02	**0.46±0.01**	0.49±0.11	0.73±0.04	0.71±0.03	0.69±0.00	0.73±0.05	0.70±0.16	0.81±0.02	0.79±0.03	0.69±0.00
Football(2010)	1.25±0.01	0.66±0.02	**0.65±0.00**	0.68±0.07	0.73±0.00	0.75±0.00	0.73±0.01	0.75±0.02	0.76±0.03	0.79±0.01	0.78±0.01	0.74±0.00
Football(2011)	1.17±0.00	**0.53±0.02**	**0.53±0.01**	0.54±0.07	0.73±0.01	0.71±0.01	0.69±0.00	0.70±0.04	0.79±0.07	0.77±0.03	0.77±0.04	0.69±0.00
Football(2012)	1.18±0.00	0.53±0.03	**0.51±0.01**	0.53±0.08	0.74±0.05	0.72±0.01	0.71±0.00	0.64±0.03	0.77±0.09	0.74±0.02	0.76±0.05	0.71±0.00
Football(2013)	0.94±0.00	0.48±0.04	**0.46±0.01**	0.47±0.12	0.72±0.02	0.72±0.03	0.71±0.00	0.65±0.06	0.69±0.11	0.80±0.03	0.79±0.04	0.71±0.00
Football(2014)	1.49±0.00	0.74±0.01	**0.69±0.01**	**0.69±0.07**	0.87±0.01	0.87±0.02	0.85±0.00	0.87±0.02	0.96±0.05	0.88±0.01	0.90±0.03	0.85±0.00
Football finer(2009)	0.73±0.04	**0.17±0.01**	**0.17±0.00**	**0.17±0.03**	0.21±0.00	0.22±0.01	0.21±0.00	0.30±0.01	0.30±0.01	0.29±0.01	0.28±0.02	0.21±0.00
Football finer(2010)	0.67±0.00	0.18±0.01	0.19±0.01	**0.17±0.00**	0.18±0.00	0.18±0.01	0.18±0.00	0.25±0.01	0.25±0.00	0.24±0.01	0.23±0.01	0.19±0.00
Football finer(2011)	0.58±0.00	**0.17±0.00**	**0.17±0.00**	0.19±0.05	0.19±0.00	0.20±0.00	0.19±0.00	0.28±0.01	0.27±0.02	0.26±0.01	0.26±0.02	0.20±0.00
Football finer(2012)	0.66±0.04	0.16±0.00	**0.15±0.00**	0.16±0.03	0.17±0.00	0.17±0.00	0.17±0.00	0.25±0.01	0.25±0.00	0.24±0.01	0.23±0.02	0.17±0.00
Football finer(2013)	0.52±0.00	0.21±0.01	0.21±0.01	**0.19±0.04**	0.25±0.00	0.25±0.02	0.24±0.01	0.33±0.01	0.33±0.01	0.33±0.01	0.31±0.02	0.25±0.00
Football finer(2014)	0.98±0.00	0.35±0.00	**0.34±0.00**	0.35±0.05	0.38±0.00	0.39±0.03	0.38±0.00	0.47±0.01	0.47±0.01	0.46±0.02	0.46±0.01	0.38±0.00

Table D.9: $\mathcal{L}_{\text{upset, simple}}$ comparison for different variants on selected real-world data, averaged over 10 runs, and plus/minus one standard deviation. "avg" for time series data first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best for each group of variants (GNNRank-N or GNNRank-P) is marked in **bold red** while the second best is highlighted in underline blue .

Methods	GNNRank-N			GNNRank-P						
Data/Variant	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	no pretrain	$\{\alpha_\gamma\}_{\gamma=1}^{\Gamma}$not trainable	$\Gamma = 3$	$\Gamma = 7$
Animal	0.43±0.06	0.59±0.08	**0.41±0.09**	**0.25±0.00**	**0.25±0.00**	**0.25±0.01**	**0.25±0.00**	**0.25±0.00**	**0.25±0.00**	**0.25±0.00**
Faculty: Business	0.40±0.02	0.49±0.16	**0.38±0.01**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**	**0.36±0.00**
Faculty: CS	0.35±0.01	0.36±0.01	**0.33±0.03**	**0.32±0.00**	**0.32±0.00**	**0.32±0.00**	0.33±0.00	**0.32±0.00**	**0.32±0.00**	**0.32±0.00**
Faculty: History	**0.28±0.01**	0.31±0.01	**0.28±0.01**	**0.30±0.01**	**0.30±0.01**	**0.30±0.02**	**0.30±0.01**	**0.30±0.01**	**0.30±0.01**	**0.30±0.01**
Football(2009)	0.76±0.03	0.79±0.04	**0.75±0.06**	**0.61±0.00**	**0.61±0.00**	**0.61±0.00**	**0.61±0.00**	**0.61±0.00**	**0.61±0.00**	**0.61±0.00**
Football(2010)	**0.90±0.02**	0.93±0.05	0.92±0.03	**0.95±0.05**	0.96±0.06	0.96±0.05	**0.95±0.06**	0.96±0.06	**0.95±0.05**	**0.95±0.04**
Football(2011)	0.86±0.10	**0.83±0.03**	0.84±0.08	**0.80±0.00**	**0.80±0.00**	**0.80±0.01**	**0.80±0.00**	**0.80±0.01**	**0.80±0.00**	**0.80±0.00**
Football(2012)	0.81±0.09	**0.75±0.03**	0.78±0.03	0.80±0.08	0.80±0.08	0.84±0.00	0.84±0.00	0.80±0.08	**0.78±0.03**	0.80±0.06
Football(2013)	0.65±0.03	0.75±0.05	**0.64±0.04**	**0.56±0.00**	**0.56±0.00**	**0.56±0.00**	**0.56±0.00**	**0.56±0.00**	**0.56±0.00**	**0.56±0.00**
Football(2014)	**0.96±0.12**	1.00±0.11	0.98±0.17	0.98±0.06	0.96±0.07	0.98±0.09	0.99±0.10	0.98±0.06	**0.95±0.05**	0.97±0.08
Football finer(2009)	**0.76±0.03**	0.93±0.38	**0.76±0.18**	0.65±0.02	0.65±0.02	0.65±0.03	0.66±0.00	0.65±0.03	**0.63±0.03**	0.64±0.02
Football finer(2010)	**1.00±0.01**	**1.00±0.01**	**1.00±0.01**	**0.99±0.02**	1.00±0.01	1.00±0.00	**0.99±0.02**	1.00±0.01	1.00±0.00	1.00±0.00
Football finer(2011)	0.99±0.03	**0.92±0.16**	0.99±0.04	0.85±0.01	0.84±0.02	0.85±0.00	0.85±0.00	0.85±0.01	**0.83±0.03**	0.84±0.03
Football finer(2012)	**0.93±0.07**	1.00±0.02	0.96±0.03	**0.86±0.02**	**0.86±0.03**	**0.86±0.02**	**0.86±0.00**	**0.86±0.03**	**0.86±0.01**	**0.86±0.00**
Football finer(2013)	0.74±0.04	0.98±0.04	**0.73±0.19**	0.59±0.02	0.69±0.03	**0.57±0.01**	0.58±0.00	0.58±0.01	**0.57±0.02**	**0.57±0.02**
Football finer(2014)	**1.00±0.00**	1.01±0.04	**1.00±0.00**	**1.00±0.00**	1.01±0.03	**1.00±0.03**	1.08±0.12	**1.00±0.03**	**1.00±0.00**	**1.00±0.00**

Table D.10: $\mathcal{L}_{\text{upset, naive}}$ comparison for different variants on selected real-world data, averaged over 10 runs, and plus/minus one standard deviation. "avg" for time series data first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best is marked in **bold red** while the second best is highlighted in underline blue .

Methods	GNNRank-N			GNNRank-P						
Data/Variant	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	no pretrain	$\{\alpha_\gamma\}_{\gamma=1}^{\Gamma}$not trainable	$\Gamma = 3$	$\Gamma = 7$
Animal	0.11±0.02	0.15±0.02	**0.10±0.02**	**0.06±0.00**	**0.06±0.00**	**0.06±0.00**	**0.06±0.00**	**0.06±0.00**	**0.06±0.00**	**0.06±0.00**
Faculty: Business	**0.10±0.00**	0.12±0.04	**0.10±0.00**	**0.09±0.00**	**0.09±0.00**	**0.09±0.00**	**0.09±0.00**	**0.09±0.00**	**0.09±0.00**	**0.09±0.00**
Faculty: CS	0.09±0.00	0.09±0.00	**0.08±0.01**	**0.08±0.00**	**0.08±0.00**	**0.08±0.00**	**0.08±0.00**	**0.08±0.00**	**0.08±0.00**	**0.08±0.00**
Faculty: History	**0.07±0.00**	0.08±0.00	**0.07±0.00**	**0.07±0.00**	0.08±0.00	0.08±0.00	**0.07±0.00**	**0.07±0.00**	0.08±0.00	0.08±0.00
Football(2009)	**0.19±0.01**	0.26±0.06	**0.19±0.01**	**0.15±0.00**	**0.15±0.00**	**0.15±0.00**	**0.15±0.00**	**0.15±0.00**	**0.15±0.00**	**0.15±0.00**
Football(2010)	**0.30±0.01**	0.36±0.06	**0.30±0.01**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.01**	**0.29±0.00**
Football(2011)	**0.22±0.01**	0.26±0.07	**0.22±0.01**	**0.20±0.00**	**0.20±0.00**	**0.20±0.00**	**0.20±0.00**	**0.20±0.00**	**0.20±0.00**	**0.20±0.00**
Football(2012)	**0.20±0.02**	0.29±0.06	0.21±0.04	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	0.21±0.00	**0.20±0.00**	0.21±0.00
Football(2013)	**0.16±0.01**	0.20±0.07	**0.16±0.01**	**0.14±0.00**	**0.14±0.00**	**0.14±0.00**	**0.14±0.00**	**0.14±0.00**	**0.14±0.00**	**0.14±0.00**
Football(2014)	**0.24±0.01**	0.30±0.02	0.25±0.04	**0.26±0.01**	0.27±0.00	0.27±0.02	0.27±0.00	**0.26±0.01**	0.27±0.02	0.27±0.00
Football finer(2009)	**0.19±0.01**	0.23±0.09	**0.19±0.05**	**0.16±0.01**	**0.16±0.01**	**0.16±0.01**	**0.16±0.00**	**0.16±0.01**	**0.16±0.01**	**0.16±0.00**
Football finer(2010)	**0.31±0.01**	0.35±0.05	**0.31±0.02**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**	**0.29±0.00**
Football finer(2011)	0.25±0.01	**0.23±0.04**	0.25±0.01	**0.21±0.00**	**0.21±0.01**	**0.21±0.00**	**0.21±0.00**	**0.21±0.01**	**0.21±0.01**	**0.21±0.01**
Football finer(2012)	**0.24±0.02**	0.25±0.04	**0.24±0.01**	**0.21±0.00**	**0.21±0.01**	0.22±0.01	0.22±0.00	**0.21±0.01**	0.22±0.00	0.22±0.00
Football finer(2013)	0.19±0.01	0.26±0.12	**0.18±0.05**	0.15±0.00	0.17±0.01	**0.14±0.00**	0.15±0.00	**0.14±0.00**	**0.14±0.00**	**0.14±0.00**
Football finer(2014)	**0.27±0.01**	0.35±0.09	0.28±0.04	**0.27±0.02**	**0.27±0.00**	**0.27±0.02**	**0.27±0.00**	**0.27±0.02**	**0.27±0.00**	**0.27±0.01**

Table D.11: $\mathcal{L}_{\text{upset, ratio}}$ comparison for different variants on selected real-world data, averaged over 10 runs, and plus/minus one standard deviation. "avg" for time series data first average over all seasons, then consider mean and standard deviation over the 10 averaged values. The best for each group of variants (GNNRank-N or GNNRank-P) is marked in **bold red** while the second best is highlighted in underline blue .

Methods	GNNRank-N			GNNRank-P						
Data/Variant	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	loss sum	$\mathcal{L}_{\text{upset,margin}}$	$\mathcal{L}_{\text{upset, ratio}}$	no pretrain	$\{\alpha_\gamma\}_{\gamma=1}^{\Gamma}$not trainable	$\Gamma = 3$	$\Gamma = 7$
Animal	**0.24±0.01**	0.41±0.09	**0.24±0.03**	**0.66±0.00**	**0.66±0.00**	**0.66±0.00**	**0.66±0.00**	**0.66±0.00**	**0.66±0.00**	**0.66±0.00**
Faculty: Business	0.32±0.02	0.71±0.14	**0.31±0.00**	**0.89±0.00**	**0.89±0.01**	**0.89±0.00**	**0.89±0.01**	**0.89±0.00**	**0.89±0.01**	**0.89±0.00**
Faculty: CS	0.27±0.01	0.69±0.05	**0.26±0.02**	0.90±0.00	**0.86±0.00**	0.90±0.00	0.90±0.00	**0.86±0.00**	**0.86±0.00**	**0.86±0.00**
Faculty: History	**0.21±0.00**	0.60±0.10	**0.21±0.00**	0.86±0.00	**0.84±0.02**	0.85±0.00	0.87±0.00	**0.84±0.02**	**0.84±0.02**	**0.84±0.02**
Football(2009)	**0.46±0.01**	0.72±0.14	0.48±0.01	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**
Football(2010)	**0.65±0.00**	0.75±0.02	0.68±0.07	**0.73±0.01**	0.74±0.00	0.74±0.00	**0.73±0.00**	**0.73±0.01**	**0.73±0.00**	**0.73±0.01**
Football(2011)	**0.53±0.01**	0.70±0.04	0.54±0.02	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**	**0.69±0.00**
Football(2012)	**0.51±0.01**	0.65±0.04	0.53±0.08	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**
Football(2013)	**0.46±0.01**	0.56±0.16	**0.46±0.01**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**	**0.71±0.00**
Football(2014)	**0.69±0.01**	0.92±0.07	**0.69±0.07**	**0.85±0.00**	**0.85±0.00**	**0.85±0.00**	**0.85±0.00**	**0.85±0.00**	**0.85±0.00**	**0.85±0.00**
Football finer(2009)	**0.17±0.00**	0.28±0.07	**0.17±0.03**	0.21±0.00	0.21±0.00	0.21±0.01	0.21±0.00	0.21±0.00	**0.20±0.00**	0.21±0.01
Football finer(2010)	0.19±0.03	0.25±0.00	**0.17±0.00**	**0.18±0.00**	0.19±0.00	**0.18±0.00**	**0.18±0.00**	**0.18±0.00**	**0.18±0.00**	**0.18±0.00**
Football finer(2011)	**0.17±0.00**	0.22±0.09	0.18±0.01	**0.19±0.00**	0.20±0.02	**0.19±0.00**	**0.19±0.00**	**0.19±0.00**	**0.19±0.00**	**0.19±0.00**
Football finer(2012)	**0.15±0.00**	0.23±0.02	0.16±0.03	**0.17±0.00**	**0.17±0.00**	**0.17±0.00**	**0.17±0.00**	**0.17±0.00**	**0.17±0.00**	**0.17±0.00**
Football finer(2013)	0.21±0.01	0.29±0.03	**0.19±0.04**	0.25±0.01	0.25±0.00	**0.24±0.01**	0.25±0.00	**0.24±0.01**	**0.24±0.00**	0.25±0.00
Football finer(2014)	**0.34±0.00**	0.46±0.04	0.35±0.05	**0.38±0.00**	**0.38±0.00**	**0.38±0.00**	**0.38±0.00**	**0.38±0.00**	**0.38±0.00**	**0.38±0.00**

Table D.12: Result table on $\mathcal{L}_{\text{upset, simple}}$ for each year in the time series matches, applying the trained model for 1985 on all seasons without further training, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue . As MVR could not generate results after a week, we omit the results here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	Directly Apply	Train Specifically
Basketball finer(1985)	0.76±0.00	1.63±0.00	1.96±0.10	1.46±0.05	0.83±0.00	1.18±0.00	1.16±0.00	1.97±0.00	1.00±0.00	0.87±0.00	**0.71±0.00**	**0.71±0.00**
Basketball finer(1986)	0.77±0.00	1.81±0.00	1.99±0.00	1.42±0.06	0.84±0.00	1.16±0.00	1.15±0.00	1.99±0.00	1.09±0.00	0.86±0.00	**0.69±0.00**	**0.69±0.00**
Basketball finer(1987)	0.82±0.00	1.79±0.00	1.87±0.00	1.41±0.06	0.89±0.00	1.17±0.00	1.21±0.00	1.95±0.00	0.99±0.00	0.91±0.00	**0.77±0.00**	**0.77±0.00**
Basketball finer(1988)	0.78±0.00	1.79±0.00	1.90±0.00	1.43±0.10	0.84±0.00	1.23±0.00	1.19±0.00	1.97±0.00	0.97±0.00	0.83±0.00	**0.70±0.00**	**0.70±0.00**
Basketball finer(1989)	0.77±0.00	1.67±0.00	1.86±0.00	1.43±0.05	0.83±0.00	1.13±0.00	1.14±0.00	1.94±0.00	0.99±0.00	0.90±0.00	0.88±0.52	**0.70±0.00**
Basketball finer(1990)	0.79±0.00	1.67±0.00	1.93±0.00	1.45±0.05	0.82±0.00	1.28±0.00	1.17±0.00	1.98±0.00	0.91±0.00	0.84±0.00	**0.71±0.00**	**0.71±0.00**
Basketball finer(1991)	0.81±0.00	1.83±0.00	2.03±0.00	1.36±0.06	0.83±0.00	1.38±0.00	1.31±0.00	1.97±0.00	0.99±0.00	0.89±0.00	**0.71±0.00**	**0.71±0.00**
Basketball finer(1992)	0.73±0.00	1.72±0.00	1.88±0.00	1.33±0.06	0.77±0.00	1.26±0.00	1.21±0.00	1.87±0.00	0.95±0.00	0.84±0.00	**0.67±0.00**	**0.67±0.00**
Basketball finer(1993)	0.75±0.00	1.66±0.00	2.03±0.00	1.35±0.05	0.78±0.00	1.18±0.00	1.10±0.00	1.97±0.00	0.98±0.00	0.86±0.00	**0.69±0.00**	**0.69±0.00**
Basketball finer(1994)	0.74±0.00	1.69±0.00	2.01±0.00	1.35±0.08	0.78±0.00	1.23±0.00	1.10±0.00	1.94±0.00	0.90±0.00	0.83±0.00	**0.67±0.00**	**0.67±0.00**
Basketball finer(1995)	0.79±0.00	1.78±0.00	1.89±0.00	1.35±0.06	0.83±0.00	1.19±0.00	1.13±0.00	1.92±0.01	0.95±0.00	0.87±0.00	**0.73±0.00**	**0.73±0.00**
Basketball finer(1996)	0.81±0.00	1.67±0.00	1.95±0.00	1.44±0.06	0.88±0.00	1.22±0.00	1.20±0.00	1.94±0.00	1.08±0.00	0.95±0.00	**0.77±0.00**	**0.77±0.00**
Basketball finer(1997)	0.83±0.00	1.77±0.00	1.94±0.00	1.40±0.04	0.86±0.00	1.19±0.00	1.16±0.00	2.05±0.00	0.96±0.00	0.92±0.00	**0.77±0.00**	**0.77±0.00**
Basketball finer(1998)	0.78±0.00	1.70±0.00	1.92±0.00	1.36±0.07	0.83±0.00	1.14±0.00	1.13±0.00	1.91±0.00	0.97±0.00	0.90±0.00	**0.74±0.00**	**0.74±0.00**
Basketball finer(1999)	0.81±0.00	1.64±0.00	2.02±0.00	1.38±0.07	0.86±0.00	1.17±0.00	1.11±0.00	1.99±0.00	1.17±0.00	0.94±0.00	**0.73±0.00**	**0.73±0.00**
Basketball finer(2000)	0.84±0.00	1.75±0.00	1.97±0.00	1.39±0.05	0.90±0.00	1.26±0.00	1.18±0.00	1.92±0.00	1.12±0.00	0.95±0.00	**0.78±0.00**	**0.78±0.00**
Basketball finer(2001)	0.81±0.00	1.69±0.00	2.06±0.00	1.41±0.06	0.86±0.00	1.25±0.00	1.18±0.00	2.03±0.00	1.08±0.00	0.97±0.00	**0.73±0.00**	**0.73±0.00**
Basketball finer(2002)	0.87±0.00	1.75±0.00	1.86±0.00	1.43±0.08	0.89±0.00	1.20±0.00	1.13±0.00	2.03±0.00	1.07±0.00	0.92±0.00	**0.78±0.00**	**0.78±0.00**
Basketball finer(2003)	0.87±0.00	1.78±0.00	1.98±0.07	1.46±0.09	0.91±0.00	1.18±0.00	1.14±0.00	2.00±0.00	1.02±0.00	0.95±0.00	**0.78±0.00**	**0.78±0.00**
Basketball finer(2004)	0.77±0.00	1.71±0.06	1.87±0.00	1.41±0.06	0.80±0.00	1.17±0.00	1.13±0.00	1.98±0.02	0.95±0.00	0.88±0.00	**0.72±0.00**	**0.72±0.00**
Basketball finer(2005)	0.84±0.00	1.82±0.00	1.93±0.00	1.38±0.06	0.88±0.00	1.14±0.00	1.09±0.00	2.00±0.00	1.08±0.00	0.95±0.00	0.93±0.52	**0.75±0.00**
Basketball finer(2006)	0.86±0.00	1.76±0.00	1.97±0.00	1.40±0.07	0.85±0.00	1.21±0.00	1.11±0.00	1.96±0.00	1.06±0.00	0.94±0.00	**0.76±0.00**	**0.76±0.00**
Basketball finer(2007)	0.86±0.00	1.85±0.00	1.97±0.14	1.39±0.05	0.93±0.00	1.15±0.00	1.09±0.00	1.92±0.00	0.95±0.00	0.93±0.00	**0.80±0.00**	**0.80±0.00**
Basketball finer(2008)	0.85±0.00	1.72±0.00	1.98±0.00	1.36±0.08	0.88±0.00	1.20±0.00	1.13±0.00	1.96±0.00	0.99±0.00	0.91±0.00	**0.78±0.00**	**0.78±0.00**
Basketball finer(2009)	0.84±0.00	1.71±0.00	2.00±0.04	1.37±0.04	0.90±0.00	1.17±0.00	1.12±0.00	2.06±0.00	0.99±0.00	0.91±0.00	**0.75±0.00**	**0.75±0.00**
Basketball finer(2010)	0.82±0.00	1.68±0.00	1.98±0.00	1.34±0.06	0.84±0.00	1.10±0.00	1.13±0.00	1.97±0.00	0.94±0.00	0.92±0.00	**0.75±0.00**	**0.75±0.00**
Basketball finer(2011)	0.85±0.00	1.65±0.00	1.96±0.00	1.36±0.06	0.87±0.00	1.12±0.00	1.14±0.00	2.01±0.00	0.96±0.00	0.89±0.00	**0.77±0.00**	**0.77±0.00**
Basketball finer(2012)	0.80±0.00	1.68±0.00	1.97±0.00	1.37±0.07	0.83±0.00	1.17±0.00	1.10±0.00	1.91±0.00	0.93±0.00	0.89±0.00	**0.75±0.00**	**0.75±0.00**
Basketball finer(2013)	0.83±0.00	1.73±0.00	2.06±0.00	1.37±0.07	0.88±0.00	1.15±0.00	1.13±0.00	1.97±0.00	1.01±0.00	0.92±0.00	**0.79±0.00**	**0.79±0.00**
Basketball finer(2014)	0.84±0.00	1.77±0.19	2.07±0.00	1.42±0.09	0.86±0.00	1.18±0.00	1.13±0.00	1.97±0.00	1.02±0.00	0.88±0.00	**0.79±0.00**	**0.79±0.00**

Table D.13: Result table on $\mathcal{L}_{\text{upset, naive}}$ for each year in the time series matches, applying the trained model for 1985 directly without further training, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue . As MVR could not generate scores, we omit the results here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	Directly Apply	Train Specifically
Basketball finer(1985)	0.19±0.00	0.41±0.00	0.49±0.02	0.36±0.01	0.21±0.00	0.29±0.00	0.29±0.00	0.49±0.00	0.25±0.00	0.22±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(1986)	0.19±0.00	0.45±0.00	0.50±0.00	0.36±0.02	0.21±0.00	0.29±0.00	0.29±0.00	0.50±0.00	0.27±0.00	0.21±0.00	**0.17±0.00**	**0.17±0.00**
Basketball finer(1987)	0.20±0.00	0.45±0.00	0.47±0.00	0.35±0.01	0.22±0.00	0.29±0.00	0.30±0.00	0.49±0.00	0.25±0.00	0.23±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(1988)	0.19±0.00	0.45±0.00	0.48±0.00	0.36±0.03	0.21±0.00	0.31±0.00	0.30±0.00	0.49±0.00	0.24±0.00	0.21±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(1989)	0.19±0.00	0.42±0.00	0.46±0.00	0.36±0.01	0.21±0.00	0.28±0.00	0.29±0.00	0.49±0.00	0.25±0.00	0.23±0.00	0.22±0.13	**0.18±0.00**
Basketball finer(1990)	0.20±0.00	0.42±0.00	0.48±0.00	0.36±0.01	0.21±0.00	0.32±0.00	0.29±0.00	0.50±0.00	0.23±0.00	0.21±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(1991)	0.20±0.00	0.46±0.00	0.51±0.00	0.34±0.02	0.21±0.00	0.35±0.00	0.33±0.00	0.49±0.00	0.25±0.00	0.22±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(1992)	0.18±0.00	0.43±0.00	0.47±0.00	0.33±0.01	0.19±0.00	0.31±0.00	0.30±0.00	0.47±0.00	0.24±0.00	0.21±0.00	**0.17±0.00**	**0.17±0.00**
Basketball finer(1993)	0.19±0.00	0.42±0.00	0.51±0.00	0.34±0.01	0.20±0.00	0.29±0.00	0.27±0.00	0.49±0.00	0.25±0.00	0.21±0.00	**0.17±0.00**	**0.17±0.00**
Basketball finer(1994)	0.18±0.00	0.42±0.00	0.50±0.00	0.34±0.02	0.19±0.00	0.31±0.00	0.27±0.00	0.49±0.00	0.22±0.00	0.21±0.00	**0.17±0.00**	**0.17±0.00**
Basketball finer(1995)	0.20±0.00	0.44±0.00	0.47±0.00	0.34±0.02	0.21±0.00	0.30±0.00	0.28±0.00	0.48±0.00	0.24±0.00	0.22±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(1996)	0.20±0.00	0.42±0.00	0.49±0.00	0.36±0.02	0.22±0.00	0.30±0.00	0.30±0.00	0.49±0.00	0.27±0.00	0.24±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(1997)	0.21±0.00	0.44±0.00	0.49±0.00	0.35±0.01	0.21±0.00	0.30±0.00	0.29±0.00	0.51±0.00	0.24±0.00	0.23±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(1998)	0.20±0.00	0.42±0.00	0.48±0.00	0.34±0.02	0.21±0.00	0.29±0.00	0.28±0.00	0.48±0.00	0.24±0.00	0.22±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(1999)	0.20±0.00	0.41±0.00	0.50±0.00	0.34±0.02	0.22±0.00	0.29±0.00	0.28±0.00	0.50±0.00	0.29±0.00	0.24±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(2000)	0.21±0.00	0.44±0.00	0.49±0.00	0.35±0.01	0.23±0.00	0.32±0.00	0.30±0.00	0.48±0.00	0.28±0.00	0.24±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2001)	0.20±0.00	0.42±0.00	0.51±0.00	0.35±0.01	0.21±0.00	0.31±0.00	0.30±0.00	0.51±0.00	0.27±0.00	0.24±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(2002)	0.22±0.00	0.44±0.00	0.47±0.00	0.36±0.02	0.22±0.00	0.30±0.00	0.28±0.00	0.51±0.00	0.27±0.00	0.23±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2003)	0.22±0.00	0.45±0.00	0.50±0.02	0.36±0.02	0.23±0.00	0.29±0.00	0.29±0.00	0.50±0.00	0.26±0.00	0.24±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2004)	0.19±0.00	0.43±0.01	0.47±0.00	0.35±0.01	0.20±0.00	0.29±0.00	0.28±0.00	0.49±0.00	0.24±0.00	0.22±0.00	**0.18±0.00**	**0.18±0.00**
Basketball finer(2005)	0.21±0.00	0.46±0.00	0.48±0.00	0.34±0.01	0.22±0.00	0.29±0.00	0.27±0.00	0.50±0.00	0.27±0.00	0.24±0.00	0.23±0.13	**0.19±0.00**
Basketball finer(2006)	0.21±0.00	0.44±0.00	0.49±0.00	0.35±0.02	0.21±0.00	0.30±0.00	0.28±0.00	0.49±0.00	0.27±0.00	0.23±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2007)	0.21±0.00	0.46±0.00	0.49±0.03	0.35±0.01	0.23±0.00	0.29±0.00	0.27±0.00	0.48±0.00	0.24±0.00	0.23±0.00	**0.20±0.00**	**0.20±0.00**
Basketball finer(2008)	0.21±0.00	0.43±0.00	0.49±0.00	0.34±0.02	0.22±0.00	0.30±0.00	0.28±0.00	0.49±0.00	0.25±0.00	0.23±0.00	**0.20±0.00**	**0.20±0.00**
Basketball finer(2009)	0.21±0.00	0.43±0.00	0.50±0.01	0.34±0.01	0.22±0.00	0.29±0.00	0.28±0.00	0.52±0.00	0.25±0.00	0.23±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2010)	0.20±0.00	0.42±0.00	0.49±0.00	0.33±0.02	0.21±0.00	0.28±0.00	0.28±0.00	0.49±0.00	0.24±0.00	0.23±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2011)	0.21±0.00	0.41±0.00	0.49±0.00	0.34±0.01	0.22±0.00	0.28±0.00	0.28±0.00	0.50±0.00	0.24±0.00	0.22±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2012)	0.20±0.00	0.42±0.00	0.49±0.00	0.34±0.02	0.21±0.00	0.29±0.00	0.28±0.00	0.48±0.00	0.23±0.00	0.22±0.00	**0.19±0.00**	**0.19±0.00**
Basketball finer(2013)	0.21±0.00	0.43±0.00	0.52±0.00	0.34±0.02	0.22±0.00	0.29±0.00	0.28±0.00	0.49±0.00	0.25±0.00	0.23±0.00	**0.20±0.00**	**0.20±0.00**
Basketball finer(2014)	0.21±0.00	0.44±0.05	0.52±0.00	0.36±0.02	0.22±0.00	0.30±0.00	0.28±0.00	0.49±0.00	0.26±0.00	0.22±0.00	**0.20±0.00**	**0.20±0.00**

Table D.14: Result table on $\mathcal{L}_{\text{upset, ratio}}$ for each year in the time series matches, applying the trained model for 1985 directly without further training, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue . As MVR could not generate scores, we omit the results here.

Data	SpringRank	SyncRank	SerialRank	BTL	DavidScore	Eig.Cent.	PageRank	RankCent.	SVD_RS	SVD_NRS	Directly Apply	Train Specifically
Basketball finer(1985)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.47±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.46±0.00	0.46±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1986)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.49±0.00	**0.01±0.00**	**0.01±0.00**	0.60±0.00	0.48±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1987)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.46±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1988)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.56±0.00	0.52±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1989)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.49±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.49±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1990)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.49±0.00	**0.01±0.00**	**0.01±0.00**	0.55±0.00	0.48±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1991)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.49±0.00	**0.01±0.00**	**0.01±0.00**	0.54±0.00	0.49±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1992)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.49±0.00	**0.01±0.00**	**0.01±0.00**	0.55±0.00	0.44±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1993)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.47±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1994)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.49±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.46±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1995)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.55±0.00	0.47±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1996)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.48±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.45±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1997)	0.04±0.00	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.48±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1998)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.47±0.00	0.45±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(1999)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.48±0.00	0.46±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2000)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.55±0.00	0.50±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2001)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.59±0.00	0.51±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2002)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.51±0.00	0.50±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2003)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.50±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.47±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2004)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.45±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2005)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.60±0.00	0.43±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2006)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.46±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2007)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.53±0.00	**0.01±0.00**	**0.01±0.00**	0.53±0.00	0.49±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2008)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.51±0.00	**0.01±0.00**	**0.01±0.00**	0.59±0.00	0.47±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2009)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.50±0.00	0.49±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2010)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.58±0.00	0.48±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2011)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.59±0.00	0.46±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2012)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.57±0.00	0.48±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2013)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.52±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.46±0.00	0.48±0.00	**0.01±0.00**	**0.01±0.00**
Basketball finer(2014)	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	**0.01±0.00**	0.53±0.00	**0.01±0.00**	**0.01±0.00**	0.56±0.00	0.48±0.00	0.47±0.00	**0.01±0.00**	**0.01±0.00**

Table D.15: Result table on $\mathcal{L}_{\text{upset, simple}}$ improvement with "proximal baseline" training starting from a baseline as initial guess, for individual directed graphs, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue .

Data	SpringRank	SyncRank	SerialRank	BTL	Eig.Cent.	PageRank	SVD_NRS
HeadToHead	-0.01±0.00	-0.98±0.00	**-1.02±0.00**	-0.12±0.03	-0.48±0.00	-0.37±0.00	-0.43±0.00
Finance	-0.63±0.00	**-0.98±0.00**	-0.61±0.00	-0.78±0.01	-0.74±0.00	-0.75±0.00	-0.64±0.00
Animal	-0.09±0.05	-1.36±0.24	**-1.42±0.50**	-0.01±0.01	-0.02±0.02	-0.08±0.07	-0.02±0.07
Faculty: Business	-0.03±0.00	-0.47±0.00	**-0.70±0.04**	0.00±0.02	-0.02±0.03	-0.04±0.03	-0.03±0.02
Faculty: CS	-0.00±0.00	-0.66±0.10	**-0.72±0.07**	0.00±0.01	0.01±0.03	-0.00±0.00	-0.10±0.04
Faculty: History	-0.02±0.01	-0.27±0.00	**-1.77±0.82**	0.00±0.01	-0.01±0.03	-0.00±0.00	-0.00±0.00
Football(2009)	-0.00±0.00	**-0.70±0.46**	-0.16±0.06	-0.03±0.08	-0.12±0.05	-0.24±0.06	-0.02±0.01
Football(2010)	-0.24±0.06	**-0.85±0.17**	-0.58±0.06	-0.22±0.09	-0.40±0.04	-0.41±0.04	-0.21±0.05
Football(2011)	-0.00±0.01	**-0.74±0.37**	-0.10±0.06	-0.03±0.04	-0.13±0.09	-0.19±0.09	-0.02±0.09
Football(2012)	-0.17±0.06	**-0.72±0.04**	-0.57±0.07	-0.14±0.10	-0.10±0.09	-0.22±0.03	-0.05±0.08
Football(2013)	-0.03±0.03	**-1.08±0.10**	-0.08±0.06	-0.06±0.05	-0.16±0.06	-0.15±0.03	-0.04±0.03
Football(2014)	-0.17±0.05	**-0.83±0.20**	-0.59±0.08	-0.17±0.06	-0.27±0.09	-0.24±0.07	-0.07±0.09
Football finer(2009)	-0.03±0.03	**-1.04±0.05**	-0.18±0.06	-0.06±0.14	-0.17±0.11	-0.24±0.12	-0.02±0.03
Football finer(2010)	-0.30±0.02	**-0.63±0.02**	-0.45±0.02	-0.31±0.09	-0.37±0.00	-0.49±0.00	-0.17±0.00
Football finer(2011)	-0.00±0.02	**-0.75±0.04**	-0.14±0.06	-0.03±0.06	-0.08±0.05	-0.19±0.04	-0.01±0.02
Football finer(2012)	-0.03±0.03	**-0.90±0.18**	-0.12±0.08	-0.03±0.06	-0.14±0.06	-0.29±0.05	-0.01±0.03
Football finer(2013)	-0.13±0.07	**-1.01±0.02**	-0.12±0.07	-0.08±0.05	-0.19±0.11	-0.26±0.09	-0.07±0.03
Football finer(2014)	-0.16±0.00	**-0.74±0.04**	-0.57±0.00	-0.19±0.04	-0.23±0.00	-0.23±0.00	-0.09±0.03
Basketball(1985)	-0.01±0.01	-0.91±0.00	**-1.26±0.04**	-0.10±0.07	-0.05±0.01	-0.07±0.02	-0.07±0.01
Basketball(1986)	-0.01±0.01	-1.10±0.00	**-1.11±0.03**	-0.14±0.02	-0.07±0.01	-0.06±0.02	-0.03±0.02
Basketball(1987)	-0.00±0.01	**-1.02±0.01**	-0.94±0.03	-0.11±0.02	-0.11±0.02	-0.01±0.01	-0.05±0.02
Basketball(1988)	-0.00±0.00	-0.96±0.00	**-1.07±0.04**	-0.11±0.04	-0.06±0.02	-0.04±0.02	-0.03±0.01
Basketball(1989)	0.00±0.00	**-1.04±0.00**	-0.99±0.02	-0.12±0.02	-0.00±0.00	-0.02±0.01	-0.07±0.01
Basketball(1990)	-0.00±0.01	-0.98±0.00	**-1.09±0.03**	-0.10±0.11	-0.09±0.01	-0.07±0.01	-0.05±0.01
Basketball(1991)	-0.02±0.01	-1.10±0.00	**-1.15±0.04**	-0.09±0.06	-0.03±0.01	-0.01±0.02	-0.08±0.01
Basketball(1992)	-0.01±0.01	**-1.07±0.00**	-1.05±0.02	-0.06±0.10	-0.04±0.01	0.00±0.00	-0.07±0.01
Basketball(1993)	-0.01±0.01	-1.00±0.01	**-1.18±0.06**	-0.09±0.07	-0.05±0.01	-0.06±0.01	-0.07±0.01
Basketball(1994)	0.00±0.00	-0.98±0.00	**-1.15±0.02**	-0.02±0.09	-0.02±0.02	-0.03±0.01	-0.05±0.01
Basketball(1995)	0.00±0.00	**-1.06±0.01**	-1.01±0.03	-0.03±0.10	-0.02±0.01	-0.06±0.01	-0.04±0.01
Basketball(1996)	-0.00±0.01	-0.90±0.01	**-1.11±0.02**	-0.08±0.08	-0.02±0.01	-0.02±0.01	-0.07±0.01
Basketball(1997)	-0.01±0.01	-0.96±0.01	**-1.02±0.04**	-0.12±0.04	-0.05±0.01	-0.03±0.01	-0.07±0.01
Basketball(1998)	-0.00±0.01	-0.95±0.01	**-1.00±0.06**	-0.14±0.02	-0.06±0.01	-0.02±0.01	-0.07±0.01
Basketball(1999)	-0.00±0.01	-0.83±0.00	**-1.10±0.02**	-0.09±0.07	-0.07±0.01	-0.07±0.01	-0.07±0.01
Basketball(2000)	-0.01±0.01	-1.00±0.00	**-1.03±0.03**	-0.09±0.02	-0.02±0.01	-0.03±0.01	-0.05±0.01
Basketball(2001)	-0.00±0.00	-1.02±0.00	**-1.10±0.11**	-0.09±0.08	-0.05±0.02	-0.02±0.02	-0.11±0.02
Basketball(2002)	-0.02±0.00	-0.95±0.00	**-1.05±0.02**	-0.09±0.06	-0.01±0.01	-0.06±0.01	-0.05±0.01
Basketball(2003)	0.00±0.00	-1.07±0.00	**-1.13±0.01**	-0.08±0.07	-0.08±0.01	-0.07±0.01	-0.07±0.01
Basketball(2004)	-0.00±0.00	-0.98±0.00	**-1.14±0.11**	-0.09±0.06	-0.05±0.01	-0.06±0.01	-0.07±0.01
Basketball(2005)	0.00±0.00	-1.05±0.01	**-1.06±0.06**	-0.14±0.03	-0.00±0.01	-0.06±0.01	-0.07±0.01
Basketball(2006)	0.00±0.00	-1.04±0.00	**-1.09±0.03**	0.02±0.09	0.00±0.00	-0.03±0.01	-0.04±0.01
Basketball(2007)	-0.00±0.00	**-1.10±0.00**	-1.03±0.08	-0.09±0.02	-0.03±0.01	-0.02±0.01	-0.04±0.01
Basketball(2008)	0.00±0.00	-0.97±0.00	**-1.10±0.03**	-0.03±0.07	-0.04±0.01	-0.05±0.01	-0.06±0.01
Basketball(2009)	-0.00±0.00	-1.02±0.01	**-1.17±0.02**	-0.12±0.02	-0.03±0.01	-0.03±0.01	-0.06±0.01
Basketball(2010)	-0.00±0.00	-0.83±0.00	**-1.06±0.01**	-0.03±0.13	-0.04±0.01	-0.04±0.01	-0.05±0.01
Basketball(2011)	-0.00±0.01	-0.90±0.00	**-1.09±0.03**	0.00±0.08	-0.04±0.01	-0.05±0.01	-0.02±0.01
Basketball(2012)	-0.00±0.00	**-1.11±0.00**	**-1.11±0.05**	-0.07±0.08	-0.04±0.01	-0.03±0.02	-0.04±0.01
Basketball(2013)	-0.00±0.00	-0.95±0.00	**-1.09±0.06**	-0.11±0.06	-0.06±0.01	-0.03±0.01	-0.07±0.01
Basketball(2014)	-0.00±0.01	-0.96±0.00	**-1.14±0.05**	-0.00±0.10	-0.01±0.01	-0.04±0.01	-0.02±0.01
Basketball finer(1985)	-0.01±0.01	-0.92±0.00	**-0.96±0.10**	-0.56±0.07	-0.35±0.01	-0.32±0.02	-0.07±0.01
Basketball finer(1986)	-0.00±0.00	**-1.12±0.00**	-1.01±0.03	-0.52±0.08	-0.30±0.02	-0.27±0.01	-0.05±0.01
Basketball finer(1987)	-0.01±0.00	**-1.02±0.01**	-0.90±0.02	-0.45±0.07	-0.25±0.01	-0.28±0.01	-0.05±0.01
Basketball finer(1988)	-0.01±0.00	**-1.09±0.01**	-1.02±0.02	-0.55±0.11	-0.38±0.02	-0.34±0.01	-0.03±0.01
Basketball finer(1989)	-0.00±0.00	**-0.96±0.00**	-0.87±0.03	-0.51±0.06	-0.27±0.01	-0.29±0.01	-0.07±0.01
Basketball finer(1990)	-0.01±0.01	-0.96±0.00	**-1.02±0.01**	-0.55±0.05	-0.42±0.01	-0.31±0.01	-0.03±0.01
Basketball finer(1991)	-0.01±0.00	**-1.13±0.00**	-1.03±0.00	-0.45±0.06	-0.46±0.02	-0.41±0.01	-0.05±0.01
Basketball finer(1992)	-0.00±0.00	**-1.04±0.00**	-0.99±0.02	-0.46±0.06	-0.41±0.02	-0.37±0.02	-0.07±0.01
Basketball finer(1993)	-0.00±0.01	-0.98±0.01	**-1.08±0.10**	-0.49±0.06	-0.35±0.01	-0.27±0.01	-0.05±0.01
Basketball finer(1994)	0.00±0.00	-1.02±0.00	**-1.11±0.04**	-0.50±0.07	-0.39±0.02	-0.28±0.01	-0.03±0.01
Basketball finer(1995)	-0.01±0.01	**-1.06±0.01**	-0.98±0.01	-0.48±0.08	-0.35±0.01	-0.30±0.01	-0.06±0.01
Basketball finer(1996)	-0.00±0.01	-0.90±0.00	**-0.95±0.00**	-0.44±0.06	-0.27±0.02	-0.25±0.02	-0.06±0.01
Basketball finer(1997)	-0.01±0.01	**-1.02±0.01**	-0.98±0.01	-0.47±0.04	-0.28±0.01	-0.24±0.01	-0.08±0.01
Basketball finer(1998)	-0.01±0.01	**-0.96±0.01**	-0.95±0.01	-0.44±0.06	-0.25±0.01	-0.24±0.01	-0.08±0.01
Basketball finer(1999)	-0.00±0.01	-0.90±0.00	**-1.05±0.02**	-0.44±0.08	-0.27±0.01	-0.19±0.01	-0.08±0.01
Basketball finer(2000)	0.00±0.00	-0.97±0.00	**-0.99±0.02**	-0.42±0.05	-0.31±0.02	-0.25±0.01	-0.06±0.01
Basketball finer(2001)	-0.00±0.00	-0.96±0.00	**-1.06±0.00**	-0.46±0.06	-0.31±0.01	-0.25±0.02	-0.10±0.01
Basketball finer(2002)	-0.04±0.02	**-0.99±0.01**	-0.92±0.02	-0.52±0.09	-0.30±0.02	-0.25±0.01	-0.05±0.01
Basketball finer(2003)	0.00±0.00	**-1.01±0.00**	-0.98±0.07	-0.49±0.10	-0.25±0.01	-0.21±0.01	-0.05±0.01
Basketball finer(2004)	-0.02±0.01	**-1.00±0.06**	-0.93±0.01	-0.48±0.07	-0.29±0.01	-0.24±0.01	-0.07±0.01
Basketball finer(2005)	-0.01±0.01	**-1.08±0.01**	-0.93±0.00	-0.42±0.05	-0.22±0.01	-0.18±0.01	-0.06±0.01
Basketball finer(2006)	-0.00±0.01	**-1.01±0.00**	-0.97±0.00	-0.42±0.07	-0.24±0.02	-0.17±0.01	-0.04±0.01
Basketball finer(2007)	-0.00±0.01	**-1.05±0.01**	-0.97±0.14	-0.46±0.06	-0.25±0.02	-0.18±0.02	-0.06±0.01
Basketball finer(2008)	-0.00±0.00	-0.94±0.00	**-0.98±0.00**	-0.42±0.09	-0.29±0.01	-0.21±0.01	-0.04±0.01
Basketball finer(2009)	-0.00±0.00	-0.96±0.00	**-1.00±0.04**	-0.45±0.04	-0.26±0.01	-0.21±0.02	-0.06±0.01
Basketball finer(2010)	-0.00±0.00	-0.93±0.00	**-0.98±0.02**	-0.41±0.07	-0.18±0.01	-0.21±0.01	-0.05±0.01
Basketball finer(2011)	-0.01±0.00	-0.88±0.00	**-0.99±0.01**	-0.44±0.06	-0.21±0.01	-0.22±0.01	-0.03±0.01
Basketball finer(2012)	0.00±0.00	-0.94±0.01	**-0.98±0.03**	-0.45±0.07	-0.26±0.01	-0.21±0.01	-0.03±0.00
Basketball finer(2013)	-0.00±0.01	-0.95±0.01	**-1.06±0.00**	-0.45±0.08	-0.27±0.01	-0.25±0.01	-0.07±0.01
Basketball finer(2014)	-0.01±0.00	-0.99±0.19	**-1.07±0.00**	-0.47±0.09	-0.24±0.02	-0.20±0.01	-0.02±0.01
ERO(p=0.05, style=uniform,η=0)	-0.05±0.02	**-1.71±0.01**	-1.64±0.01	-0.16±0.11	0.08±0.01	-0.00±0.01	-0.06±0.02
ERO(p=0.05, style=gamma,η=0)	-0.13±0.01	**-1.76±0.03**	-1.61±0.40	0.22±0.03	0.31±0.03	0.22±0.02	0.07±0.01
ERO(p=0.05, style=uniform,η=0.1)	-0.00±0.00	**-1.49±0.02**	-1.30±0.01	-0.05±0.04	-0.27±0.01	-0.24±0.01	-0.00±0.01
ERO(p=0.05, style=gamma,η=0.1)	-0.00±0.00	**-1.60±0.01**	-1.19±0.04	0.05±0.01	-0.05±0.03	0.02±0.03	0.02±0.01
ERO(p=0.05, style=uniform,η=0.2)	-0.01±0.02	**-1.37±0.01**	-1.21±0.04	-0.01±0.05	-0.22±0.04	-0.31±0.04	-0.02±0.03
ERO(p=0.05, style=gamma,η=0.2)	-0.00±0.01	**-1.43±0.02**	-1.14±0.02	0.00±0.01	-0.17±0.03	-0.14±0.02	-0.01±0.01
ERO(p=0.05, style=uniform,η=0.3)	-0.01±0.01	**-1.10±0.02**	-0.98±0.11	0.05±0.05	-0.14±0.05	-0.21±0.06	-0.01±0.03
ERO(p=0.05, style=gamma,η=0.3)	-0.03±0.01	**-1.16±0.02**	-1.05±0.02	-0.04±0.02	-0.29±0.01	-0.30±0.02	-0.05±0.01
ERO(p=0.05, style=uniform,η=0.4)	-0.00±0.00	**-1.00±0.05**	-0.75±0.03	0.04±0.05	-0.01±0.02	-0.09±0.04	0.02±0.01
ERO(p=0.05, style=gamma,η=0.4)	-0.11±0.03	**-0.98±0.09**	-0.98±0.09	-0.10±0.03	-0.31±0.06	-0.34±0.02	-0.17±0.15
ERO(p=0.05, style=uniform,η=0.5)	-0.00±0.00	**-0.84±0.01**	-0.64±0.02	0.02±0.15	-0.01±0.03	-0.04±0.05	-0.02±0.04
ERO(p=0.05, style=gamma,η=0.5)	-0.27±0.01	-0.84±0.04	**-0.87±0.13**	-0.25±0.04	-0.38±0.06	-0.44±0.02	-0.66±0.02
ERO(p=0.05, style=uniform,η=0.6)	-0.00±0.00	**-0.66±0.01**	-0.57±0.01	0.00±0.03	-0.00±0.02	-0.05±0.05	-0.18±0.05
ERO(p=0.05, style=gamma,η=0.6)	-0.39±0.02	-0.83±0.09	**-0.92±0.02**	-0.32±0.09	-0.53±0.01	-0.52±0.03	-0.76±0.09
ERO(p=0.05, style=uniform,η=0.7)	-0.01±0.03	**-0.64±0.02**	-0.56±0.26	-0.01±0.04	-0.14±0.27	-0.23±0.30	-0.48±0.27
ERO(p=0.05, style=gamma,η=0.7)	-0.42±0.00	-0.86±0.05	**-0.89±0.05**	-0.38±0.06	-0.59±0.02	-0.54±0.02	-0.87±0.04
ERO(p=0.05, style=uniform,η=0.8)	-0.24±0.20	-0.61±0.23	-0.69±0.23	-0.19±0.21	-0.26±0.28	-0.26±0.24	**-0.72±0.24**
ERO(p=0.05, style=gamma,η=0.8)	-0.51±0.00	-0.84±0.03	**-0.90±0.03**	-0.44±0.05	-0.61±0.03	-0.55±0.05	-0.88±0.09
ERO(p=1, style=uniform,η=0)	0.00±0.00	**-1.85±0.00**	0.00±0.00	-0.23±0.11	0.00±0.00	0.00±0.00	0.00±0.00
ERO(p=1, style=gamma,η=0)	0.00±0.00	**-1.83±0.00**	0.00±0.00	-0.09±0.01	0.00±0.00	0.00±0.00	-0.01±0.00
ERO(p=1, style=uniform,η=0.1)	-0.02±0.01	**-1.67±0.00**	0.00±0.00	-0.12±0.07	-0.09±0.02	-0.49±0.03	-0.00±0.00
ERO(p=1, style=gamma,η=0.1)	-0.01±0.01	**-1.58±0.00**	0.00±0.00	-0.03±0.01	-0.25±0.02	-0.51±0.02	-0.00±0.00
ERO(p=1, style=uniform,η=0.2)	-0.01±0.01	**-1.41±0.00**	-0.00±0.00	-0.11±0.05	-0.10±0.01	-0.34±0.02	-0.00±0.00
ERO(p=1, style=gamma,η=0.2)	-0.01±0.01	**-1.45±0.00**	-0.00±0.00	-0.01±0.02	-0.15±0.02	-0.32±0.03	-0.00±0.00
ERO(p=1, style=uniform,η=0.3)	-0.00±0.00	**-1.21±0.00**	0.00±0.00	-0.06±0.04	-0.07±0.01	-0.23±0.02	-0.00±0.00
ERO(p=1, style=gamma,η=0.3)	-0.00±0.00	**-1.25±0.00**	-0.00±0.00	-0.00±0.01	-0.12±0.03	-0.21±0.03	0.00±0.00
ERO(p=1, style=uniform,η=0.4)	-0.00±0.00	**-1.05±0.00**	-0.00±0.00	-0.05±0.04	-0.04±0.01	-0.14±0.02	-0.00±0.01
ERO(p=1, style=gamma,η=0.4)	-0.02±0.00	**-1.07±0.00**	0.00±0.00	-0.04±0.00	-0.15±0.00	-0.21±0.00	-0.00±0.00
ERO(p=1, style=uniform,η=0.5)	-0.13±0.00	**-0.94±0.00**	-0.06±0.00	-0.18±0.03	-0.19±0.00	-0.26±0.00	-0.11±0.00
ERO(p=1, style=gamma,η=0.5)	-0.21±0.00	**-0.98±0.00**	-0.07±0.00	-0.23±0.01	-0.30±0.00	-0.33±0.00	-0.19±0.00
ERO(p=1, style=uniform,η=0.6)	-0.32±0.00	**-0.94±0.00**	-0.27±0.00	-0.36±0.03	-0.37±0.00	-0.40±0.00	-0.31±0.00
ERO(p=1, style=gamma,η=0.6)	-0.38±0.00	**-0.93±0.00**	-0.27±0.00	-0.42±0.01	-0.44±0.00	-0.46±0.00	-0.37±0.00
ERO(p=1, style=uniform,η=0.7)	-0.51±0.00	**-0.95±0.00**	-0.49±0.00	-0.54±0.03	-0.53±0.00	-0.55±0.00	-0.51±0.00
ERO(p=1, style=gamma,η=0.7)	-0.55±0.00	**-0.98±0.00**	-0.52±0.00	-0.59±0.01	-0.59±0.00	-0.60±0.00	-0.56±0.00
ERO(p=1, style=uniform,η=0.8)	-0.68±0.00	**-0.97±0.00**	-0.95±0.00	-0.71±0.02	-0.70±0.00	-0.71±0.00	-0.70±0.00
ERO(p=1, style=gamma,η=0.8)	-0.71±0.00	-0.96±0.00	**-0.98±0.00**	-0.75±0.01	-0.73±0.00	-0.73±0.00	-0.73±0.00

Table D.16: Result table on $\mathcal{L}_{\text{upset, naive}}$ improvement with "proximal baseline" training starting from a baseline as initial guess, for individual directed graphs, averaged over 10 runs, and plus/minus one standard deviation. The best is marked in **bold red** while the second best is highlighted in underline blue .

Data	SpringRank	SyncRank	SerialRank	BTL	Eig.Cent.	PageRank	SVD_NRS
HeadToHead	-0.00±0.00	**-0.24±0.00**	-0.18±0.00	0.00±0.00	-0.05±0.00	-0.02±0.00	-0.04±0.00
Finance	-0.00±0.00	**-0.09±0.00**	0.00±0.00	-0.03±0.00	-0.02±0.00	-0.02±0.00	-0.00±0.00
Animal	-0.02±0.01	-0.34±0.06	**-0.44±0.12**	0.00±0.00	-0.01±0.00	-0.02±0.02	-0.00±0.02
Faculty: Business	-0.01±0.00	-0.12±0.00	**-0.17±0.01**	-0.00±0.00	-0.01±0.01	-0.01±0.01	-0.01±0.00
Faculty: CS	-0.00±0.00	-0.17±0.02	**-0.18±0.02**	0.00±0.00	0.00±0.01	-0.00±0.00	-0.02±0.01
Faculty: History	-0.00±0.00	-0.07±0.00	**-0.44±0.20**	0.00±0.00	-0.00±0.01	-0.00±0.00	-0.00±0.00
Football(2009)	0.00±0.00	**-0.18±0.11**	-0.04±0.02	-0.01±0.02	-0.03±0.01	-0.04±0.02	-0.00±0.00
Football(2010)	-0.01±0.01	**-0.15±0.04**	-0.08±0.01	-0.00±0.01	-0.03±0.02	-0.03±0.02	-0.00±0.00
Football(2011)	-0.00±0.00	**-0.19±0.09**	-0.01±0.01	-0.01±0.01	-0.01±0.01	-0.03±0.02	-0.00±0.01
Football(2012)	-0.01±0.01	**-0.17±0.01**	-0.12±0.03	-0.01±0.02	-0.00±0.01	-0.01±0.00	0.00±0.00
Football(2013)	-0.01±0.01	**-0.27±0.03**	-0.02±0.01	-0.02±0.01	-0.04±0.01	-0.04±0.01	-0.01±0.01
Football(2014)	-0.00±0.01	**-0.18±0.05**	-0.12±0.03	-0.01±0.01	-0.02±0.03	-0.00±0.00	-0.01±0.01
Football finer(2009)	-0.01±0.01	**-0.26±0.01**	-0.04±0.01	-0.01±0.03	-0.04±0.03	-0.06±0.03	-0.01±0.01
Football finer(2010)	-0.02±0.01	**-0.11±0.01**	-0.05±0.02	-0.03±0.03	-0.03±0.02	-0.06±0.01	-0.00±0.00
Football finer(2011)	-0.00±0.01	**-0.19±0.01**	-0.04±0.01	-0.01±0.01	-0.02±0.01	-0.05±0.01	-0.00±0.01
Football finer(2012)	-0.01±0.01	**-0.23±0.04**	-0.03±0.02	-0.01±0.01	-0.04±0.02	-0.07±0.01	-0.00±0.01
Football finer(2013)	-0.03±0.03	**-0.25±0.00**	-0.03±0.02	-0.02±0.02	-0.05±0.03	-0.06±0.02	-0.02±0.01
Football finer(2014)	-0.00±0.01	**-0.17±0.02**	-0.09±0.03	-0.02±0.02	-0.01±0.01	-0.02±0.01	-0.00±0.01
Basketball(1985)	-0.00±0.00	-0.23±0.00	**-0.32±0.01**	-0.03±0.02	-0.01±0.00	-0.02±0.01	-0.02±0.00
Basketball(1986)	-0.00±0.00	**-0.28±0.00**	**-0.28±0.01**	-0.03±0.01	-0.02±0.00	-0.01±0.00	-0.01±0.00
Basketball(1987)	-0.00±0.00	**-0.25±0.00**	-0.23±0.01	-0.03±0.01	-0.03±0.01	-0.00±0.00	-0.01±0.00
Basketball(1988)	0.00±0.00	-0.24±0.00	**-0.27±0.01**	-0.03±0.01	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(1989)	0.00±0.00	**-0.26±0.00**	-0.25±0.01	-0.03±0.00	0.00±0.00	-0.01±0.00	-0.02±0.00
Basketball(1990)	-0.00±0.00	-0.25±0.00	**-0.27±0.01**	-0.03±0.03	-0.02±0.00	-0.02±0.00	-0.01±0.00
Basketball(1991)	-0.00±0.00	-0.27±0.00	**-0.29±0.01**	-0.02±0.01	-0.01±0.00	-0.00±0.00	-0.02±0.00
Basketball(1992)	-0.00±0.00	**-0.27±0.00**	-0.26±0.00	-0.01±0.03	-0.01±0.00	0.00±0.00	-0.02±0.00
Basketball(1993)	-0.00±0.00	-0.25±0.00	**-0.29±0.02**	-0.02±0.02	-0.01±0.00	-0.02±0.00	-0.02±0.00
Basketball(1994)	0.00±0.00	-0.25±0.00	**-0.29±0.00**	-0.01±0.02	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(1995)	0.00±0.00	**-0.26±0.00**	-0.25±0.01	-0.01±0.02	-0.00±0.00	-0.02±0.00	-0.01±0.00
Basketball(1996)	-0.00±0.00	-0.23±0.00	**-0.28±0.01**	-0.02±0.02	-0.01±0.00	-0.01±0.00	-0.02±0.00
Basketball(1997)	-0.00±0.00	-0.24±0.00	**-0.25±0.01**	-0.03±0.01	-0.01±0.00	-0.01±0.00	-0.02±0.00
Basketball(1998)	-0.00±0.00	-0.24±0.00	**-0.25±0.01**	-0.03±0.01	-0.02±0.00	-0.01±0.00	-0.02±0.00
Basketball(1999)	-0.00±0.00	-0.21±0.00	**-0.27±0.01**	-0.02±0.02	-0.02±0.00	-0.02±0.00	-0.02±0.00
Basketball(2000)	-0.00±0.00	-0.25±0.00	**-0.26±0.01**	-0.02±0.01	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(2001)	0.00±0.00	-0.26±0.00	**-0.28±0.03**	-0.02±0.02	-0.01±0.00	-0.01±0.00	-0.03±0.01
Basketball(2002)	-0.00±0.00	-0.24±0.00	**-0.26±0.00**	-0.02±0.01	-0.00±0.00	-0.02±0.00	-0.01±0.00
Basketball(2003)	0.00±0.00	-0.27±0.00	**-0.28±0.00**	-0.02±0.02	-0.02±0.00	-0.02±0.00	-0.02±0.00
Basketball(2004)	-0.00±0.00	-0.25±0.00	**-0.28±0.03**	-0.02±0.01	-0.01±0.00	-0.02±0.00	-0.02±0.00
Basketball(2005)	0.00±0.00	**-0.26±0.00**	**-0.26±0.02**	-0.04±0.01	-0.00±0.00	-0.01±0.00	-0.02±0.00
Basketball(2006)	0.00±0.00	-0.26±0.00	**-0.27±0.01**	0.00±0.02	0.00±0.00	-0.01±0.00	-0.01±0.00
Basketball(2007)	-0.00±0.00	**-0.28±0.00**	-0.26±0.02	-0.02±0.01	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(2008)	0.00±0.00	-0.24±0.00	**-0.27±0.01**	-0.01±0.02	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(2009)	-0.00±0.00	-0.26±0.00	**-0.29±0.01**	-0.03±0.01	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(2010)	-0.00±0.00	-0.21±0.00	**-0.26±0.01**	-0.01±0.03	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(2011)	-0.00±0.00	-0.22±0.00	**-0.27±0.01**	0.00±0.02	-0.01±0.00	-0.01±0.00	-0.02±0.00
Basketball(2012)	-0.00±0.00	**-0.28±0.00**	**-0.28±0.01**	-0.02±0.02	-0.01±0.00	-0.01±0.00	-0.01±0.00
Basketball(2013)	-0.00±0.00	-0.24±0.00	**-0.27±0.02**	-0.03±0.01	-0.01±0.00	-0.01±0.00	-0.02±0.00
Basketball(2014)	-0.00±0.00	-0.24±0.00	**-0.28±0.01**	-0.00±0.03	-0.00±0.00	-0.01±0.00	-0.00±0.00
Basketball finer(1985)	-0.00±0.00	-0.23±0.00	**-0.24±0.02**	-0.14±0.02	-0.09±0.00	-0.08±0.00	-0.02±0.00
Basketball finer(1986)	0.00±0.00	**-0.28±0.00**	-0.25±0.01	-0.13±0.02	-0.08±0.00	-0.07±0.00	-0.01±0.00
Basketball finer(1987)	-0.00±0.00	**-0.26±0.00**	-0.23±0.00	-0.11±0.02	-0.06±0.00	-0.07±0.00	-0.01±0.00
Basketball finer(1988)	-0.00±0.00	**-0.27±0.00**	-0.25±0.00	-0.14±0.03	-0.09±0.00	-0.09±0.00	-0.01±0.00
Basketball finer(1989)	0.00±0.00	**-0.24±0.00**	-0.22±0.01	-0.13±0.01	-0.07±0.00	-0.07±0.00	-0.02±0.00
Basketball finer(1990)	-0.00±0.00	-0.24±0.00	**-0.25±0.00**	-0.14±0.01	-0.10±0.00	-0.08±0.00	-0.01±0.00
Basketball finer(1991)	-0.00±0.00	**-0.28±0.00**	-0.26±0.01	-0.11±0.02	-0.12±0.00	-0.10±0.00	-0.01±0.00
Basketball finer(1992)	-0.00±0.00	**-0.26±0.00**	-0.25±0.00	-0.12±0.01	-0.10±0.00	-0.09±0.00	-0.02±0.00
Basketball finer(1993)	-0.00±0.00	-0.24±0.00	**-0.27±0.02**	-0.12±0.01	-0.09±0.00	-0.07±0.00	-0.01±0.00
Basketball finer(1994)	0.00±0.00	-0.25±0.00	**-0.28±0.01**	-0.12±0.02	-0.10±0.00	-0.07±0.00	-0.01±0.00
Basketball finer(1995)	-0.00±0.00	**-0.26±0.00**	-0.25±0.00	-0.12±0.02	-0.09±0.00	-0.07±0.00	-0.02±0.00
Basketball finer(1996)	-0.00±0.00	**-0.22±0.00**	**-0.22±0.01**	-0.11±0.01	-0.07±0.01	-0.06±0.00	-0.01±0.00
Basketball finer(1997)	-0.00±0.00	**-0.26±0.00**	-0.24±0.00	-0.12±0.01	-0.07±0.00	-0.06±0.00	-0.02±0.00
Basketball finer(1998)	-0.00±0.00	**-0.24±0.00**	**-0.24±0.00**	-0.11±0.02	-0.06±0.00	-0.06±0.00	-0.02±0.00
Basketball finer(1999)	-0.00±0.00	-0.23±0.00	**-0.26±0.01**	-0.11±0.02	-0.07±0.00	-0.05±0.00	-0.02±0.00
Basketball finer(2000)	0.00±0.00	-0.24±0.00	**-0.25±0.00**	-0.10±0.01	-0.08±0.00	-0.06±0.00	-0.01±0.00
Basketball finer(2001)	-0.00±0.00	-0.24±0.00	**-0.25±0.01**	-0.11±0.01	-0.08±0.00	-0.06±0.00	-0.02±0.00
Basketball finer(2002)	-0.01±0.00	**-0.25±0.00**	-0.23±0.01	-0.13±0.02	-0.08±0.00	-0.06±0.00	-0.01±0.00
Basketball finer(2003)	0.00±0.00	**-0.25±0.00**	-0.22±0.02	-0.12±0.02	-0.06±0.00	-0.05±0.00	-0.01±0.00
Basketball finer(2004)	-0.00±0.00	**-0.25±0.01**	-0.23±0.00	-0.12±0.02	-0.07±0.00	-0.06±0.00	-0.02±0.00
Basketball finer(2005)	-0.00±0.00	**-0.27±0.00**	-0.22±0.00	-0.10±0.01	-0.05±0.00	-0.04±0.00	-0.02±0.00
Basketball finer(2006)	-0.00±0.00	**-0.25±0.00**	-0.24±0.00	-0.10±0.02	-0.06±0.00	-0.04±0.00	-0.01±0.00
Basketball finer(2007)	-0.00±0.00	**-0.26±0.00**	-0.23±0.03	-0.12±0.01	-0.06±0.00	-0.04±0.00	-0.01±0.00
Basketball finer(2008)	-0.00±0.00	-0.23±0.00	**-0.24±0.01**	-0.11±0.02	-0.07±0.00	-0.05±0.00	-0.01±0.00
Basketball finer(2009)	-0.00±0.00	-0.24±0.00	**-0.25±0.01**	-0.11±0.01	-0.07±0.00	-0.05±0.00	-0.01±0.00
Basketball finer(2010)	-0.00±0.00	-0.23±0.00	**-0.25±0.00**	-0.10±0.02	-0.05±0.00	-0.05±0.00	-0.01±0.00
Basketball finer(2011)	-0.00±0.00	-0.22±0.00	**-0.25±0.00**	-0.11±0.01	-0.05±0.00	-0.06±0.00	-0.01±0.00
Basketball finer(2012)	0.00±0.00	**-0.24±0.00**	**-0.24±0.01**	-0.11±0.02	-0.07±0.00	-0.05±0.00	-0.01±0.00
Basketball finer(2013)	-0.00±0.00	-0.24±0.00	**-0.26±0.02**	-0.11±0.02	-0.07±0.00	-0.06±0.00	-0.02±0.00
Basketball finer(2014)	-0.00±0.00	-0.25±0.05	**-0.26±0.01**	-0.12±0.02	-0.06±0.01	-0.05±0.00	-0.01±0.00
ERO(p=0.05, style=uniform,η=0)	-0.01±0.01	**-0.43±0.00**	-0.40±0.01	-0.04±0.03	0.02±0.00	-0.00±0.00	-0.01±0.00
ERO(p=0.05, style=gamma,η=0)	-0.03±0.00	**-0.44±0.01**	-0.40±0.10	0.06±0.01	0.08±0.01	0.05±0.01	0.02±0.00
ERO(p=0.05, style=uniform,η=0.1)	-0.00±0.00	**-0.37±0.00**	-0.32±0.00	-0.01±0.01	-0.06±0.00	-0.06±0.00	-0.00±0.00
ERO(p=0.05, style=gamma,η=0.1)	-0.00±0.00	**-0.40±0.00**	-0.26±0.02	0.01±0.00	-0.01±0.00	0.01±0.00	0.01±0.00
ERO(p=0.05, style=uniform,η=0.2)	-0.00±0.01	**-0.34±0.00**	-0.30±0.01	-0.00±0.01	-0.05±0.01	-0.08±0.01	-0.00±0.01
ERO(p=0.05, style=gamma,η=0.2)	-0.00±0.00	**-0.36±0.00**	-0.21±0.01	0.00±0.00	-0.02±0.00	-0.02±0.00	-0.00±0.00
ERO(p=0.05, style=uniform,η=0.3)	-0.00±0.00	**-0.28±0.01**	-0.25±0.03	0.01±0.01	-0.03±0.01	-0.05±0.02	-0.00±0.01
ERO(p=0.05, style=gamma,η=0.3)	-0.01±0.00	**-0.29±0.00**	-0.16±0.02	-0.00±0.00	-0.01±0.01	-0.01±0.00	-0.00±0.00
ERO(p=0.05, style=uniform,η=0.4)	-0.00±0.00	**-0.25±0.01**	-0.18±0.01	0.01±0.01	-0.00±0.00	-0.02±0.01	0.00±0.00
ERO(p=0.05, style=gamma,η=0.4)	-0.00±0.01	**-0.24±0.01**	-0.11±0.03	0.00±0.01	-0.01±0.00	-0.01±0.00	-0.00±0.00
ERO(p=0.05, style=uniform,η=0.5)	-0.00±0.00	**-0.21±0.00**	-0.16±0.00	0.01±0.01	-0.00±0.00	-0.01±0.01	-0.00±0.01
ERO(p=0.05, style=gamma,η=0.5)	-0.01±0.01	**-0.21±0.01**	-0.06±0.04	-0.01±0.01	-0.01±0.00	-0.00±0.00	-0.03±0.03
ERO(p=0.05, style=uniform,η=0.6)	-0.00±0.00	**-0.17±0.00**	-0.14±0.00	0.00±0.01	-0.00±0.00	-0.01±0.01	-0.04±0.01
ERO(p=0.05, style=gamma,η=0.6)	-0.01±0.01	**-0.18±0.01**	-0.04±0.00	-0.01±0.01	-0.01±0.01	-0.01±0.01	-0.01±0.01
ERO(p=0.05, style=uniform,η=0.7)	-0.00±0.01	**-0.16±0.00**	-0.09±0.00	-0.00±0.01	0.00±0.00	-0.00±0.00	-0.06±0.00
ERO(p=0.05, style=gamma,η=0.7)	-0.00±0.01	**-0.16±0.01**	-0.05±0.01	-0.01±0.01	-0.01±0.01	0.00±0.00	-0.02±0.01
ERO(p=0.05, style=uniform,η=0.8)	-0.00±0.00	**-0.14±0.00**	-0.07±0.00	0.00±0.01	-0.00±0.00	0.00±0.00	-0.07±0.00
ERO(p=0.05, style=gamma,η=0.8)	-0.01±0.01	**-0.14±0.01**	-0.04±0.01	-0.01±0.01	-0.01±0.01	0.00±0.01	-0.03±0.02
ERO(p=1, style=uniform,η=0)	0.00±0.00	**-0.46±0.00**	0.00±0.00	-0.06±0.03	0.00±0.00	0.00±0.00	0.00±0.00
ERO(p=1, style=gamma,η=0)	0.00±0.00	**-0.46±0.00**	0.00±0.00	-0.02±0.00	0.00±0.00	0.00±0.00	-0.00±0.00
ERO(p=1, style=uniform,η=0.1)	-0.00±0.00	**-0.42±0.00**	0.00±0.00	-0.03±0.02	-0.02±0.01	-0.12±0.01	-0.00±0.00
ERO(p=1, style=gamma,η=0.1)	-0.00±0.00	**-0.39±0.00**	0.00±0.00	0.01±0.00	-0.05±0.02	-0.11±0.01	-0.00±0.00
ERO(p=1, style=uniform,η=0.2)	-0.00±0.00	**-0.35±0.00**	-0.00±0.00	-0.03±0.01	-0.03±0.00	-0.09±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.2)	-0.00±0.00	**-0.36±0.00**	-0.00±0.00	-0.00±0.00	-0.04±0.01	-0.08±0.01	-0.00±0.00
ERO(p=1, style=uniform,η=0.3)	-0.00±0.00	**-0.30±0.00**	0.00±0.00	-0.02±0.01	-0.02±0.00	-0.06±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.3)	-0.00±0.00	**-0.31±0.00**	-0.00±0.00	-0.00±0.00	-0.03±0.01	-0.05±0.01	0.00±0.00
ERO(p=1, style=uniform,η=0.4)	-0.00±0.00	**-0.26±0.00**	-0.00±0.00	-0.01±0.01	-0.01±0.00	-0.04±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.4)	-0.00±0.00	**-0.27±0.00**	0.00±0.00	-0.00±0.00	-0.02±0.01	-0.03±0.01	0.00±0.00
ERO(p=1, style=uniform,η=0.5)	-0.00±0.00	**-0.22±0.00**	-0.00±0.00	-0.01±0.01	-0.01±0.00	-0.03±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.5)	-0.00±0.00	**-0.22±0.00**	-0.00±0.00	0.00±0.01	-0.01±0.01	-0.01±0.01	0.00±0.00
ERO(p=1, style=uniform,η=0.6)	-0.00±0.00	**-0.17±0.00**	-0.00±0.00	-0.01±0.01	-0.01±0.00	-0.02±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.6)	-0.00±0.00	**-0.16±0.00**	-0.00±0.00	-0.00±0.00	-0.01±0.00	-0.01±0.00	-0.00±0.00
ERO(p=1, style=uniform,η=0.7)	-0.01±0.00	**-0.12±0.00**	-0.00±0.00	-0.00±0.01	-0.00±0.00	-0.00±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.7)	-0.01±0.00	**-0.13±0.00**	0.00±0.00	-0.00±0.00	-0.00±0.00	-0.00±0.00	-0.00±0.00
ERO(p=1, style=uniform,η=0.8)	-0.00±0.00	**-0.08±0.00**	-0.05±0.00	-0.00±0.01	-0.00±0.00	-0.00±0.00	-0.00±0.00
ERO(p=1, style=gamma,η=0.8)	-0.00±0.00	**-0.08±0.00**	-0.05±0.00	-0.00±0.00	-0.00±0.00	0.00±0.00	-0.00±0.00

Table D.17: GNN selection among GNNRank-N methods for the lowest $\mathcal{L}_{\text{upset, simple}}$.

Data	Variant	train with	pretrain with	$\mathcal{L}_{\text{upset,margin}}$ coefficient	baseline	$\mathcal{L}_{\text{upset,ratio}}$ coefficient
HeadToHead	ib dist	proximal baseline	innerproduct	1	SVD_NRS	1
Finance	ib dist	innerproduct	–	0	–	1
Animal	DIMPA dist	dist	–	0	–	1
Faculty: Business	DIMPA innerproduct	innerproduct	–	0	–	1
Faculty: CS	DIMPA dist	dist	–	0	–	1
Faculty: History	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1985)	DIMPA dist	dist	–	0	–	1
Basketball(1986)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1987)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1988)	DIMPA dist	dist	–	0	–	1
Basketball(1989)	DIMPA dist	dist	–	0	–	1
Basketball(1990)	DIMPA dist	dist	–	0	–	1
Basketball(1991)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1992)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1993)	DIMPA dist	dist	–	1	–	1
Basketball(1994)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1995)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1996)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1997)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1998)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1999)	DIMPA dist	dist	–	0	–	1
Basketball(2000)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2001)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2002)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2003)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2004)	DIMPA dist	dist	–	0	–	1
Basketball(2005)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2006)	DIMPA dist	dist	–	0	–	1
Basketball(2007)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2008)	DIMPA dist	dist	–	0	–	1
Basketball(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2010)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2011)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2012)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2013)	DIMPA dist	dist	–	0	–	1
Basketball(2014)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1985)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1986)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1987)	DIMPA dist	dist	–	1	–	1
Basketball finer(1988)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1989)	DIMPA dist	dist	–	1	–	1
Basketball finer(1990)	DIMPA dist	dist	–	1	–	1
Basketball finer(1991)	DIMPA dist	dist	–	1	–	0
Basketball finer(1992)	DIMPA dist	dist	–	1	–	1
Basketball finer(1993)	DIMPA dist	dist	–	1	–	0
Basketball finer(1994)	DIMPA dist	dist	–	1	–	1
Basketball finer(1995)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1996)	DIMPA dist	dist	–	1	–	1
Basketball finer(1997)	DIMPA dist	dist	–	1	–	1
Basketball finer(1998)	DIMPA dist	dist	–	1	–	0
Basketball finer(1999)	DIMPA dist	dist	–	1	–	1
Basketball finer(2000)	DIMPA dist	dist	–	1	–	1
Basketball finer(2001)	DIMPA dist	dist	–	1	–	1
Basketball finer(2002)	DIMPA dist	dist	–	1	–	1
Basketball finer(2003)	DIMPA dist	dist	–	1	–	1
Basketball finer(2004)	DIMPA dist	dist	–	1	–	0
Basketball finer(2005)	DIMPA dist	dist	–	1	–	1
Basketball finer(2006)	DIMPA dist	dist	–	1	–	1
Basketball finer(2007)	DIMPA dist	dist	–	1	–	0
Basketball finer(2008)	DIMPA dist	dist	–	1	–	1
Basketball finer(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2010)	DIMPA dist	dist	–	1	–	1
Basketball finer(2011)	DIMPA dist	dist	–	1	–	1
Basketball finer(2012)	DIMPA dist	dist	–	1	–	0
Basketball finer(2013)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2014)	DIMPA innerproduct	innerproduct	–	0	–	1
Football(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Football(2010)	ib dist	proximal innerproduct	dist	1	–	1
Football(2011)	ib dist	proximal baseline	dist	1	BTL	0
Football(2012)	ib dist	dist	–	1	–	0
Football(2013)	DIMPA dist	dist	–	0	–	1
Football(2014)	ib dist	proximal baseline	innerproduct	1	Eign.Cent.	1
Football finer(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Football finer(2010)	ib innerproduct	proximal innerproduct	innerproduct	1	–	0
Football finer(2011)	DIMPA dist	dist	–	1	–	0
Football finer(2012)	DIMPA dist	proximal baseline	dist	1	BTL	1
Football finer(2013)	DIMPA innerproduct	innerproduct	–	0	–	1
Football finer(2014)	ib dist	proximal innerproduct	SerialRank similarity	1	–	1

Table D.18: GNN selection among GNNRank-P methods for the lowest $\mathcal{L}_{\text{upset, simple}}$.

Data	Variant	train with	pretrain with	$\mathcal{L}_{\text{upset,margin}}$ coefficient	baseline	$\mathcal{L}_{\text{upset,ratio}}$ coefficient
HeadToHead	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Finance	ib proximal dist	innerproduct	–	0	–	1
Animal	ib proximal baseline	proximal baseline	innerproduct	1	SyncRank	0
Faculty: Business	ib proximal baseline	proximal baseline	innerproduct	0	SyncRank	1
Faculty: CS	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Faculty: History	DIMPA proximal dist	dist	–	1	–	1
Basketball(1985)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1986)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1987)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1
Basketball(1988)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1989)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball(1990)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1991)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1992)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1993)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball(1994)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1995)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(1996)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball(1997)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1
Basketball(1998)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	0
Basketball(1999)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2000)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2001)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2002)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2003)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2004)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2005)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball(2006)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(2007)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(2008)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2009)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball(2010)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2011)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2012)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2013)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(2014)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(1985)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1986)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1987)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(1988)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(1989)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1990)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1991)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1992)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1993)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(1994)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1995)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball finer(1996)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1997)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball finer(1998)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball finer(1999)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2000)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2001)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2002)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(2003)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2004)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	1
Basketball finer(2005)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(2006)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2007)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(2008)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2009)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2010)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2011)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2012)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(2013)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball finer(2014)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	1
Football(2009)	ib proximal baseline	dist	–	0	SpringRank	1
Football(2010)	ib proximal innerproduct	proximal baseline	innerproduct	1	SpringRank	1
Football(2011)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SpringRank	1
Football(2012)	ib proximal dist	proximal baseline	SerialRank similarity	1	PageRank	0
Football(2013)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Football(2014)	ib proximal dist	proximal baseline	innerproduct	1	PageRank	0
Football finer(2009)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SVD_NRS	1
Football finer(2010)	ib proximal innerproduct	proximal baseline	innerproduct	1	SerialRank	1
Football finer(2011)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SVD_NRS	0
Football finer(2012)	ib proximal baseline	proximal baseline	dist	1	SVD_NRS	0
Football finer(2013)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Football finer(2014)	ib proximal dist	proximal baseline	innerproduct	0	SVD_NRS	1

Table D.19: GNN selection among GNNRank-N methods for the lowest $\mathcal{L}_{\text{upset, naive}}$.

Data	Variant	train with	pretrain with	$\mathcal{L}_{\text{upset,margin}}$ coefficient	baseline	$\mathcal{L}_{\text{upset,ratio}}$ coefficient
HeadToHead	DIMPA dist	dist	–	0	–	1
Finance	DIMPA innerproduct	innerproduct	–	1	–	0
Animal	DIMPA dist	dist	–	0	–	1
Faculty: Business	DIMPA innerproduct	innerproduct	–	0	–	1
Faculty: CS	DIMPA dist	dist	–	0	–	1
Faculty: History	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1985)	DIMPA dist	dist	–	0	–	1
Basketball(1986)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1987)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1988)	DIMPA dist	dist	–	0	–	1
Basketball(1989)	DIMPA dist	dist	–	0	–	1
Basketball(1990)	DIMPA dist	dist	–	0	–	1
Basketball(1991)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1992)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1993)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1994)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1995)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1996)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1997)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1998)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1999)	DIMPA dist	dist	–	0	–	1
Basketball(2000)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2001)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2002)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2003)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2004)	DIMPA dist	dist	–	0	–	1
Basketball(2005)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2006)	DIMPA dist	dist	–	0	–	1
Basketball(2007)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2008)	DIMPA dist	dist	–	0	–	1
Basketball(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2010)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2011)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2012)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2013)	DIMPA dist	dist	–	0	–	1
Basketball(2014)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1985)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1986)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1987)	DIMPA dist	dist	–	1	–	1
Basketball finer(1988)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1989)	DIMPA dist	dist	–	1	–	1
Basketball finer(1990)	DIMPA dist	dist	–	1	–	1
Basketball finer(1991)	DIMPA dist	dist	–	1	–	0
Basketball finer(1992)	DIMPA dist	dist	–	1	–	1
Basketball finer(1993)	DIMPA dist	dist	–	1	–	0
Basketball finer(1994)	DIMPA dist	dist	–	1	–	1
Basketball finer(1995)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1996)	DIMPA dist	dist	–	1	–	1
Basketball finer(1997)	DIMPA dist	dist	–	1	–	1
Basketball finer(1998)	DIMPA dist	dist	–	1	–	0
Basketball finer(1999)	DIMPA dist	dist	–	1	–	1
Basketball finer(2000)	DIMPA dist	dist	–	1	–	1
Basketball finer(2001)	DIMPA dist	dist	–	1	–	1
Basketball finer(2002)	DIMPA dist	dist	–	1	–	1
Basketball finer(2003)	DIMPA dist	dist	–	1	–	1
Basketball finer(2004)	DIMPA dist	dist	–	1	–	0
Basketball finer(2005)	DIMPA dist	dist	–	1	–	1
Basketball finer(2006)	DIMPA dist	dist	–	1	–	1
Basketball finer(2007)	DIMPA dist	dist	–	1	–	0
Basketball finer(2008)	DIMPA dist	dist	–	1	–	1
Basketball finer(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2010)	DIMPA dist	dist	–	1	–	1
Basketball finer(2011)	DIMPA dist	dist	–	1	–	1
Basketball finer(2012)	DIMPA dist	dist	–	1	–	0
Basketball finer(2013)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2014)	DIMPA innerproduct	innerproduct	–	0	–	1
Football(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Football(2010)	DIMPA dist	proximal dist	dist	1	–	1
Football(2011)	DIMPA dist	dist	–	0	–	1
Football(2012)	DIMPA dist	dist	–	1	–	1
Football(2013)	DIMPA dist	dist	–	0	–	1
Football(2014)	DIMPA dist	dist	–	1	–	1
Football finer(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Football finer(2010)	DIMPA dist	proximal dist	dist	0	–	1
Football finer(2011)	DIMPA dist	dist	–	1	–	0
Football finer(2012)	DIMPA dist	innerproduct	–	1	–	1
Football finer(2013)	DIMPA innerproduct	innerproduct	–	0	–	1
Football finer(2014)	DIMPA dist	dist	–	1	–	1
ERO(p=0.05, style=uniform,η=0)	DIMPA dist	dist	–	0	–	1
ERO(p=0.05, style=gamma,η=0)	DIMPA dist	dist	–	0	–	1
ERO(p=0.05, style=uniform,η=0.1)	ib innerproduct	innerproduct	–	0	–	1
ERO(p=0.05, style=gamma,η=0.1)	DIMPA dist	dist	–	0	–	1
ERO(p=0.05, style=uniform,η=0.2)	DIMPA dist	dist	–	1	–	1
ERO(p=0.05, style=gamma,η=0.2)	ib innerproduct	proximal baseline	innerproduct	0	SpringRank	1
ERO(p=0.05, style=uniform,η=0.3)	ib dist	dist	–	1	–	1
ERO(p=0.05, style=gamma,η=0.3)	ib innerproduct	proximal baseline	innerproduct	1	SyncRank	1
ERO(p=0.05, style=uniform,η=0.4)	ib innerproduct	innerproduct	–	1	–	0
ERO(p=0.05, style=gamma,η=0.4)	ib innerproduct	proximal baseline	innerproduct	1	SyncRank	1
ERO(p=0.05, style=uniform,η=0.5)	ib innerproduct	innerproduct	–	1	–	0
ERO(p=0.05, style=gamma,η=0.5)	ib innerproduct	proximal innerproduct	innerproduct	1	–	1
ERO(p=0.05, style=uniform,η=0.6)	ib innerproduct	innerproduct	–	1	–	0
ERO(p=0.05, style=gamma,η=0.6)	ib innerproduct	innerproduct	–	0	–	1
ERO(p=0.05, style=uniform,η=0.7)	ib innerproduct	innerproduct	–	1	–	0
ERO(p=0.05, style=gamma,η=0.7)	ib innerproduct	innerproduct	–	0	–	1
ERO(p=0.05, style=uniform,η=0.8)	ib innerproduct	innerproduct	–	1	–	0
ERO(p=0.05, style=gamma,η=0.8)	ib innerproduct	innerproduct	–	0	–	1
ERO(p=1, style=uniform,η=0)	ib dist	dist	–	1	–	0
ERO(p=1, style=gamma,η=0)	ib dist	dist	–	1	–	0
ERO(p=1, style=uniform,η=0.1)	DIMPA innerproduct	proximal baseline	innerproduct	0	SVD_NRS	1
ERO(p=1, style=gamma,η=0.1)	DIMPA dist	dist	–	0	–	1
ERO(p=1, style=uniform,η=0.2)	DIMPA dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.2)	DIMPA dist	dist	–	0	–	1
ERO(p=1, style=uniform,η=0.3)	DIMPA dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.3)	DIMPA dist	dist	–	1	–	1
ERO(p=1, style=uniform,η=0.4)	DIMPA innerproduct	innerproduct	–	0	–	1
ERO(p=1, style=gamma,η=0.4)	DIMPA dist	dist	–	1	–	1
ERO(p=1, style=uniform,η=0.5)	ib dist	innerproduct	–	0	–	1
ERO(p=1, style=gamma,η=0.5)	ib dist	dist	–	1	–	1
ERO(p=1, style=uniform,η=0.6)	ib dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.6)	ib dist	dist	–	1	–	0
ERO(p=1, style=uniform,η=0.7)	ib dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.7)	ib dist	dist	–	0	–	1
ERO(p=1, style=uniform,η=0.8)	ib dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.8)	ib dist	dist	–	0	–	1

Table D.20: GNN selection among GNNRank-P methods for the lowest $\mathcal{L}_{\text{upset, naive}}$.

Data	Variant	train with	pretrain with	$\mathcal{L}_{\text{upset,margin}}$ coefficient	baseline	$\mathcal{L}_{\text{upset,ratio}}$ coefficient
HeadToHead	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Finance	ib proximal baseline	proximal baseline	SerialRank similarity	0	SerialRank	1
Animal	ib proximal baseline	proximal baseline	innerproduct	1	SyncRank	0
Faculty: Business	ib proximal baseline	proximal baseline	innerproduct	0	SyncRank	1
Faculty: CS	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Faculty: History	DIMPA proximal dist	dist	–	1	–	1
Basketball(1985)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1986)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1987)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1
Basketball(1988)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1989)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball(1990)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1991)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1992)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1993)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball(1994)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(1995)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(1996)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball(1997)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1
Basketball(1998)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	0
Basketball(1999)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2000)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2001)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2002)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2003)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2004)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2005)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball(2006)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(2007)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(2008)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2009)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball(2010)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2011)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2012)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball(2013)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball(2014)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(1985)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1986)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1987)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball finer(1988)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(1989)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1990)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1991)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1992)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1993)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(1994)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1995)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball finer(1996)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(1997)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball finer(1998)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
Basketball finer(1999)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2000)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2001)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2002)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(2003)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2004)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	1
Basketball finer(2005)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(2006)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2007)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
Basketball finer(2008)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2009)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2010)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2011)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Basketball finer(2012)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
Basketball finer(2013)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0
Basketball finer(2014)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	1
Football(2009)	ib proximal baseline	dist	–	0	SpringRank	1
Football(2010)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SVD_NRS	1
Football(2011)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SpringRank	1
Football(2012)	ib proximal baseline	proximal baseline	dist	1	SyncRank	0
Football(2013)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Football(2014)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SVD_NRS	1
Football finer(2009)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SVD_NRS	1
Football finer(2010)	ib proximal baseline	proximal baseline	dist	1	SVD_NRS	0
Football finer(2011)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SVD_NRS	0
Football finer(2012)	ib proximal baseline	proximal baseline	dist	1	SVD_NRS	0
Football finer(2013)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
Football finer(2014)	DIMPA proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1
ERO(p=0.05, style=uniform,η=0)	ib proximal baseline	proximal baseline	innerproduct	1	SyncRank	1
ERO(p=0.05, style=gamma,η=0)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=uniform,η=0.1)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=gamma,η=0.1)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=uniform,η=0.2)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=gamma,η=0.2)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=uniform,η=0.3)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=gamma,η=0.3)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
ERO(p=0.05, style=uniform,η=0.4)	DIMPA proximal baseline	proximal baseline	innerproduct	1	SyncRank	0
ERO(p=0.05, style=gamma,η=0.4)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1
ERO(p=0.05, style=uniform,η=0.5)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
ERO(p=0.05, style=gamma,η=0.5)	ib proximal innerproduct	innerproduct	–	1	–	1
ERO(p=0.05, style=uniform,η=0.6)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1
ERO(p=0.05, style=gamma,η=0.6)	ib proximal dist	innerproduct	–	0	–	1
ERO(p=0.05, style=uniform,η=0.7)	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	0
ERO(p=0.05, style=gamma,η=0.7)	ib proximal innerproduct	innerproduct	–	0	–	1
ERO(p=0.05, style=uniform,η=0.8)	ib proximal dist	proximal baseline	dist	1	SpringRank	1
ERO(p=0.05, style=gamma,η=0.8)	ib proximal innerproduct	innerproduct	–	0	–	1
ERO(p=1, style=uniform,η=0)	ib proximal baseline	proximal baseline	innerproduct	1	SerialRank	0
ERO(p=1, style=gamma,η=0)	ib proximal baseline	proximal baseline	dist	0	SerialRank	1
ERO(p=1, style=uniform,η=0.1)	ib proximal baseline	proximal baseline	innerproduct	0	SerialRank	1
ERO(p=1, style=gamma,η=0.1)	ib proximal baseline	proximal baseline	innerproduct	0	SerialRank	1
ERO(p=1, style=uniform,η=0.2)	ib proximal baseline	proximal baseline	innerproduct	0	SerialRank	1
ERO(p=1, style=gamma,η=0.2)	ib proximal baseline	proximal baseline	innerproduct	1	SerialRank	0
ERO(p=1, style=uniform,η=0.3)	ib proximal baseline	proximal baseline	innerproduct	0	SerialRank	1
ERO(p=1, style=gamma,η=0.3)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SerialRank	0
ERO(p=1, style=uniform,η=0.4)	ib proximal baseline	proximal baseline	innerproduct	1	SerialRank	0
ERO(p=1, style=gamma,η=0.4)	ib proximal baseline	proximal baseline	innerproduct	0	SerialRank	1
ERO(p=1, style=uniform,η=0.5)	ib proximal dist	innerproduct	–	0	–	1
ERO(p=1, style=gamma,η=0.5)	ib proximal dist	dist	–	1	–	1
ERO(p=1, style=uniform,η=0.6)	ib proximal dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.6)	ib proximal dist	dist	–	1	–	0
ERO(p=1, style=uniform,η=0.7)	ib proximal dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.7)	ib proximal dist	dist	–	0	–	1
ERO(p=1, style=uniform,η=0.8)	ib proximal dist	dist	–	0	–	1
ERO(p=1, style=gamma,η=0.8)	ib proximal dist	dist	–	0	–	1

Table D.21: GNN selection among GNNRank-N methods for the lowest $\mathcal{L}_{\text{upset, ratio}}$.

Data	Variant	train with	pretrain with	$\mathcal{L}_{\text{upset,margin}}$ coefficient	baseline	$\mathcal{L}_{\text{upset,ratio}}$ coefficient
HeadToHead	DIMPA dist	dist	–	0	–	1
Finance	DIMPA innerproduct	innerproduct	–	0	–	1
Animal	DIMPA dist	dist	–	0	–	1
Faculty: Business	DIMPA innerproduct	innerproduct	–	0	–	1
Faculty: CS	DIMPA dist	dist	–	0	–	1
Faculty: History	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1985)	DIMPA dist	dist	–	0	–	1
Basketball(1986)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1987)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1988)	DIMPA dist	dist	–	0	–	1
Basketball(1989)	DIMPA dist	dist	–	0	–	1
Basketball(1990)	DIMPA dist	dist	–	0	–	1
Basketball(1991)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1992)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1993)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1994)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1995)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1996)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1997)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1998)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(1999)	DIMPA dist	dist	–	0	–	1
Basketball(2000)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2001)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2002)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2003)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2004)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2005)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2006)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2007)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2008)	DIMPA dist	dist	–	0	–	1
Basketball(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2010)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2011)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2012)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball(2013)	DIMPA dist	dist	–	0	–	1
Basketball(2014)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1985)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1986)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1987)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1988)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1989)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1990)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1991)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1992)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1993)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1994)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1995)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1996)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1997)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1998)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(1999)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2000)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2001)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2002)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2003)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2004)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2005)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2006)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2007)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2008)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2009)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2010)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2011)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2012)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2013)	DIMPA innerproduct	innerproduct	–	0	–	1
Basketball finer(2014)	DIMPA innerproduct	innerproduct	–	0	–	1
Football(2009)	DIMPA dist	dist	–	1	–	1
Football(2010)	DIMPA dist	dist	–	1	–	1
Football(2011)	DIMPA dist	dist	–	1	–	1
Football(2012)	DIMPA dist	dist	–	1	–	1
Football(2013)	DIMPA dist	dist	–	1	–	1
Football(2014)	DIMPA dist	dist	–	1	–	1
Football finer(2009)	DIMPA dist	dist	–	1	–	1
Football finer(2010)	DIMPA innerproduct	innerproduct	–	0	–	1
Football finer(2011)	DIMPA dist	dist	–	1	–	1
Football finer(2012)	DIMPA dist	dist	–	1	–	1
Football finer(2013)	DIMPA innerproduct	innerproduct	–	0	–	1
Football finer(2014)	DIMPA dist	dist	–	1	–	1

Table D.22: GNN selection among GNNRank-P methods for the lowest $\mathcal{L}_{\text{upset, ratio}}$.

Data	Variant	learning rate	train with	pretrain with	$\mathcal{L}_{\text{upset,margin}}$ coefficient	baseline	$\mathcal{L}_{\text{upset,ratio}}$ coefficient
HeadToHead	DIMPA proximal baseline	proximal baseline	dist	1	SyncRank	1	
Finance	DIMPA proximal baseline	proximal innerproduct	dist	1	SpringRank	1	
Animal	ib proximal baseline	proximal baseline	SerialRank similarity	0	SpringRank	1	
Faculty: Business	DIMPA proximal dist	proximal baseline	SerialRank similarity	0	SerialRank	1	
Faculty: CS	DIMPA proximal baseline	proximal baseline	dist	1	PageRank	0	
Faculty: History	DIMPA proximal baseline	proximal baseline	dist	1	BTL	0	
Basketball(1985)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1986)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1987)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1988)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1	
Basketball(1989)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1990)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1	
Basketball(1991)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1992)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SVD_NRS	0	
Basketball(1993)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1	
Basketball(1994)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1995)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1996)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1997)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(1998)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0	
Basketball(1999)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2000)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2001)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2002)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2003)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2004)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2005)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2006)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1	
Basketball(2007)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	0	
Basketball(2008)	ib proximal baseline	proximal baseline	dist	0	SyncRank	1	
Basketball(2009)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1	
Basketball(2010)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2011)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2012)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1	
Basketball(2013)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball(2014)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Basketball finer(1985)	ib proximal baseline	dist	–	0	SpringRank	1	
Basketball finer(1986)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1987)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1988)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1989)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1990)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1991)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(1992)	ib proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1993)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1994)	ib proximal baseline	dist	–	1	SpringRank	0	
Basketball finer(1995)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1996)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(1997)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(1998)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(1999)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(2000)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(2001)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(2002)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2003)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2004)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2005)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2006)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2007)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2008)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2009)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2010)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2011)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2012)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Basketball finer(2013)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Basketball finer(2014)	DIMPA proximal baseline	innerproduct	–	0	SpringRank	1	
Football(2009)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Football(2010)	DIMPA proximal baseline	proximal innerproduct	dist	1	SpringRank	1	
Football(2011)	ib proximal baseline	proximal baseline	dist	1	SyncRank	0	
Football(2012)	ib proximal baseline	proximal baseline	SerialRank similarity	1	SyncRank	1	
Football(2013)	ib proximal baseline	proximal baseline	SerialRank similarity	0	SyncRank	1	
Football(2014)	ib proximal baseline	proximal baseline	dist	1	SyncRank	0	
Football finer(2009)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Football finer(2010)	DIMPA proximal baseline	dist	–	1	SpringRank	1	
Football finer(2011)	DIMPA proximal baseline	proximal innerproduct	dist	1	SpringRank	1	
Football finer(2012)	DIMPA proximal baseline	proximal innerproduct	dist	0	SpringRank	1	
Football finer(2013)	DIMPA proximal baseline	proximal innerproduct	dist	0	SpringRank	1	
Football finer(2014)	DIMPA proximal dist	dist	–	1	–	1	

E

Robust Angular Synchronization Using Directed Graph Neural Networks Supplementary Information

E.1 Analytical discussions

E.1.1 Properties of the loss functions

In classical convex optimization of the type $\inf_{\mathbf{x}} g(x)$, optimal values are achieved at stationary points; when the function g is differentiable, then following Fermat's rule, stationary points are points at which the gradient of g vanishes. Such points are typically found via gradient descent methods. When the function g is not differentiable, then there are weaker variants for differentiablity available such as the directional derivative. First we recall the notion of a directional derivative and a directional stationary point from [207]. The *directional derivative* of a function f at point $\mathbf{x} \in \mathbb{R}^m$ in the direction $\mathbf{d} \in \mathbb{R}^m$ is defined by

$$f'(\mathbf{x}, \mathbf{d}) = \lim_{t \searrow 0} \frac{f(\mathbf{x} + t\mathbf{d}) - f(\mathbf{x})}{t}.$$

A *directional stationary* point $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ of the problem $\inf_{\mathbf{x} \in C} g(x)$ for $C \subset \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}$ is a point such that the directional derivatives in any direction $\mathbf{d} \in \mathbb{R}^n$ satisfy $(g + \mathbb{1}_C)'(\mathbf{x}, \mathbf{d}) \geq 0$. This notion is broad enough to include functions such as the maximum which is not everywhere differentiable.

Moreover we say that a function $f : \mathbb{R}^m \to \mathbb{R}$ is *locally Lipschitz* if for any bounded set $\mathcal{S} \subset \mathbb{R}^m$, there exists a constant $L > 0$ such that

$$|f(\mathbf{x} - \mathbf{y})| \leq L \parallel \mathbf{x} - \mathbf{y} \parallel_2$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{S}$. Note that a locally Lipschitz function f is differentiable almost everywhere, see for example [231, Theorem 9.60] where also more background on directional derivatives and subdifferentials can be found.

Proof of Proposition 2 Here we prove Proposition 2 from the main text; for convenience, we repeat it here.

Proposition 5. *Every local minimum of eq.* (6.1) *is a directional stationary point of eq.* (6.2).

Proof. eq. (6.2) gives that

$$\begin{aligned}
&\mathcal{L}_{\text{upset}} \\
&= \|\mathbf{M}\|_F / t \\
&= \frac{1}{t}\sqrt{\sum_{i,j} \mathbb{1}(\mathbf{T}_{i,j} - \mathbf{A}_{i,j} \neq 0, 2\pi) \min(\mathbf{T}_{i,j} - \mathbf{A}_{i,j} \bmod 2\pi, \mathbf{A}_{i,j} - \mathbf{T}_{i,j} \bmod 2\pi)^2} \\
&= \frac{1}{t}\sqrt{\sum_{i,j} \mathbb{1}(\mathbf{T}_{i,j} - \mathbf{A}_{i,j} \neq 0, 2\pi) \min\{\mathbf{T}_{i,j} - \mathbf{A}_{i,j} \bmod 2\pi, 2\pi - (\mathbf{T}_{i,j} - \mathbf{A}_{i,j} \bmod 2\pi)\}^2}
\end{aligned}$$

where t is the number of nonzero elements in $\mathbf{A}$. The function $f : (0, 2\pi) \mapsto (0, 2\pi)$ given by $f(x) = \min(x^2, (2\pi - x)^2)$ is differentiable with derivative uniformly bounded by 4π (and is hence locally Lipschitz) except at the point $x = \pi$ where it takes on its maximum, π^2. Thus, this function is a directionally differentiable Lipschitz function (which can be seen by writing $\min(a, b) = \frac{1}{2}(a + b) - \frac{1}{2}|a - b|$ and noting that $f(x) = -|x|$ is a directionally differentiable Lipschitz function). Moreover we can phrase the optimization problem for the upset loss as an optimization problem over the closed set $C = [0, 2\pi]^n$ representing sets$\{r_i, i = 1, \ldots, n\}$ which are then used to obtain matrices T with entries $T_{ij} = r_i - r_j \bmod 2\pi$. Fact 6 in [207] then guarantees that every local minimum is a directional stationary point of eq. (6.2). □

Discussion of the case of general k For general k-synchronization, we only require $\mathbf{M}_{i,j}^{(l)}$ to be close to zero for one l instead of all because each edge is assumed to belong to exactly one graph $\mathbf{G}_l$. Therefore in an ideal setting, for each edge $(i, j) \in \mathcal{E}$, exactly one of the entries $\mathbf{M}_{i,j}^{(l)}, l = 1, \ldots, k$ is zero. If all entries are large then this indicates that the information for (i, j) is very noisy. Subsequently, the entry for (i, j) is downweighted in the updated graph for the cycle loss function. The rationale is that when edge information is very noisy, the cycle consistency will often be violated; violations for edge information that is not so noisy are more important for angular synchronization as they should contain a stronger signal.

In terms of the cycle loss function itself, the confidence matrix $\tilde{\mathbf{C}}$ for edges in $\mathcal{G}$ arises. First consider the optimization problem in which $\tilde{\mathbf{C}}$ is omitted; taking $\tilde{\mathbf{A}} = (\mathbf{A} - \mathbf{A}^\top) \bmod 2\pi$ and we optimize the upset loss, the cycle loss, or both. For the upset loss, eq. (6.5) itself when considering fixed $\tilde{\mathbf{A}}^{(l)}$ terms, can be analyzed similarly to the analysis of $\mathcal{L}_{\text{upset}}$ in Proposition 1, using that the minimum as appearing in $\mathbf{M}_{i,j} = \min_{l \in \{1,\ldots,k\}} \mathbf{M}_{i,j}^{(l)}$ is a directionally differentiable Lipschitz function. For the cycle loss function, the expression of $\mathcal{L}_{\text{cycle}}^{(l)}$ takes a constant (when we regard $\tilde{\mathbf{A}}^{(l)}$ as fixed) away from the minimum of $S_{i,j,q}^{(l)} \bmod 2\pi$ and $(-S_{i,j,q}^{(l)}) \bmod 2\pi$. This minimum is equivalent to $|\pi - (S_{i,j,q}^{(l)} \bmod 2\pi)|$, which is again a directionally differentiable Lipschitz function. Arguing as for Proposition 1 thus shows that the statement of this proposition extends to this special treatment of the k-synchronization problem.

In our general treatment of the k-synchronization problem, the confidence matrix $\tilde{\mathbf{C}}$ depends on the maximum $\mathbf{M}$ of the residual matrices and involves the expression

$\frac{1}{1+\mathbf{M}_{i,j}}\mathbb{1}(\mathbf{A}_{i,j} \neq 0)$. While the function $f(x) = \frac{1}{1+x}$ is differentiable for $x > 0$, its composition with a function such as $\mathbf{M}$ may not be differentiable, as $\mathbf{M}$ only possesses a very weak notion of differential, called a *limiting subdifferential* in [207], to which the chain rule does not apply. This complex dependence hinders a more rigorous analysis of the general treatment of the k-synchronization problem, where even the chain rule is not guaranteed to hold.

Non-differentiable points of the loss function Although the Frobenius norm, the min function, and modulo have non-differentiable points, these points have measure zero. Moreover, as we use PyTorch autograd [1] for gradient calculation, even in the presence of non-differentiable points, backpropagation can be carried out whenever an approximate gradient can be constructed. Note that the absolute value function is convex, and hence autograd will apply the sub-gradient of the minimum norm. There also exist differentiable approximations for the modulo, and hence backpropagation can still be executed. Finally, in our experiments, we do not empirically observe any issue of convergence.

Novelty While the design of the upset loss in isolation may be relatively straightforward for the $k = 1$ case, we provide theoretical support as well as a less obvious loss function extension to handle broader $k \geq 2$ cases that rely on assigning edges to different graphs. The design of the cycle loss is not trivial and based on problem-specific insights.

E.1.2 Robustness of GNNSync

First we review DIMPA (Directed Mixed Path Aggregation) from [5] for obtaining a network embedding. DIMPA captures local network information by taking a weighted average of information from neighbors within h hops. Here we use $h = 2$ hops throughout the paper. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be an adjacency and $\mathbf{A}_s$ its row-normalization. A weighted self-loop is added to each node; then we normalize by setting $\mathbf{A}_s = (\tilde{\mathbf{D}}^s)^{-1}\tilde{\mathbf{A}}^s$, where $\tilde{\mathbf{A}}^s = \mathbf{A} + \tau\mathbf{I}_n$, with $\tilde{\mathbf{D}}^s$ the diagonal matrix with entries $\tilde{\mathbf{D}}^s_{i,i} = \sum_j \tilde{\mathbf{A}}^s_{i,j}$, $\mathbf{I}_n$ the $n \times n$ identity matrix, and τ is a small value; as in [5] we take $\tau = 0.5$.

The h-hop **source** matrix is given by $(\mathbf{A}_s)^h$. The set of *up-to-h-hop* source neighborhood matrices is denoted as $\mathbb{A}^{s,h} = \{\mathbf{I}_n, \mathbf{A}_s, \ldots, (\mathbf{A}_s)^h\}$. Similarly, for aggregating information when each node is viewed as a **target** node of a link, we carry out the same procedure for the transpose $\mathbf{A}^\top$. The set of up-to-h-hop target neighborhood matrices is denotes as $\mathbb{A}^{t,h} = \{\mathbf{I}_n, \mathbf{A}_t, \ldots, (\mathbf{A}_t)^h\}$, where $\mathbf{A}_t$ is the row-normalized target adjacency matrix calculated from $\mathbf{A}^\top$.

Let the input feature matrix be denoted by $\mathbf{X} \in \mathbb{R}^{n\times d_{\text{in}}}$. The source embedding is given by

$$\mathbf{Z}_s = \left(\sum_{\mathbf{N}\in\mathbb{A}^{s,h}} \omega^s_{\mathbf{N}} \cdot \mathbf{N}\right) \cdot \mathbf{Q}^s \in \mathbb{R}^{n\times d}, \tag{E.1}$$

where for each $\mathbf{N}$, $\omega^s_{\mathbf{N}}$ is a learnable scalar, d is the dimension of this embedding, and $\mathbf{Q}^s = \mathbf{MLP}^{(s,L)}(\mathbf{X})$. Here, the hyperparameter L controls the number of layers in the multilayer perceptron (MLP) with ReLU activation but without the bias terms; as in [5] we fix $L = 2$ throughout. Each layer of the MLP has the same number d of hidden

[1] `https://pytorch.org/docs/stable/notes/autograd.html`

units. The target embedding $\mathbf{Z}_t$ is defined similarly, with s replaced by t ineq. (E.1). After these two decoupled aggregations, the embeddings are concatenated to obtain the final node embedding as a $n \times (2d)$ matrix $\mathbf{Z} = \text{CONCAT}(\mathbf{Z}_s, \mathbf{Z}_t)$.

Proof of Proposition 3 Here we prove Proposition 3 from the main text; for convenience, we repeat it here.

Proposition 6. *For adjacency matrices* $\mathbf{A}, \hat{\mathbf{A}}$, *assume their row-normalized variants* $\mathbf{A}_s, \hat{\mathbf{A}}_s, \mathbf{A}_t, \hat{\mathbf{A}}_t$ *satisfy* $\left\|\mathbf{A}_s - \hat{\mathbf{A}}_s\right\|_F < \epsilon_s$ *and* $\left\|\mathbf{A}_t - \hat{\mathbf{A}}_t\right\|_F < \epsilon_t$, *where subscripts* s, t *denote source and target, resp. Assume further their input feature matrices* $\mathbf{X}, \hat{\mathbf{X}}$ *satisfy* $\left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F < \epsilon_f$. *Then their initial angles* $\mathbf{r}^{(0)}, \hat{\mathbf{r}}^{(0)}$ *from a trained GNNSync using DIMPA satisfy* $\left\|\mathbf{r}^{(0)} - \hat{\mathbf{r}}^{(0)}\right\|_F < B_s\epsilon_s + B_t\epsilon_s + B_f\epsilon_f$, *for values* B_s, B_t, B_f *that can be bounded by imposing constraints on model parameters and input.*

Proof. Let us assume the input feature matrices are $\mathbf{X}, \hat{\mathbf{X}} \in \mathbb{R}^{n \times d_{\text{in}}}$ for $\mathbf{A}$ and $\hat{\mathbf{A}}$, respectively.

The DIMPA procedures for the input row-normalized adjacency matrices $\mathbf{A}_s, \mathbf{A}_t$ with 2 hops and hidden dimension d can be written as a concatenation of the source and target node embeddings $\mathbf{Z}_s$ and $\mathbf{Z}_t$, where

$$\begin{aligned} \mathbf{Z}_s &= (\mathbf{I}_n + a_{s1}\mathbf{A}_s + a_{s2}\mathbf{A}_s^2)\text{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1}, \\ \mathbf{Z}_t &= (\mathbf{I}_n + a_{t1}\mathbf{A}_t + a_{t2}\mathbf{A}_t^2)\text{ReLU}(\mathbf{X}\mathbf{W}_{t0})\mathbf{W}_{t1}. \end{aligned} \tag{E.2}$$

Here $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix, $a_{s1}, a_{s2}, a_{t1}, a_{t2} \in \mathbb{R}$, $\mathbf{W}_{s0}, \mathbf{W}_{t0} \in \mathbb{R}^{d_{\text{in}} \times d}$ where d is the hidden dimension, and $\mathbf{W}_{s1}, \mathbf{W}_{t1} \in \mathbb{R}^{h \times h}$. Similarly, we have for $\hat{\mathbf{A}}_s$ and $\hat{\mathbf{A}}_t$

$$\begin{aligned} \hat{\mathbf{Z}}_s &= (\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2)\text{ReLU}(\hat{\mathbf{X}}\mathbf{W}_{s0})\mathbf{W}_{s1}, \\ \hat{\mathbf{Z}}_t &= (\mathbf{I}_n + a_{t1}\hat{\mathbf{A}}_t + a_{t2}\hat{\mathbf{A}}_t^2)\text{ReLU}(\hat{\mathbf{X}}\mathbf{W}_{t0})\mathbf{W}_{t1}. \end{aligned} \tag{E.3}$$

After DIMPA, we carry out the innerproduct procedure and sigmoid rescaling, to obtain for $k = 1$

$$\mathbf{r}^{(0)} = 2\pi\text{sigmoid}(\mathbf{Z}_s\mathbf{a}_s + \mathbf{Z}_t\mathbf{a}_t + b), \quad \hat{\mathbf{r}}^{(0)} = 2\pi\text{sigmoid}(\hat{\mathbf{Z}}_s\mathbf{a}_s + \hat{\mathbf{Z}}_t\mathbf{a}_t + b), \tag{E.4}$$

where $\mathbf{a}_s, \mathbf{a}_t \in \mathbb{R}^{d \times 1}$ and b is a trained scalar.

For $k > 1$, we have (before reshaping $\mathbf{r}^{(0)}$ and $\hat{\mathbf{r}}^{(0)}$ from shape $nk \times 1$ to shape $n \times k$ which does not change the Frobenius norm)

$$\mathbf{r}^{(0)} = 2\pi\text{sigmoid}(\mathbf{Z}_s\mathbf{a}_s + \mathbf{Z}_t\mathbf{a}_t + \mathbf{b}), \quad \hat{\mathbf{r}}^{(0)} = 2\pi\text{sigmoid}(\hat{\mathbf{Z}}_s\mathbf{a}_s + \hat{\mathbf{Z}}_t\mathbf{a}_t + \mathbf{b}), \tag{E.5}$$

where $\mathbf{a}_s, \mathbf{a}_t \in \mathbb{R}^{dk}$ and $\mathbf{b} \in \mathbb{R}^k$. Indeed, we could view the scalar b as a 1D vector, and consider eq. (E.4) as a special case of eq. (E.5).

Using eq. (E.2) and eq. (E.3), along with the triangle inequality, we have that

$$\begin{aligned}
&\left\|\mathbf{Z}_s - \hat{\mathbf{Z}}_s\right\|_F \\
=&\left\|(\mathbf{I}_n + a_{s1}\mathbf{A}_s + a_{s2}\mathbf{A}_s^2)\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1} - (\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2)\mathrm{ReLU}(\hat{\mathbf{X}}\mathbf{W}_{s0})\mathbf{W}_{s1}\right\|_F \\
\leq&\left\|(\mathbf{I}_n + a_{s1}\mathbf{A}_s + a_{s2}\mathbf{A}_s^2)\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1} - (\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2)\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1}\right\|_F + \\
&\left\|(\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2)\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1} - (\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2)\mathrm{ReLU}(\hat{\mathbf{X}}\mathbf{W}_{s0})\mathbf{W}_{s1}\right\|_F \\
\leq&\left\|[a_{s1}(\mathbf{A}_s - \hat{\mathbf{A}}_s) + a_{s2}(\mathbf{A}_s^2 - \hat{\mathbf{A}}_s^2)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1}\right\|_F + \\
&\left\|\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2\right\|_F \|\mathbf{W}_{s1}\|_F \left\|\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0}) - \mathrm{ReLU}(\hat{\mathbf{X}}\mathbf{W}_{s0})\right\|_F \\
\leq&\left\|\mathbf{A}_s - \hat{\mathbf{A}}_s\right\|_F \left\|[a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1}\right\|_F + \\
&\left\|\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2\right\|_F \|\mathbf{W}_{s1}\|_F \|\mathbf{W}_{s0}\|_F \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F \\
<&\epsilon_s B_{s0} + \epsilon_f B_{fs},
\end{aligned} \tag{E.6}$$

where $B_{s0} = \left\|[a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1}\right\|_F$ and
$B_{fs} = \left\|\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2\right\|_F \|\mathbf{W}_{s1}\|_F \|\mathbf{W}_{s0}\|_F$. Note that we also use the fact that the ReLU function is Lipschitz with Lipschitz constant 1.

Likewise, we have

$$\left\|\mathbf{Z}_t - \hat{\mathbf{Z}}_t\right\|_F < \epsilon_t B_{t0} + \epsilon_f B_{ft}, \tag{E.7}$$

where $B_{t0} = \left\|[a_{t1} + a_{t2}(\mathbf{A}_t + \hat{\mathbf{A}}_t)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{t0})\mathbf{W}_{t1}\right\|_F$ and
$B_{ft} = \left\|\mathbf{I}_n + a_{t1}\hat{\mathbf{A}}_t + a_{t2}\hat{\mathbf{A}}_t^2\right\|_F \|\mathbf{W}_{t1}\|_F \|\mathbf{W}_{t0}\|_F$.

With eq. (E.6) and eq. (E.7), noting that the sigmoid function is Lipschitz with Lipschitz constant 1, we employ eq. (E.5) to obtain

$$\begin{aligned}
&\left\|\mathbf{r}^{(0)} - \hat{\mathbf{r}}^{(0)}\right\|_F \\
=&\left\|2\pi\mathrm{sigmoid}(\mathbf{Z}_s\mathbf{a}_s + \mathbf{Z}_t\mathbf{a}_t + \mathbf{b}) - 2\pi\mathrm{sigmoid}(\hat{\mathbf{Z}}_s\mathbf{a}_s + \hat{\mathbf{Z}}_t\mathbf{a}_t + \mathbf{b})\right\|_F \\
\leq&2\pi\left\|(\mathbf{Z}_s\mathbf{a}_s + \mathbf{Z}_t\mathbf{a}_t + \mathbf{b}) - (\hat{\mathbf{Z}}_s\mathbf{a}_s + \hat{\mathbf{Z}}_t\mathbf{a}_t + \mathbf{b})\right\|_F \\
=&2\pi\left\|(\mathbf{Z}_s - \hat{\mathbf{Z}}_s)\mathbf{a}_s + (\mathbf{Z}_t - \hat{\mathbf{Z}}_t)\mathbf{a}_t\right\|_F \\
\leq&2\pi\left[\left\|(\mathbf{Z}_s - \hat{\mathbf{Z}}_s)\mathbf{a}_s\right\|_F + \left\|(\mathbf{Z}_t - \hat{\mathbf{Z}}_t)\mathbf{a}_t\right\|_F\right] \\
=&2\pi\left(\|\mathbf{a}_s\|_F\left\|\mathbf{Z}_s - \hat{\mathbf{Z}}_s\right\|_F + \|\mathbf{a}_t\|_F\left\|\mathbf{Z}_t - \hat{\mathbf{Z}}_t\right\|_F\right) \\
<&2\pi(\epsilon_s B_{s0} + \epsilon_f B_{fs} + \epsilon_t B_{t0} + \epsilon_f B_{ft}) \\
=&\epsilon_s B_s + \epsilon_t B_t + \epsilon_f B_f,
\end{aligned}$$

with values

$$\begin{aligned} B_s =& 2\pi B_{s0} = 2\pi \left\| [a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1} \right\|_F, \\ B_t =& 2\pi B_{t0} = 2\pi \left\| [a_{t1} + a_{t2}(\mathbf{A}_t + \hat{\mathbf{A}}_t)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{t0})\mathbf{W}_{t1} \right\|_F, \\ B_f =& 2\pi(B_{fs} + B_{ft}) \\ =& 2\pi \left\| \mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2 \right\|_F \|\mathbf{W}_{s1}\|_F \|\mathbf{W}_{s0}\|_F \\ & + 2\pi \left\| \mathbf{I}_n + a_{t1}\hat{\mathbf{A}}_t + a_{t2}\hat{\mathbf{A}}_t^2 \right\|_F \|\mathbf{W}_{t1}\|_F \|\mathbf{W}_{t0}\|_F . \end{aligned} \tag{E.8}$$

If we in addition have

$$\begin{aligned} &\|\mathbf{W}_{s0}\|_F \le 1, \ \ \|\mathbf{W}_{s1}\|_F \le 1, \ \ \|\mathbf{W}_{t0}\|_F \le 1, \ \ \|\mathbf{W}_{t1}\|_F \le 1, \\ &\left\|a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)\right\|_F \le 1, \ \ \left\|a_{t1} + a_{t2}(\mathbf{A}_t + \hat{\mathbf{A}}_t)\right\|_F \le 1, \\ &\|\mathbf{X}\mathbf{W}_{s0}\|_F \le 1, \ \ \left\|\mathbf{I}_n + a_{s1}\hat{\mathbf{A}}_s + a_{s2}\hat{\mathbf{A}}_s^2\right\|_F \le 1, \ \ \left\|\mathbf{I}_n + a_{t1}\hat{\mathbf{A}}_t + a_{t2}\hat{\mathbf{A}}_t^2\right\|_F \le 1, \end{aligned}$$

then the bound becomes $\left\|\mathbf{r}^{(0)} - \hat{\mathbf{r}}^{(0)}\right\|_F < 2\pi(\epsilon_s + \epsilon_t + 2\epsilon_f)$, as

$$\begin{aligned} & \left\| [a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)]\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\mathbf{W}_{s1} \right\|_F \\ \le & \left\|a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)\right\|_F \|\mathrm{ReLU}(\mathbf{X}\mathbf{W}_{s0})\|_F \|\mathbf{W}_{s1}\|_F \\ \le & \left\|a_{s1} + a_{s2}(\mathbf{A}_s + \hat{\mathbf{A}}_s)\right\|_F \|\mathbf{X}\mathbf{W}_{s0}\|_F \|\mathbf{W}_{s1}\|_F , \end{aligned}$$

and similarly for B_t.

This completes the proof. □

Discussion on GNNSync's robustness to noise With the above proposition, we could consider $\hat{\mathbf{A}}$ as the ground-truth noiseless adjacency matrix whose nonzero entries encode $(\theta_i - \theta_j) \bmod 2\pi$, and $\mathbf{A}$ as the actual noisy observed input graph. We then execute row normalization to obtain the source and target matrices $\hat{\mathbf{A}}_s$ and $\hat{\mathbf{A}}_t$ for the ground-truth and $\mathbf{A}_s$ and $\mathbf{A}_t$ for the observation, respectively. In a favorable noise regime, ϵ_s and ϵ_t would be small. The value ϵ_f comes from the feature generation method. For Spectral_RN baseline as input feature generation method for example, this involves some eigensolver corresponding to complex Hermitian matrices, and hence the Davis-Kahan Theorem [232] or one of its variants [233, 234] could be applied to upper-bound ϵ_f. As for the values B_s, B_t, and B_f, we could bound them by adding constraints to GNNSync's model parameters. Employing a backpropagation procedure with our novel loss functions could further boost the robustness of GNNSync, with learnable procedures, as shown for example in [235].

E.2 Data sets

E.2.1 Random graph outlier models

The detailed synthetic data generation process is as follows:

1. Given the number of nodes n, generate k group(s) of ground-truth angles $\{\theta_{i,l} : i \in \{1,\dots,n\}\}$ for $l \in \{1,\dots,k\}$. One option is to generate each $\theta_{i,l}$ from the same Gamma distribution with shape 0.5 and scale 2π. We denote this option with subscript "1". Since angles could be highly correlated in practical scenarios, we introduce a more realistic but challenging option "2", with multivariate normal ground-truth angles. For example, in the SNL application, angles correspond to patch rotations, and may well be that patches in similar geographic regions have corresponding rotations. The mean of the ground-truth angles is π, with covariance matrix for each $l \in \{1,\dots,k\}$ defined by $\mathbf{w}\mathbf{w}^\top$, where entries in $\mathbf{w}$ are generated independently from a standard normal distribution. We then apply mod 2π to all angles to ensure that they lie in $[0, 2\pi)$. We thus obtain the ground-truth adjacency matrix (matrices) $\mathbf{A}^l_{\mathrm{GT}} \in \mathbb{R}^{n\times n}$, whose (i,j) element is given by $(\theta_{i,l} - \theta_{j,l}) \bmod 2\pi$.

2. Generate a noisy background adjacency matrix $\mathbf{A}_{\mathrm{noise}} \in \mathbb{R}^{n\times n}$ whose entries are independently generated from a uniform distribution over $[0, 2\pi)$.

3. Generate a selection matrix $\mathbf{A}_{\mathrm{sel}} \in \mathbb{R}^{n\times n}$ whose entries are independently drawn from a Uniform(0,1) distribution. The (i,j) entry of this selection matrix is used to assign whether or not the observation is noisy, and if not noisy, to which graph it is assigned, using for $l = 1,\dots,k$

$$B_{\mathrm{noise}}(i,j;l) = \mathbb{1}\Big((1-\eta)(l-1)/k \le \mathbf{A}_{\mathrm{sel}}(i,j) < (1-\eta)l/k\Big)$$

   and $B_{\mathrm{noise}}(i,j;\infty) = \mathbb{1}\Big(\mathbf{A}_{\mathrm{sel}}(i,j) \ge (1-\eta)\Big)$, where $\mathbb{1}(\cdot)$ is the indicator function.

4. Construct a complete (without self-loops) weighted adjacency matrix $\mathbf{A}_{\mathrm{complete}} \in \mathbb{R}^{n\times n}$ by $\mathbf{A}_{\mathrm{complete}}(i,j) = \sum_{l\in\{1,\dots,k\}} \mathbf{A}^{(l)}_{\mathrm{GT}}(i,j) B_{\mathrm{noise}}(i,j;l) + \mathbf{A}_{\mathrm{noise}}(i,j) B_{\mathrm{noise}}(i,j;\infty)$.

5. Generate a measurement graph $\bar{\mathcal{G}}$ with adjacency matrix $\bar{\mathbf{G}}$ using a standard random graph model, as introduced by Sec. 6.6.1.

6. The edges in $\bar{\mathcal{G}}$ are the edges on which we observe the noisy version $\mathbf{A}_{\mathrm{complete}}$ of the ground-truth adjacency matrix (matrices) $\mathbf{A}^l_{\mathrm{GT}}$, to obtain the temporary adjacency matrix $\mathbf{T}_1$ by $\mathbf{T}_1(i,j) = \mathbf{A}_{\mathrm{complete}}(i,j)\mathbb{1}(\bar{\mathbf{G}}(i,j) \neq 0)$.

7. The true angle differences would yield a skew-symmetric matrix before taking the entries mod 2π. We therefore construct a skew-symmetric matrix $\mathbf{T}_2$ by setting $\mathbf{T}_2(i,i) = 0$ for all i, and for $i \neq j$ setting $\mathbf{T}_2(i,j) = \mathbf{T}_1(i,j)\mathbb{1}(i < j) - \mathbf{T}_1(j,i)\mathbb{1}(i > j)$.

8. In the skew-symmetric matrix, each entry appears twice, with different signs. For computational reasons, for the final adjacency matrix, we only keep the non-negative entries, except for evaluation. We obtain the final adjacency matrix $\mathbf{A}$ by $\mathbf{A}_{i,j} = \mathbf{A}(i,j) = \mathbf{T}_2(i,j)\mathbb{1}(\mathbf{T}_2(i,j) \ge 0) \bmod 2\pi$.

In addition to the two options introduced in Sec. 6.6.1, we introduce two more options for the ground-truth angle generation process here, which are both multivariate normal distributions, but with different covariance matrices. For option "3", the covariance matrix is just the identity matrix. For option "4", we have a block-diagonal covariance matrix, with six blocks, each of which is generated independently according to option "2" as stated in Sec. 6.6.1 in the main text.

As methods could be applied to different connected components of disconnected graphs separately, we focus on weakly connected networks. We have checked that all generated networks in our experiments are weakly connected.

The reason behind the naming convention "outlier" stems from the noisy offset entries in the adjacency matrix.

Steps 7 and 8 are to ensure that there does not exist an edge (i, j) such that $(\mathbf{A}_{i,j} + \mathbf{A}_{j,i}) \bmod 2\pi \neq 0$, as this would be confusing. In principle, we could work with the upper-triangular part or the lower-triangular part of the adjacency matrix first, then obtain the skew-symmetric adjacency matrix $\mathbf{T}_2$ and apply step 8 again. The procedures mentioned in Sec. 6.6.1 are what we implement in practice, which should take no more than twice the computational cost compared to working with half of the adjacency matrix at the beginning. Note that data generation only happens once before running the actual experiments.

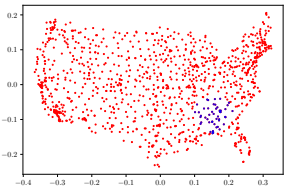
Our synthetic data settings are similar to those in previous angular synchronization papers, such as [196, 197, 203], to generate noisy samples from an outlier model (where each outlier measurement is generated uniformly at random), instead of using additive Gaussian noise in the so-called spike models. The choice of the number of nodes 360 could be changed to other numbers; we chose it to relate to 360 possible integer degrees of an angle. Note that the initial work of [197] considered n random rotation angles uniformly distributed in $[0, 2\pi)$. We do not observe a large difference in the performance of other sizes (we have also tried 300 and 500, for example).

The choice of synthetic data set construction is inspired by [197] and [196]. They are noisy versions of standard random graph models. These random graph models were chosen as they can be used for comparison. Indeed, some previous works have only used ER measurement graphs as in [203], and [197] theoretically analyzed and experimented with both sparse ER and complete measurement graphs; we already have a more thorough setup in our experiments. Furthermore, the addition of the RGG model stems from the very fact that this model is perhaps the most representative one, given the applications that have motivated the development of the group synchronization problem over the last decade. Indeed, in sensor network localization or the *molecule problem* in NMR structural biology, pairwise Euclidean distance information is only available between nearby sensors or atoms (e.g., certain sensors/atoms are connected if at most 6 miles/angstrom apart), hence leading to an RGG (disc graph) model. In this setup, in order to recover the latent coordinates, the state-of-the-art methods rely on divide-and-conquer approaches that first divide the graph into overlapping subgraphs /patches, embed locally to take advantage of the higher edge density locally, and finally aim to **stitch globally**, which is where group synchronization comes into play. Therefore, any patch-based reconstruction method that leverages the local geometry is only able to pairwise align only nearby patches that have enough points in common; far away patches that do not overlap simply cannot be aligned. Thus, the choice of RGG resembles best the real-world applications. The ER model has been predominantly used in the literature as it is easier to analyze theoretically compared to RGG, in light of available tools from the random matrix theory literature.

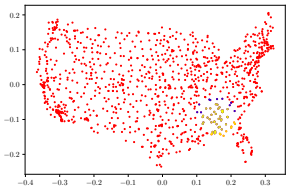
E.2.2 Sensor network localization

Previous works in the field of angular synchronization typically only consider synthetic data sets in their experiments, and those applying synchronization to real-world data do not typically publish the data sets. Concrete examples for such works include tracking the trajectory of a moving object using directional sensors [236], and habitat monitoring in an infrastructure-less environment in which radios are turned on at designated times to save power on Great Duck Island [237].

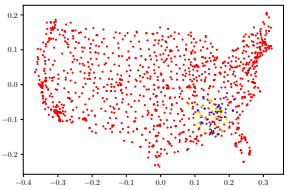
In this paper, we adapt the task on group synchronization over the Euclidean



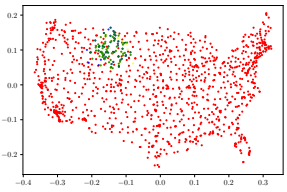
(a) Construct a patch in blue based on nearest neighbors.



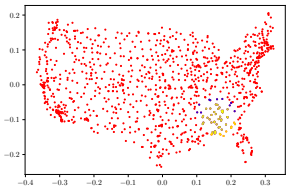
(b) Construct another patch in yellow.



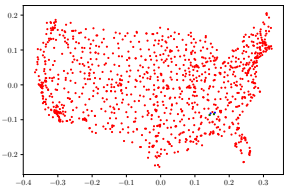
(c) Add noise to each node's coordinates in each patch.



(d) Rotate the patches based on some ground-truth rotation angles. Then apply Procrustes analysis to estimate the rotation angle based on overlapping nodes (in green). Here we use $\theta_{\text{blue}} = 174°$ and $\theta_{\text{yellow}} = 178°$ with ground-truth $\theta_{\text{blue}} - \theta_{\text{yellow}} = 356°$.



(e) Perform angular synchronization on the full adjacency matrix $\mathbf{A}$ (keeping only upper triangular entries) to obtain estimated angles for each patch. Update the estimated angles by a global shift to obtain final estimates. Apply estimated rotations to the noisy patches.



(f) Recovered central point locations (in blue) versus their ground-truth locations (in green) on the U.S. map (in red) for the two sampled patches. The locations are estimated by taking the average recovered coordinates for each city (node) from all patches that contain this node.

Figure E.1: U.S. city patch localization pipeline with two patches as an example: Starting with the ground-truth locations of U.S. cities, we construct patches using each city as a central node and add its $k_{\text{patch}} = 50$ nearest neighbors to the corresponding patch. We then add noise to each node's coordinates using independent normal distributions for x and y coordinates respectively, with mean zero and standard deviation $\eta = 0.05$ times of x and y coordinates' standard deviation, respectively. We then rotate the patches based on some ground-truth rotation angles from option "2" introduced in Sec. 4.1. Here we use $\theta_{\text{blue}} = 174°$ and $\theta_{\text{yellow}} = 178°$ with ground-truth $\theta_{\text{blue}} - \theta_{\text{yellow}} = 356°$. The estimated rotation angle from the blue patch to the yellow one is $\mathbf{A}_{\text{blue, yellow}} = 6.25$ (i.e., 358°). Then we apply Procrustes analysis to estimate the rotation angle based on overlapping nodes (but with noisy coordinates). After that, we perform angular synchronization on the full adjacency matrix $\mathbf{A}$ (keeping only upper triangular entries) to obtain estimated angles for each patch. We then update the estimated angles by shifting by the average of pairwise differences between the estimated and ground-truth angles. Here we have estimates $r_{\text{blue}} = 173°$ and $r_{\text{yellow}} = 175°$ with $r_{\text{blue}} - r_{\text{yellow}} = 358°$. Then we apply estimated rotations to the noisy patches. Finally, we obtain recovered central point locations for the two sampled patches. The locations are estimated by taking the average recovered coordinates for each city (node) from all patches that contain this node. The recovered points are colored in blue, while their ground-truth locations are colored in green.

group of rigid motions $Euc(2) = \mathbb{Z}_2 \times \mathrm{SO}(2) \times \mathbb{R}^2$ to a real-world task, by focusing

on the angular synchronization SO(2) component. This task on a real-world data set is a special case of the sensor network localization (SNL) task on the plane ($\mathbb{R}^2$) mentioned in [21], but we focus on synchronization over SO(2) only, as we do not consider any translations or reflections. Though we do not have purely real-world data sets that are employed in practice, we mimic the practical task of sensor network localization (with a focus on rotation only) and conduct the localization task on the U.S. map as well as a PACM point cloud. In detail, the task is conducted as follows, where Fig. E.1 provides an overview of the pipeline on the U.S. map with an illustrative example:

1. Starting with the ground-truth locations of U.S. cities (we have $n = 1097$ cities, see red dots in Fig. E.1), we construct patches using each city as a central node and add its $k_{\text{patch}} = 50$ nearest neighbors to the corresponding patch (see Fig. E.1(a)). We then obtain $m = n = 1097$ patches (see Fig. E.1(b) for a two-patch example). This is to represent sensor patches in the real world.

2. For each patch, we add noise to each node's coordinates using independent normal distributions for x and y coordinates respectively, with mean zero and standard deviation η times of x and y coordinates' standard deviation, respectively (see Fig. E.1(c)). Note that the noise added to the same node is independent for different patches. This is to represent noisy observations due to the lack of use of the expensive GPS service to estimate sensor coordinates.

3. We then rotate the patches based on some ground-truth rotation angles $\theta_1, \ldots, \theta_n$ (see Fig. E.1(d)). Here we generate the angles using one of the options introduced in Sec. 6.6.1 and Sec. E.2.1. This again is to represent noisy observations in the real world.

4. Then for each pair of the patches that have at least $k_{\text{thres}} = 6$ overlapping nodes, we apply Procrustes alignment [210] to estimate the rotation angle based on these overlapping nodes (but with noisy coordinates) and add an edge with the weight the estimated rotation angle to the observed (measurement) adjacency matrix $\mathbf{A}$. In other words, if two patches P_i, P_j that have at least $k_{\text{thres}} = 6$ overlapping nodes, we have $\mathbf{A}_{i,j}$ the estimated rotation angle from Procrustes alignment to rotate P_j to align with P_i. This angle is an estimation of $\theta_i - \theta_j$. The threshold is set to represent the real-world scenario where only nearby sensors may communicate with each other.

5. After that, we perform angular synchronization on the sparse adjacency matrix $\mathbf{A}$ (retaining only the upper triangular entries) to obtain the initial estimated angles $r_1^{(0)}, \ldots, r_n^{(0)}$ for each patch.

6. We then update the estimated angles by shifting by the average of pairwise differences between the estimated and ground-truth angles, in order to mod out the global degree of freedom from the synchronization step (see Fig. E.1(e)). That is, we first calculate the average of pairwise differences by $\delta_{\text{pariwise}} = \frac{1}{n}\sum_{i=1}^{n}[(r_i^{(0)} - \theta_i) \bmod 2\pi]$, then set $r_i = (r_i^{(0)} - \delta_{\text{pairwise}}) \bmod 2\pi, i = 1, \ldots, n$.

7. Next, we apply the estimated rotations to the noisy patches.

8. Finally, we estimate the city coordinates by averaging the estimated locations for each city (node) across patches that contain this city (node) (see Fig. E.1(f)).

Note that the noise in the observed adjacency matrix originates from the error by Procrustes alignment, with possible noise added to nodes' coordinates. Therefore, even when $\eta = 0$, the observed adjacency matrix may not align perfectly with ground-truth pairwise angular offsets. Besides, the observed adjacency matrix is sparse instead of complete due to the thresholding set up to only connect two nodes in the graph if the patches have enough overlapping nodes. In our experiments, we vary η from $[0, 0.05, 0.1, 0.15, 0.2, 0.25]$.

Our current experiment is focused on group synchronization over the group SO(2), and in future work we plan to explore synchronization over the full Euclidean group $\mathrm{Euc}(2) = \mathbb{Z}_2 \times \mathrm{SO}(2) \times \mathbb{R}^2$, similar to [21], where in addition to rotations, both reflections and translations are considered and synchronized over.

E.3 Implementation details

E.3.1 Setup

We use the whole graph for training for at most 1000 epochs, and stop early if the loss value does not decrease for 200 epochs. We use Stochastic Gradient Descend (SGD) as the optimizer and ℓ_2 regularization with weight decay $5 \cdot 10^{-4}$ to avoid overfitting. We use as learning rate 0.005 throughout.

For each synthetic data set, we generate 5 synthetic networks under the same setting, each with 2 repeated runs.

The DIMPA model is inherited from [5]. Indeed, other directed graph embedding neural network methods such as [114] and [6] could be employed, and we pick DIMPA just for simplicity. In our experiments, we did try out [114], and we do not observe much difference in the performance as long as some directed graph embeddings could be produced.

E.3.2 Codes, data and hardware

To fully reproduce our results, anonymized code is available at `https://github.com/SherylHYX/GNN_Sync`. Experiments were conducted on two compute nodes, each with 8 Nvidia Tesla T4, 96 Intel Xeon Platinum 8259CL CPUs @ 2.50GHz and 378GB RAM. All experiments can be completed within several days, including all variants.

The data sets considered here are relatively small and the same applies to GNNSync's competitive papers. Although each individual task does not require many resources (often $<$ 5min/run), for the synthetic data sets in this paper we have

$$\begin{aligned} & 3(\text{measurement graph styles for } k=1)\cdot 10(\text{noise levels}) \\ & \qquad \cdot 3(\text{sparsity levels})\cdot 4(\text{ground-truth options}) \\ + \quad & 3(\text{number of larger } k \text{ values})\cdot 6(\text{measurement graph} \\ & \qquad \text{options for } k>1)\cdot 8(\text{noise levels}) \\ & \qquad \cdot 3(\text{sparsity levels})\cdot 4(\text{ground-truth options}) \\ = \quad & 360 + 1,728 = 2,088 \end{aligned}$$

synthetic data sets.

Each data set requires 10 runs for each of the

$$\begin{aligned} & 1(\text{main results}) \\ & \quad +6(\text{different baselines as input features}) \\ & \quad +1(\text{no Projected Gradient Steps}) \\ & \quad +1(\text{trainable } \alpha) \\ = \; & 9 \end{aligned}$$

variants for the regular angular synchronization ($k = 1$) and

$$\begin{aligned} & 3(\text{main results}) \\ & \quad +2(\text{different baselines as input features}) \\ & \quad +2(\text{different baseline}) \\ & \quad +2(\text{trainable } \{\alpha_\gamma\}) \\ & \quad +2(\text{no Projected Gradient Steps}) \\ & \quad +2(\text{separate } \mathbf{H}^{(l)}) \\ & \quad +5(\text{other linear combinations of the loss function}) \\ = \; & 18 \end{aligned}$$

variants for general k-synchronization with $k \in \{2, 3, 4\}$. Therefore, the set of tasks in this paper requires a total of $360 \cdot 10 \cdot 9 + 1,728 \cdot 10 \cdot 18 = 32,400 + 311,040 = 343,440$ runs.

The baselines are typically faster, as they do not involve training, but GNNSync is also pretty computationally friendly, not at a significantly higher computational expense. Indeed, the set of all cycles could be pre-computed before training. For k-synchronization, we only need to verify whether all edges in a cycle are contained in an estimated graph, and to keep only these cycles for computation. There is a loop that repeats k times, but for each loop, only matrix operations are involved, which can be done in parallel for all possible cycles. This would not be too expensive, as validated by our experiments. Besides, for the MSE function, we do not use it for training, so it is not a loss function in the first place. For evaluation, it does require computing all permutations of k but the evaluation is only conducted once. Therefore, these computationally expensive operations (locating all possible cycles and permutations of k in the MSE computation) are not involved in training, but only before or after training, and hence our method is still scalable with n and k.

E.3.3 Baseline implementation

For CEMP_GCW and CEMP_MST, we adapt the MatLab code from `https://github.com/yunpeng-shi/CEMP/tree/main/SO2` to Python. For other baselines, we implement the approaches based on equations from the original papers. For TAS, we transform the MatLab codes from the authors of [204] to Python. We set the number of epochs to 50, and set the trimming parameter to zero due to the high sparsity level of our synthetic networks, as otherwise, almost all predictions would be the same, just like the Trivial solution.

Besides, we do not compare GNNSync against the method in [200] in our experiments, as there is no code available. Also, their algorithm involves integration and angular argmax in each iteration, which seems to be computationally expensive.

Finally, we are aware of Semi-Definite Programming (SDP) baselines but have found them too time-consuming or space-inefficient. Also, from [197], we know that SDP and spectral methods have comparable performance. Therefore, in our experiments, we omit the SDP results.

E.3.4 MSE calculation

As stated in eq. (6.2), the MSE function calculates the mean square error using a global angular rotation that minimizes the MSE value. The implementation of the MSE function, however, does not explicitly search for the lowest MSE value through grid search or gradient descent. Inspired by the implementation of the MSE in [208], we first map each of the predicted angles $\mathbf{r}$ and the ground-truth angles $\mathbf{R}$ to rotation matrices by the mapping function

$$rot(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

We then calculate the matrix

$$\mathbf{Q} = \frac{1}{n}\sum_{i=1}^{n} rot(\mathbf{R}_i)^\top \cdot rot(\mathbf{r}_i).$$

The MSE value is given by

$$4 - 2\sum_{i=1}^{n} sing_i(\mathbf{Q}),$$

where $sing_i(\mathbf{Q})$ is the i-th singular value of $\mathbf{Q}$ during Singular Value Decomposition (SVD).

E.4 Extended experimental results

This section reports extended experimental results mentioned in the main text.

E.4.1 Extended main results

Full main synthetic experimental results are shown in Fig. E.2 to E.13. Results on real-world data sets are shown in Fig. E.14 to E.23, while other PACM results are omitted but with the same conclusion. To accommodate potential variability in different runs, we report the mean and standard deviation of ten runs (two repeated runs on five different sets of ground-truth angles) in Tab. E.1 and E.3. We also compute the Average Normalized Error (ANE) for coordinate recovery similar to eq. (44) of [21], and report mean and one standard deviation of the results in Tab. E.2 and E.4. Specifically, denote (x_i, y_i) as the ground-truth coordinate for node i where $i = 1, \ldots, n$, and $(\hat{x}_i, \hat{y}_i)$ as the predicted coordinate, we define the Average Normalized Error (ANE) as

$$ANE = \frac{\sqrt{\sum_{i=1}^{n}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]}}{\sqrt{\sum_{i=1}^{n}[(x_i - x_0)^2 + (y_i - y_0)^2]}}, \tag{E.9}$$

where $(x_0, y_0) = (\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i) = (0, 0)$ is the center of mass of the true coordinates. We conclude that GNNSync is able to effectively recover coordinates. We also observe that GNNSync is more robust to the noise of patch coordinates. We omit the visual plots for the “Trivial" baseline but report its performance in Tab. E.1, E.2, E.3, and E.4.

Table E.1: Average MSE values (plus/minus one standard deviation) for the real-world experiments on the U.S. map over ten runs. The best is marked in **bold red** while the second best is in underline blue .

η	option	GNNSync	Spectral	Spectral_RN	GPM	TranSync	CEMP_GCW	CEMP_MST	TAS	Trivial
0	1	0.010±0.006	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	2.358±0.075	2.442±0.069
0	2	0.004±0.001	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	1.567±0.035	1.566±0.037
0	3	**0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	0.138±0.174	0.138±0.174
0	4	0.002±0.001	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.651±0.237	0.663±0.238
0.05	1	0.014±0.006	0.007±0.001	**0.006±0.001**	**0.006±0.001**	0.036±0.027	0.082±0.040	1.162±0.438	2.340±0.098	2.411±0.094
0.05	2	0.010±0.002	**0.006±0.000**	**0.006±0.000**	**0.006±0.000**	0.026±0.006	0.072±0.006	1.353±0.461	1.559±0.032	1.559±0.033
0.05	3	**0.006±0.002**	0.007±0.001	**0.006±0.001**	**0.006±0.001**	0.027±0.009	0.816±0.494	2.827±0.980	0.293±0.481	0.294±0.482
0.05	4	0.007±0.002	**0.006±0.001**	**0.006±0.000**	**0.006±0.001**	0.024±0.007	0.319±0.251	1.826±0.918	0.750±0.373	0.759±0.372
0.1	1	0.030±0.005	0.030±0.007	0.027±0.006	**0.025±0.005**	0.147±0.026	0.528±0.077	2.819±0.292	2.318±0.104	2.368±0.101
0.1	2	**0.025±0.002**	0.031±0.002	0.028±0.002	0.027±0.001	0.188±0.053	0.397±0.048	2.874±0.315	1.558±0.030	1.564±0.028
0.1	3	**0.019±0.003**	0.027±0.005	0.024±0.004	0.025±0.004	0.129±0.048	1.775±1.104	3.583±0.298	0.138±0.175	0.138±0.174
0.1	4	**0.021±0.006**	0.026±0.003	0.023±0.002	0.023±0.002	0.127±0.033	1.055±0.616	3.102±0.205	0.656±0.238	0.663±0.238
0.15	1	**0.059±0.008**	0.075±0.012	0.072±0.013	0.062±0.008	0.483±0.279	1.496±0.473	3.698±0.092	2.363±0.117	2.396±0.118
0.15	2	**0.057±0.005**	0.082±0.005	0.076±0.007	0.067±0.003	0.492±0.122	1.193±0.238	3.501±0.301	1.566±0.037	1.566±0.037
0.15	3	**0.037±0.006**	0.052±0.011	0.048±0.009	0.046±0.010	0.277±0.064	1.745±0.540	3.671±0.229	0.138±0.175	0.138±0.174
0.15	4	**0.043±0.007**	0.055±0.008	0.052±0.006	0.046±0.005	0.591±0.384	1.317±0.174	3.519±0.152	0.658±0.239	0.663±0.238
0.2	1	**0.101±0.007**	0.148±0.024	0.151±0.019	0.107±0.009	1.000±0.257	2.568±0.906	3.711±0.122	2.377±0.095	2.399±0.093
0.2	2	**0.101±0.006**	0.163±0.017	0.159±0.018	0.122±0.009	1.054±0.245	2.082±0.263	3.717±0.109	1.564±0.037	1.566±0.037
0.2	3	**0.065±0.015**	0.095±0.018	0.092±0.019	0.078±0.015	0.756±0.222	2.239±0.556	3.699±0.111	0.335±0.377	0.336±0.380
0.2	4	**0.066±0.008**	0.096±0.017	0.095±0.019	0.072±0.010	0.717±0.258	2.040±0.334	3.751±0.089	0.660±0.239	0.663±0.238
0.25	1	**0.158±0.008**	0.244±0.038	0.263±0.039	0.164±0.011	1.690±0.569	2.888±0.240	3.791±0.096	2.427±0.069	2.442±0.069
0.25	2	**0.163±0.013**	0.291±0.038	0.294±0.042	0.193±0.018	1.888±0.737	2.633±0.294	3.782±0.108	1.564±0.037	1.566±0.037
0.25	3	0.105±0.065	0.143±0.018	0.242±0.105	**0.103±0.021**	1.265±0.583	3.050±0.552	3.765±0.081	0.138±0.174	0.138±0.174
0.25	4	0.128±0.074	0.170±0.029	0.176±0.040	**0.107±0.017**	1.296±0.320	2.787±0.477	3.787±0.070	0.660±0.238	0.663±0.238

Table E.2: Average ANE values (plus/minus one standard deviation) for the real-world experiments on the U.S. map over ten runs. The best is marked in **bold red** while the second best is in underline blue .

η	option	GNNSync	Spectral	Spectral_RN	GPM	TranSync	CEMP_GCW	CEMP_MST	TAS	Trivial
0	1	0.075±0.028	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.740±0.021	0.973±0.263
0	2	0.047±0.010	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.425±0.015	0.423±0.014
0	3	0.011±0.009	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.050±0.049	0.049±0.049
0	4	0.030±0.016	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.213±0.078	0.219±0.078
0.05	1	0.074±0.027	**0.032±0.010**	**0.032±0.011**	**0.032±0.009**	0.360±0.591	0.121±0.069	0.709±0.495	0.735±0.025	0.984±0.277
0.05	2	0.054±0.013	0.284±0.508	**0.030±0.002**	0.159±0.258	0.092±0.020	0.146±0.035	0.638±0.238	0.421±0.015	0.419±0.013
0.05	3	**0.030±0.017**	0.118±0.160	0.035±0.005	0.039±0.008	0.469±0.763	0.708±0.338	0.900±0.178	0.089±0.124	0.089±0.124
0.05	4	0.229±0.569	0.404±0.744	**0.031±0.007**	0.032±0.008	0.058±0.009	0.494±0.595	0.750±0.369	0.243±0.121	0.250±0.122
0.1	1	0.078±0.014	**0.070±0.024**	0.204±0.152	0.160±0.181	0.164±0.050	0.568±0.409	1.068±0.135	0.733±0.028	0.987±0.280
0.1	2	**0.057±0.017**	0.077±0.007	0.076±0.005	0.259±0.355	0.270±0.042	0.345±0.042	0.992±0.161	0.415±0.016	0.414±0.016
0.1	3	**0.046±0.015**	0.085±0.017	0.079±0.018	0.324±0.466	0.503±0.503	0.920±0.298	1.019±0.109	0.053±0.047	0.053±0.047
0.1	4	0.247±0.583	0.075±0.019	**0.069±0.016**	0.124±0.119	0.226±0.135	0.828±0.382	0.964±0.146	0.215±0.079	0.219±0.078
0.15	1	0.313±0.449	0.216±0.153	0.602±0.713	**0.104±0.022**	0.313±0.027	0.859±0.422	1.019±0.071	0.755±0.032	1.101±0.257
0.15	2	**0.076±0.019**	0.573±0.679	0.151±0.062	0.136±0.026	0.460±0.304	0.983±0.460	0.964±0.065	0.425±0.015	0.424±0.014
0.15	3	0.181±0.347	0.189±0.165	0.102±0.024	0.107±0.025	0.524±0.527	0.787±0.145	0.979±0.105	**0.057±0.045**	**0.057±0.045**
0.15	4	**0.075±0.032**	0.104±0.030	0.096±0.020	0.088±0.024	0.929±0.522	0.947±0.460	1.019±0.060	0.215±0.077	0.220±0.078
0.2	1	0.131±0.026	0.247±0.203	0.612±0.375	**0.125±0.034**	0.835±0.435	0.876±0.140	1.087±0.058	0.749±0.027	0.976±0.267
0.2	2	**0.094±0.018**	0.451±0.332	0.225±0.109	0.433±0.506	0.609±0.234	0.967±0.272	1.005±0.038	0.424±0.017	0.424±0.014
0.2	3	**0.087±0.037**	0.133±0.025	0.130±0.028	0.132±0.030	0.964±0.649	0.889±0.211	1.020±0.053	0.109±0.092	0.110±0.093
0.2	4	**0.077±0.023**	0.123±0.029	0.126±0.025	0.103±0.027	0.724±0.484	0.911±0.217	0.999±0.031	0.215±0.076	0.221±0.077
0.25	1	**0.197±0.066**	0.223±0.101	0.337±0.255	0.478±0.418	0.656±0.188	0.978±0.133	1.008±0.032	0.760±0.022	0.974±0.263
0.25	2	**0.122±0.034**	0.494±0.399	0.393±0.219	0.546±0.701	0.912±0.215	0.895±0.061	1.030±0.025	0.425±0.017	0.425±0.014
0.25	3	0.260±0.397	0.281±0.215	0.230±0.065	0.157±0.038	1.193±0.429	1.085±0.153	1.009±0.052	**0.067±0.041**	**0.067±0.041**
0.25	4	**0.134±0.109**	0.513±0.662	0.188±0.047	0.288±0.343	0.612±0.174	0.958±0.147	1.042±0.048	0.217±0.074	0.222±0.076

Table E.3: Average MSE values (plus/minus one standard deviation) for the real-world experiments on the PACM point cloud over ten runs. The best is marked in **bold red** while the second best is in underline blue .

η	option	GNNSync	Spectral	Spectral_RN	GPM	TranSync	CEMP_GCW	CEMP_MST	TAS	Trivial
0	1	0.010±0.013	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	**-0.000±0.000**	2.249±0.128	2.468±0.139
0	2	0.001±0.001	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	1.640±0.041	1.565±0.042
0	3	0.002±0.002	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	1.221±0.779	1.160±0.783
0	4	0.001±0.001	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	1.196±0.430	1.176±0.426
0.05	1	0.015±0.011	**0.003±0.000**	**0.003±0.001**	**0.003±0.000**	0.010±0.004	0.039±0.008	0.202±0.093	2.246±0.129	2.468±0.139
0.05	2	0.004±0.001	0.003±0.000	**0.002±0.000**	**0.002±0.000**	0.010±0.003	0.024±0.016	0.885±1.157	1.611±0.057	1.543±0.051
0.05	3	0.004±0.002	**0.003±0.000**	**0.003±0.000**	**0.003±0.000**	0.009±0.004	0.057±0.058	0.528±0.496	1.218±0.777	1.160±0.783
0.05	4	**0.003±0.001**	**0.003±0.000**	**0.003±0.000**	**0.003±0.000**	0.011±0.012	0.177±0.206	0.718±0.292	1.006±0.246	0.987±0.253
0.1	1	0.015±0.005	**0.012±0.004**	**0.012±0.004**	**0.012±0.004**	0.049±0.014	0.187±0.107	1.580±0.642	2.122±0.150	2.346±0.135
0.1	2	**0.010±0.001**	0.011±0.001	**0.010±0.001**	**0.010±0.001**	0.044±0.011	0.145±0.031	1.599±0.459	1.631±0.036	1.565±0.042
0.1	3	0.012±0.003	**0.010±0.001**	**0.010±0.001**	**0.010±0.001**	0.055±0.025	0.298±0.377	1.394±0.659	1.213±0.775	1.160±0.783
0.1	4	**0.010±0.001**	0.011±0.001	**0.010±0.001**	**0.010±0.001**	0.067±0.052	0.271±0.148	1.934±0.656	1.191±0.431	1.176±0.426
0.15	1	0.032±0.009	0.029±0.012	**0.028±0.011**	**0.028±0.012**	0.136±0.088	0.380±0.192	2.009±0.657	2.262±0.136	2.468±0.139
0.15	2	0.022±0.002	0.022±0.002	**0.020±0.002**	0.021±0.002	0.109±0.034	0.522±0.528	2.927±0.254	1.587±0.064	1.530±0.057
0.15	3	0.021±0.005	0.020±0.004	**0.019±0.004**	**0.019±0.004**	0.107±0.035	0.523±0.416	2.970±0.502	0.570±0.468	0.524±0.446
0.15	4	**0.020±0.002**	0.022±0.001	**0.020±0.001**	**0.020±0.001**	0.091±0.038	0.567±0.324	2.869±0.430	1.185±0.427	1.176±0.426
0.2	1	**0.050±0.012**	0.053±0.019	0.051±0.018	0.051±0.019	0.236±0.117	0.546±0.167	2.894±0.312	2.288±0.139	2.468±0.139
0.2	2	**0.037±0.003**	0.039±0.006	**0.037±0.005**	0.038±0.006	0.250±0.109	0.826±0.410	2.982±0.595	1.610±0.041	1.565±0.042
0.2	3	0.032±0.006	0.032±0.004	**0.030±0.003**	**0.030±0.004**	0.157±0.045	0.872±0.368	3.330±0.284	1.180±1.161	1.166±1.170
0.2	4	**0.030±0.002**	0.034±0.002	0.032±0.002	0.031±0.002	0.191±0.084	0.894±0.220	3.312±0.295	0.998±0.246	0.987±0.253
0.25	1	**0.069±0.019**	0.082±0.026	0.078±0.025	0.081±0.028	0.389±0.185	1.028±0.224	3.513±0.350	2.318±0.140	2.468±0.139
0.25	2	**0.054±0.005**	0.059±0.010	0.056±0.009	0.057±0.011	0.454±0.386	0.955±0.126	3.586±0.224	1.605±0.040	1.565±0.042
0.25	3	0.050±0.013	0.051±0.009	**0.047±0.008**	0.049±0.009	0.341±0.105	0.942±0.049	3.469±0.164	1.475±1.071	1.460±1.084
0.25	4	**0.047±0.005**	0.053±0.003	0.050±0.004	0.050±0.003	0.359±0.176	1.179±0.442	3.238±0.333	1.181±0.425	1.176±0.426

Table E.4: Average ANE values (plus/minus one standard deviation) for the real-world experiments on the PACM point cloud over ten runs. The best is marked in **bold red** while the second best is in underline blue .

η	option	GNNSync	Spectral	Spectral_RN	GPM	TranSync	CEMP_GCW	CEMP_MST	TAS	Trivial
0	1	0.118±0.093	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	1.357±0.080	1.655±0.426
0	2	0.051±0.031	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.870±0.029	0.785±0.020
0	3	0.048±0.034	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.645±0.396	0.597±0.390
0	4	0.038±0.024	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	**0.000±0.000**	0.687±0.236	0.631±0.223
0.05	1	0.139±0.079	0.722±1.382	0.088±0.120	**0.029±0.009**	0.093±0.042	0.154±0.066	0.964±1.353	1.347±0.082	1.655±0.426
0.05	2	0.055±0.019	0.671±1.288	**0.022±0.003**	0.024±0.005	0.102±0.048	0.121±0.093	0.692±0.687	0.869±0.043	0.805±0.045
0.05	3	0.061±0.031	0.029±0.007	**0.026±0.007**	0.028±0.007	0.062±0.017	0.823±1.317	0.588±0.310	0.640±0.393	0.597±0.389
0.05	4	0.046±0.021	0.032±0.007	**0.030±0.006**	0.031±0.007	0.068±0.039	0.291±0.253	0.826±0.425	0.699±0.256	0.671±0.279
0.1	1	0.116±0.037	**0.072±0.039**	0.074±0.039	0.661±1.159	0.220±0.066	0.896±1.294	1.326±0.710	1.320±0.094	1.654±0.433
0.1	2	0.068±0.021	0.197±0.275	**0.053±0.009**	0.088±0.063	0.210±0.041	1.057±1.368	1.499±0.537	0.851±0.018	0.785±0.020
0.1	3	0.077±0.039	0.144±0.168	0.158±0.216	**0.055±0.016**	0.196±0.070	0.401±0.320	1.470±0.907	0.636±0.391	0.597±0.389
0.1	4	0.057±0.014	0.063±0.011	**0.056±0.014**	0.060±0.013	0.270±0.185	0.639±0.383	1.419±0.437	0.683±0.245	0.632±0.223
0.15	1	0.163±0.063	0.228±0.178	0.159±0.064	**0.138±0.060**	0.301±0.235	0.511±0.346	1.313±0.653	1.357±0.084	1.655±0.426
0.15	2	0.085±0.020	0.078±0.017	**0.076±0.015**	0.349±0.382	0.232±0.078	0.739±0.438	2.096±0.323	0.844±0.043	0.768±0.032
0.15	3	0.095±0.039	0.738±1.340	**0.076±0.031**	0.079±0.032	0.239±0.061	0.903±1.036	1.803±0.217	0.322±0.241	0.298±0.226
0.15	4	**0.079±0.020**	0.094±0.016	0.086±0.019	0.090±0.017	0.366±0.240	1.125±1.250	1.775±0.338	0.670±0.237	0.632±0.223
0.2	1	**0.189±0.063**	0.196±0.068	0.199±0.069	0.193±0.071	0.937±1.219	1.528±0.954	2.331±0.262	1.378±0.085	1.655±0.426
0.2	2	**0.102±0.029**	0.111±0.028	0.110±0.031	0.265±0.313	0.402±0.239	1.202±0.789	1.909±0.278	0.840±0.027	0.785±0.020
0.2	3	**0.097±0.036**	0.846±1.507	0.526±0.873	0.169±0.173	0.336±0.124	0.829±0.338	1.852±0.125	0.633±0.591	0.635±0.610
0.2	4	**0.096±0.017**	0.116±0.020	0.807±0.855	0.104±0.022	0.398±0.247	1.059±0.303	1.952±0.435	0.690±0.270	0.672±0.279
0.25	1	0.290±0.277	0.254±0.074	0.252±0.075	**0.247±0.081**	0.465±0.140	1.505±0.824	2.016±0.194	1.391±0.093	1.655±0.426
0.25	2	**0.109±0.029**	0.142±0.051	0.129±0.040	0.372±0.494	0.907±1.221	1.312±0.891	2.087±0.138	0.838±0.028	0.785±0.020
0.25	3	**0.311±0.615**	0.557±0.872	0.407±0.586	0.313±0.268	0.871±0.806	1.713±0.728	1.941±0.078	0.779±0.534	0.782±0.560
0.25	4	**0.109±0.022**	0.140±0.035	0.888±0.965	0.131±0.035	0.427±0.117	1.248±0.622	1.923±0.205	0.648±0.235	0.633±0.222

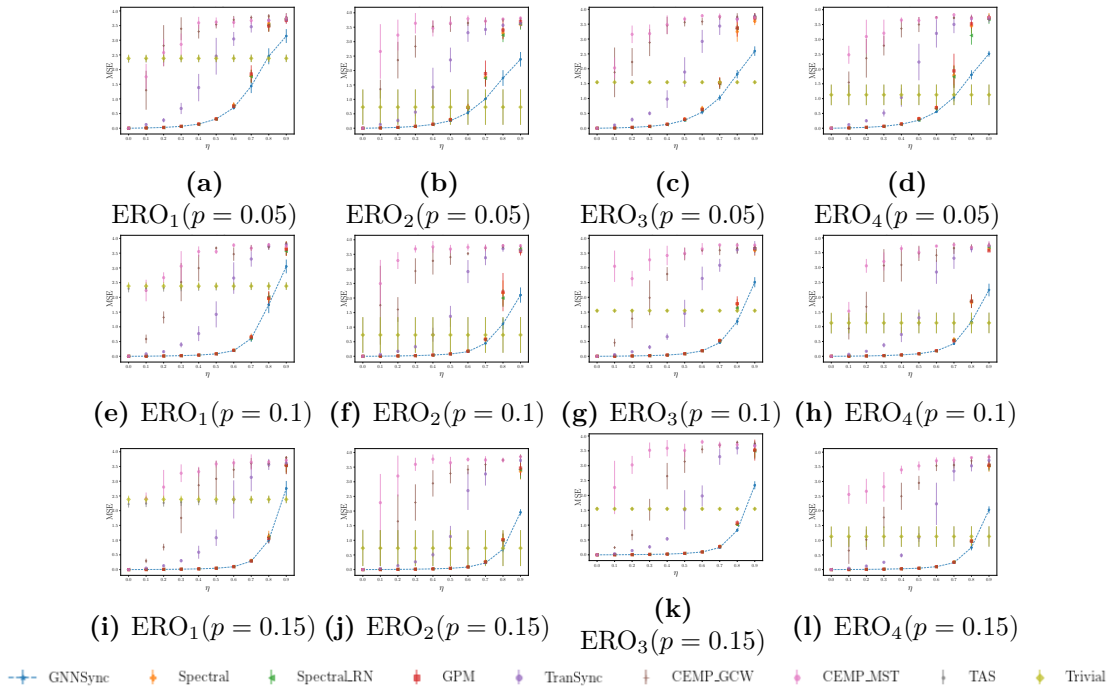


Figure E.2: MSE performance comparison on GNNSync against baselines on angular synchronization ($k = 1$) for ERO models. Error bars indicate one standard deviation.

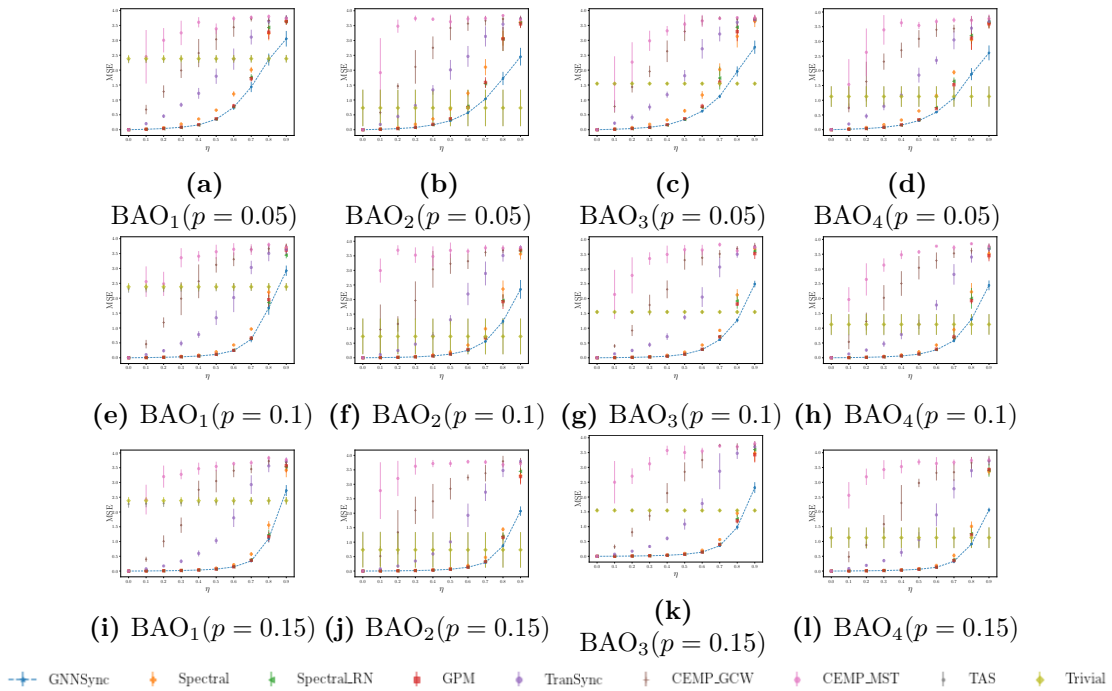


Figure E.3: MSE performance comparison on GNNSync against baselines on angular synchronization ($k = 1$) for BAO models. Error bars indicate one standard deviation.

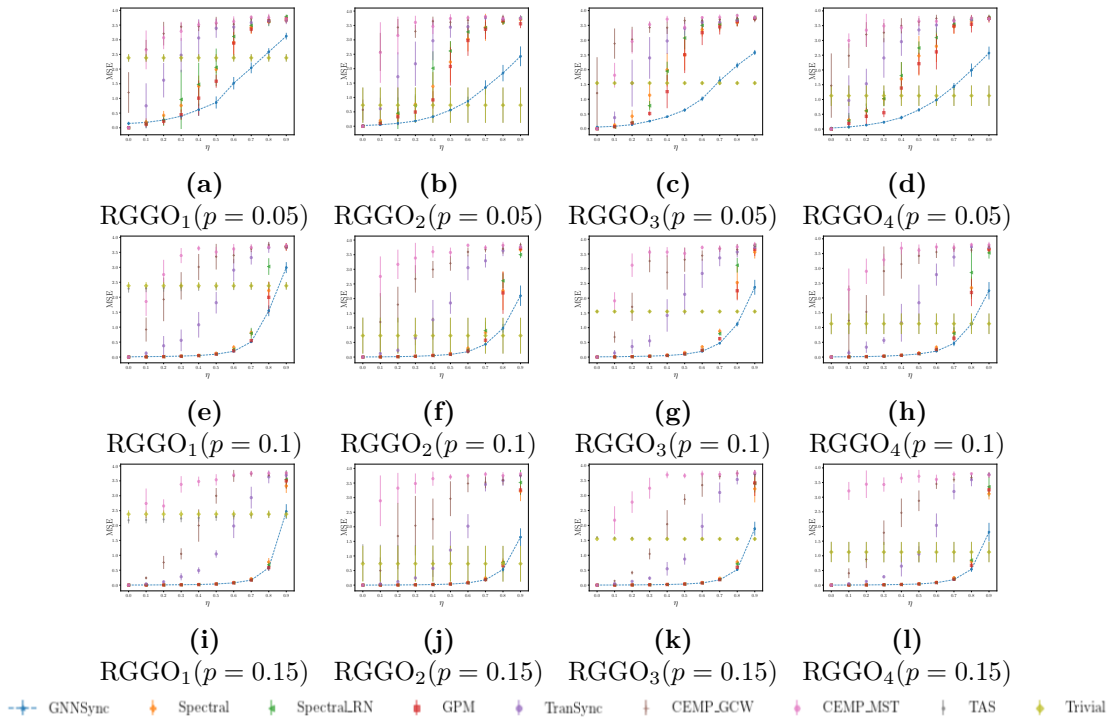


Figure E.4: MSE performance comparison on GNNSync against baselines on angular synchronization ($k = 1$) for RGGO models. Error bars indicate one standard deviation.

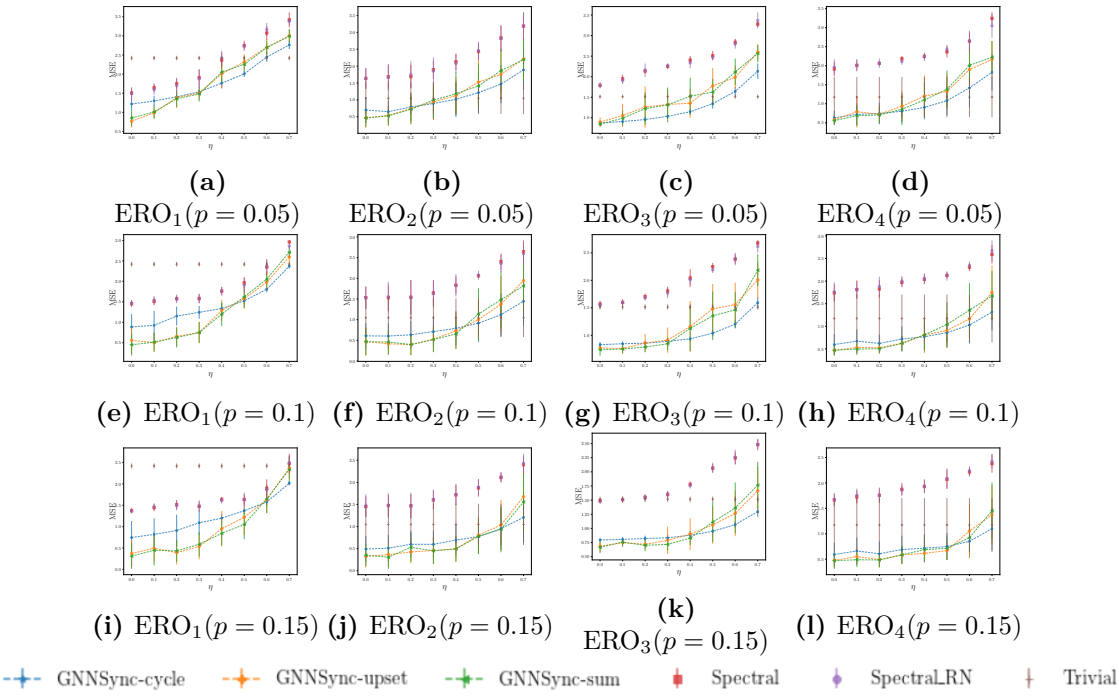


Figure E.5: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 2$ for ERO models. Error bars indicate one standard deviation.

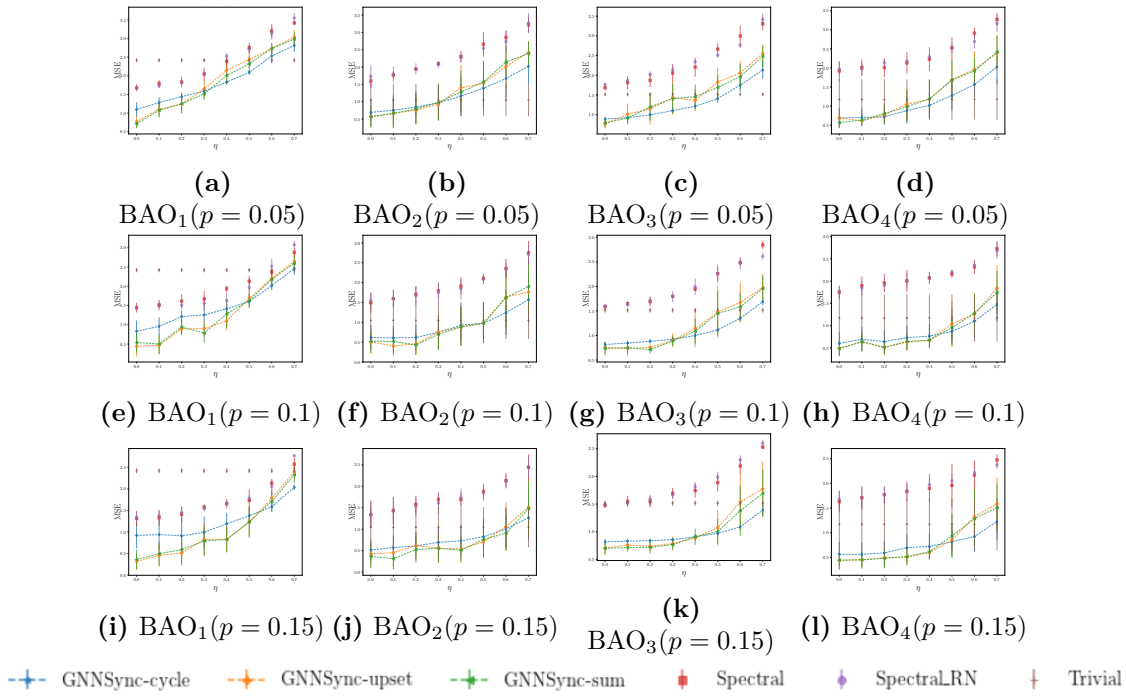


Figure E.6: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 2$ for BAO models. Error bars indicate one standard deviation.

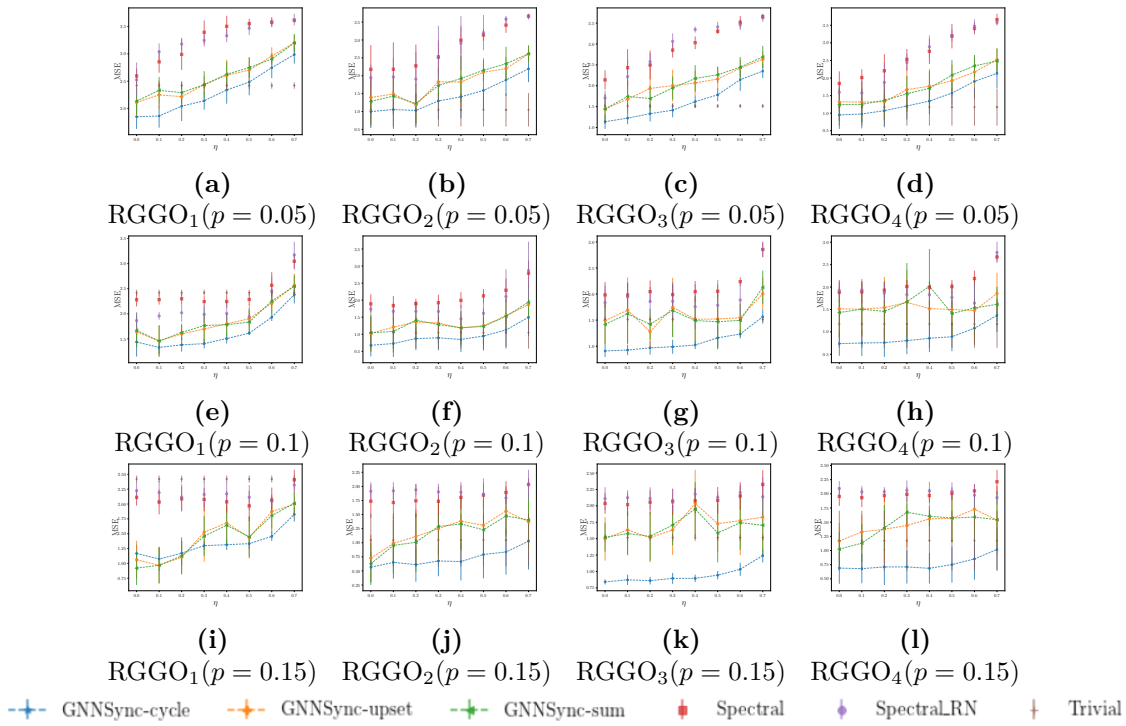


Figure E.7: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 2$ for RGGO models. Error bars indicate one standard deviation.

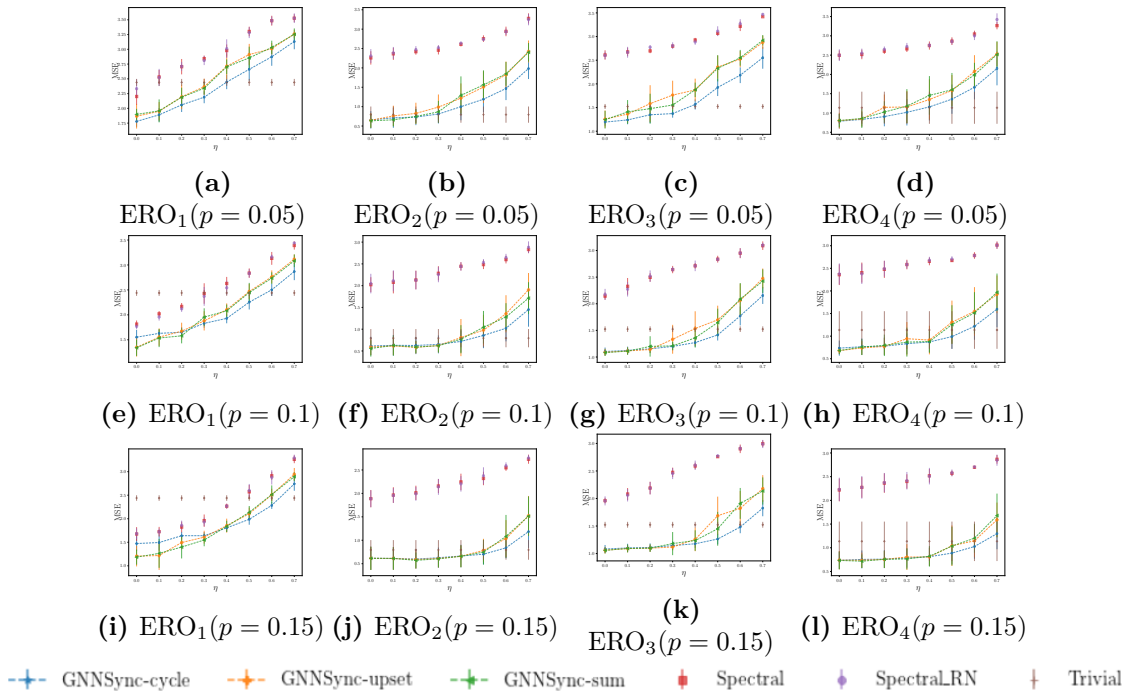


Figure E.8: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 3$ for ERO models. Error bars indicate one standard deviation.

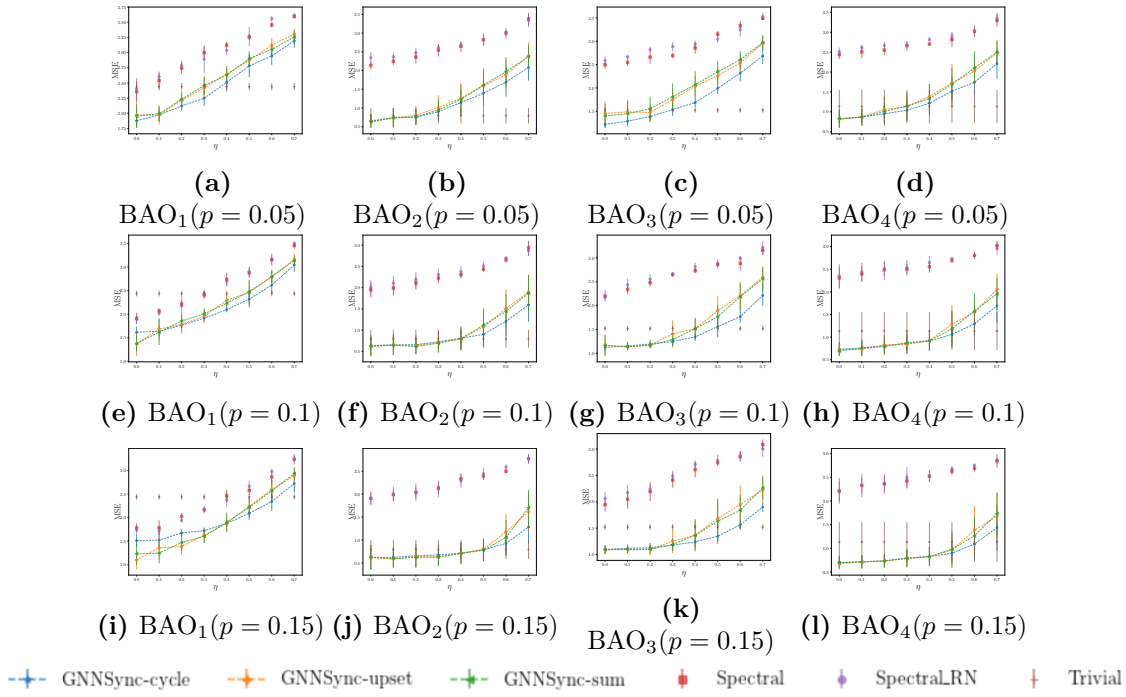


Figure E.9: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 3$ for BAO models. Error bars indicate one standard deviation.

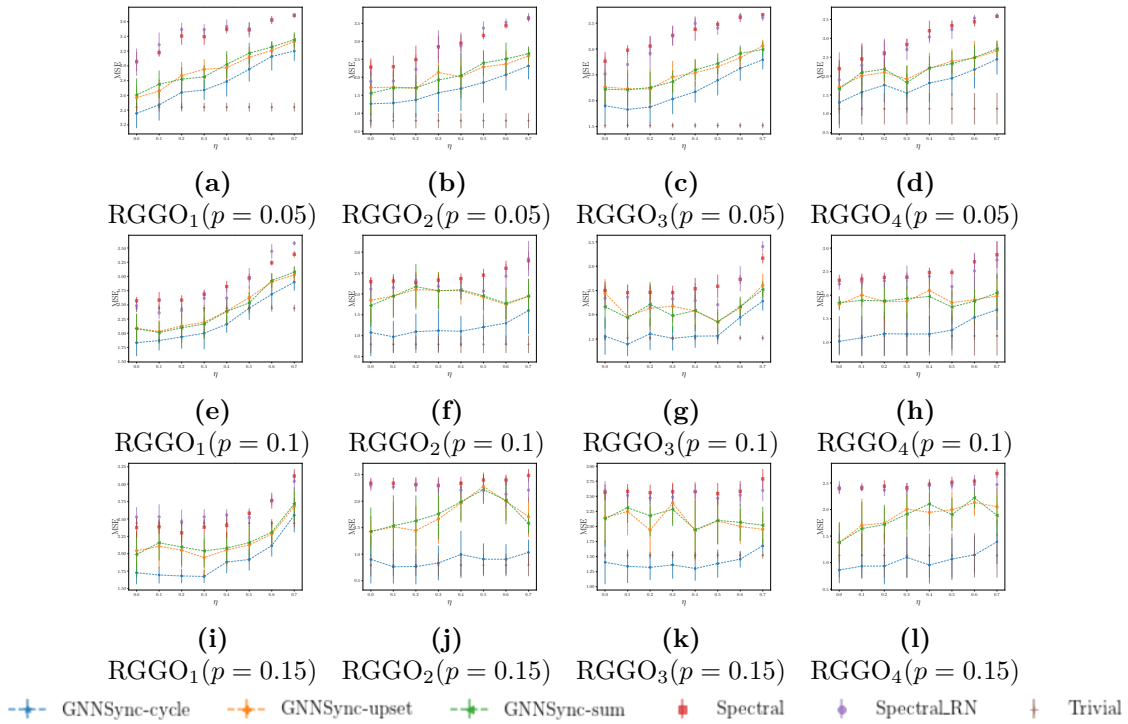


Figure E.10: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 3$ for RGGO models. Error bars indicate one standard deviation.

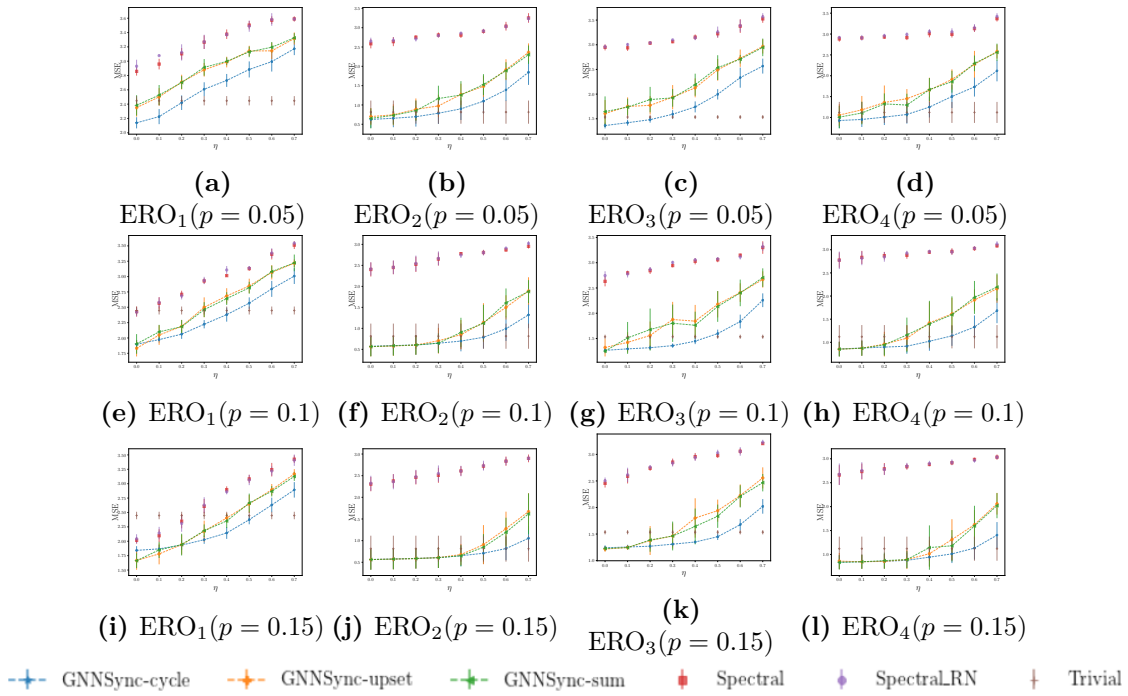


Figure E.11: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 4$ for ERO models. Error bars indicate one standard deviation.

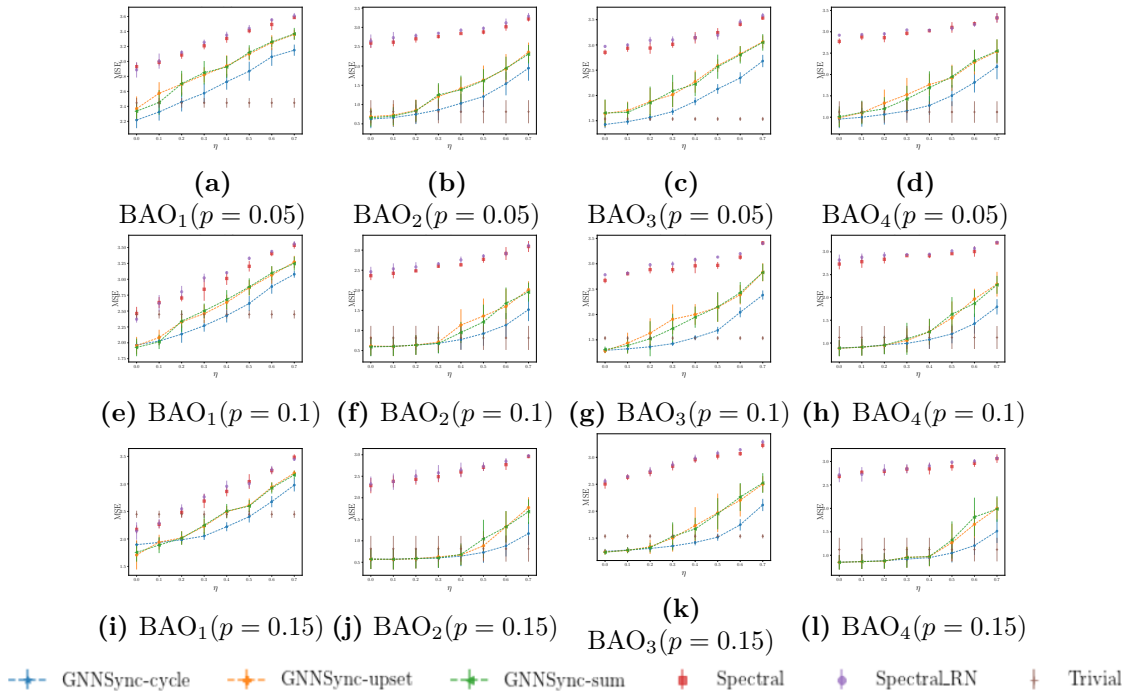


Figure E.12: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 4$ for BAO models. Error bars indicate one standard deviation.

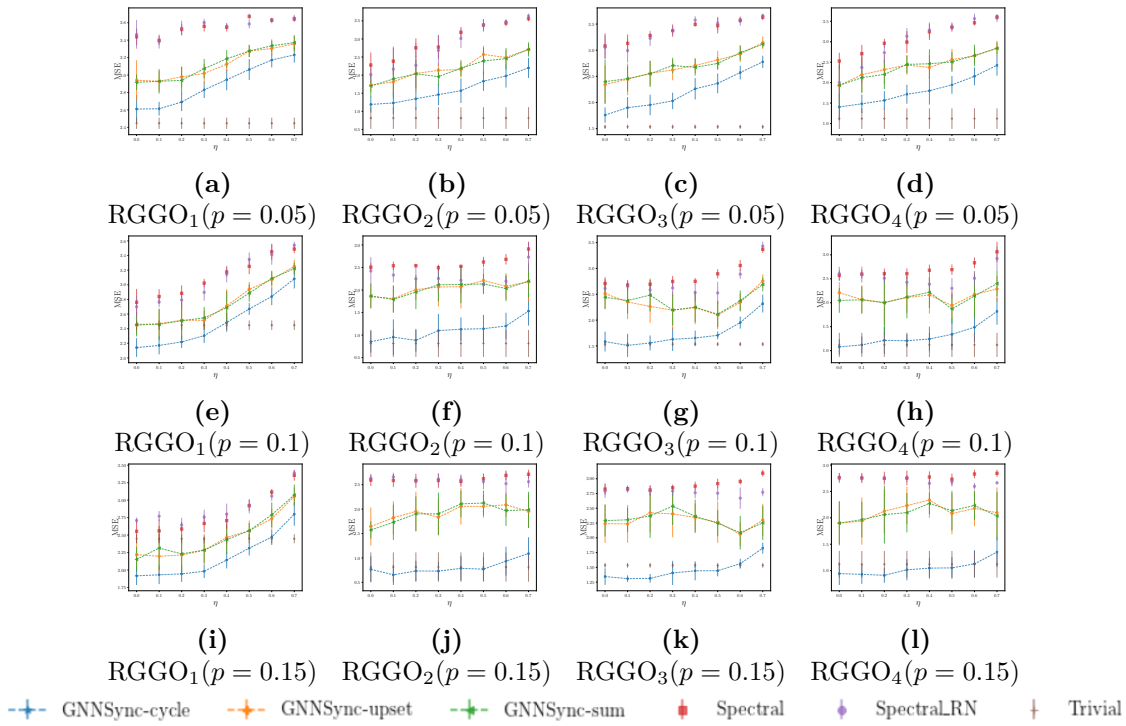
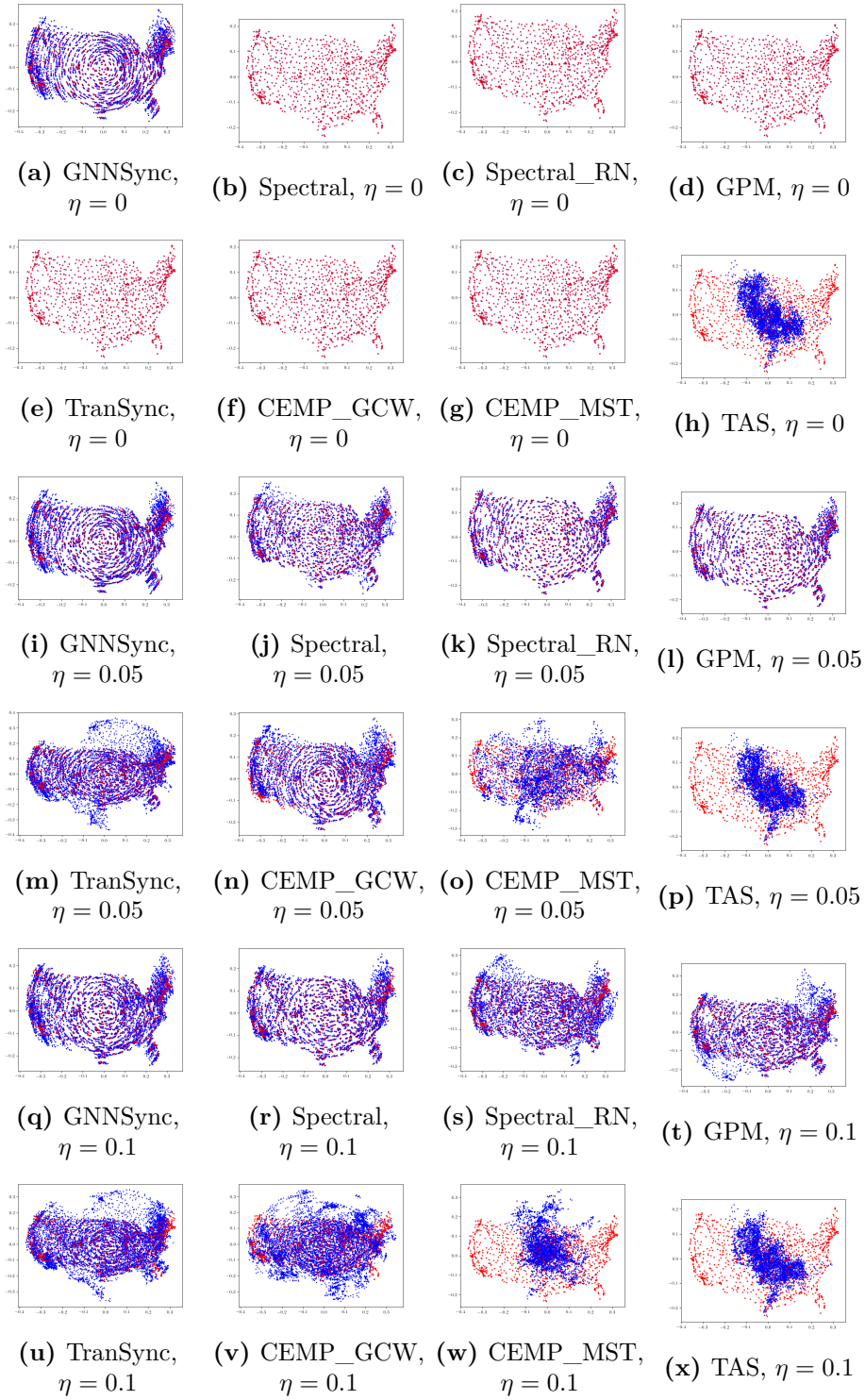


Figure E.13: MSE performance comparison on GNNSync against baselines on $k-$synchronization with $k = 4$ for RGGO models. Error bars indicate one standard deviation.



(a) GNNSync, $\eta = 0$ **(b)** Spectral, $\eta = 0$ **(c)** Spectral_RN, $\eta = 0$ **(d)** GPM, $\eta = 0$

(e) TranSync, $\eta = 0$ **(f)** CEMP_GCW, $\eta = 0$ **(g)** CEMP_MST, $\eta = 0$ **(h)** TAS, $\eta = 0$

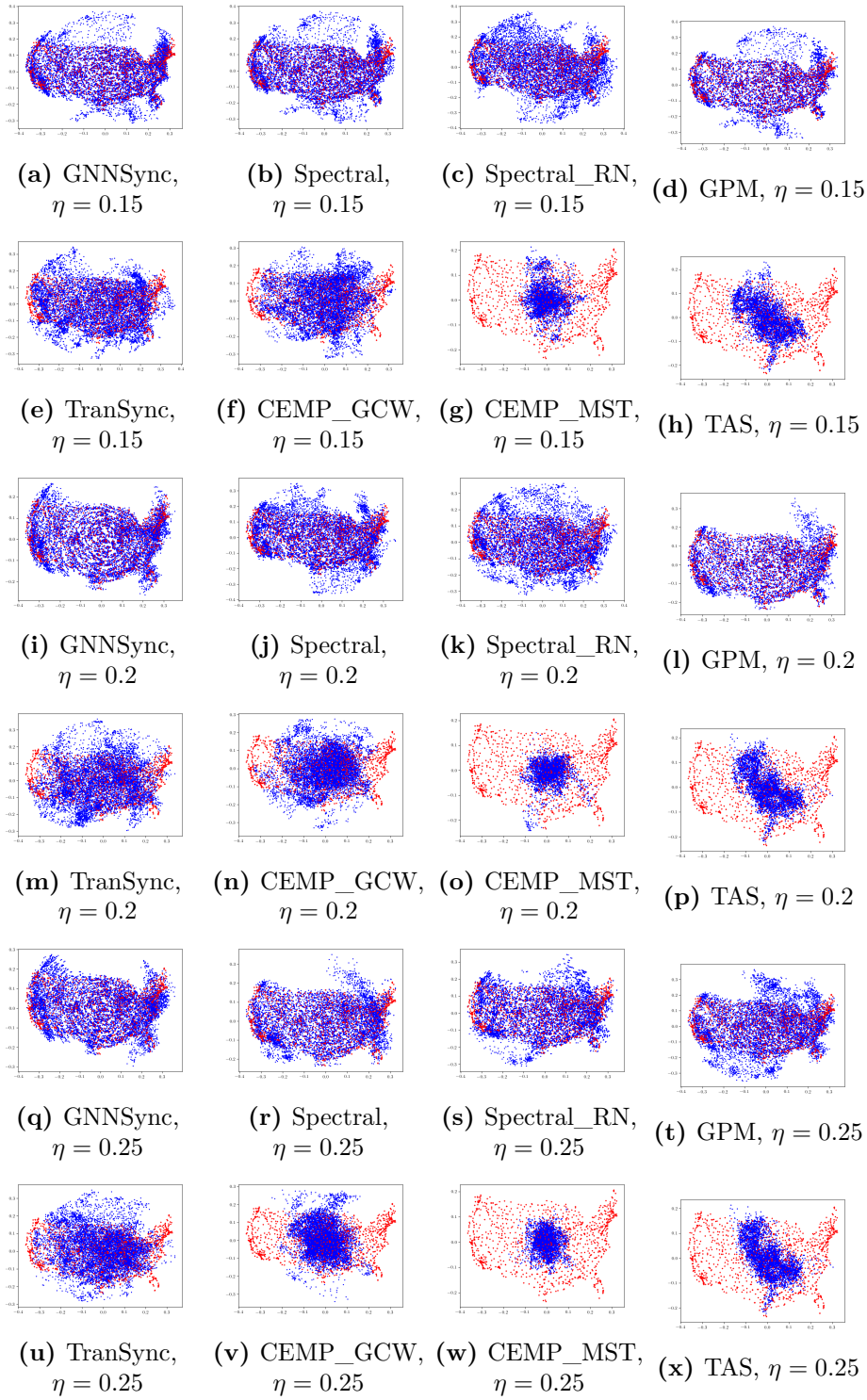
(i) GNNSync, $\eta = 0.05$ **(j)** Spectral, $\eta = 0.05$ **(k)** Spectral_RN, $\eta = 0.05$ **(l)** GPM, $\eta = 0.05$

(m) TranSync, $\eta = 0.05$ **(n)** CEMP_GCW, $\eta = 0.05$ **(o)** CEMP_MST, $\eta = 0.05$ **(p)** TAS, $\eta = 0.05$

(q) GNNSync, $\eta = 0.1$ **(r)** Spectral, $\eta = 0.1$ **(s)** Spectral_RN, $\eta = 0.1$ **(t)** GPM, $\eta = 0.1$

(u) TranSync, $\eta = 0.1$ **(v)** CEMP_GCW, $\eta = 0.1$ **(w)** CEMP_MST, $\eta = 0.1$ **(x)** TAS, $\eta = 0.1$

Figure E.14: Result visualization for the Sensor Network Localization task on the U.S. map using option “1" as ground-truth angles for low-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0.15$ **(b)** Spectral, $\eta = 0.15$ **(c)** Spectral_RN, $\eta = 0.15$ **(d)** GPM, $\eta = 0.15$

(e) TranSync, $\eta = 0.15$ **(f)** CEMP_GCW, $\eta = 0.15$ **(g)** CEMP_MST, $\eta = 0.15$ **(h)** TAS, $\eta = 0.15$

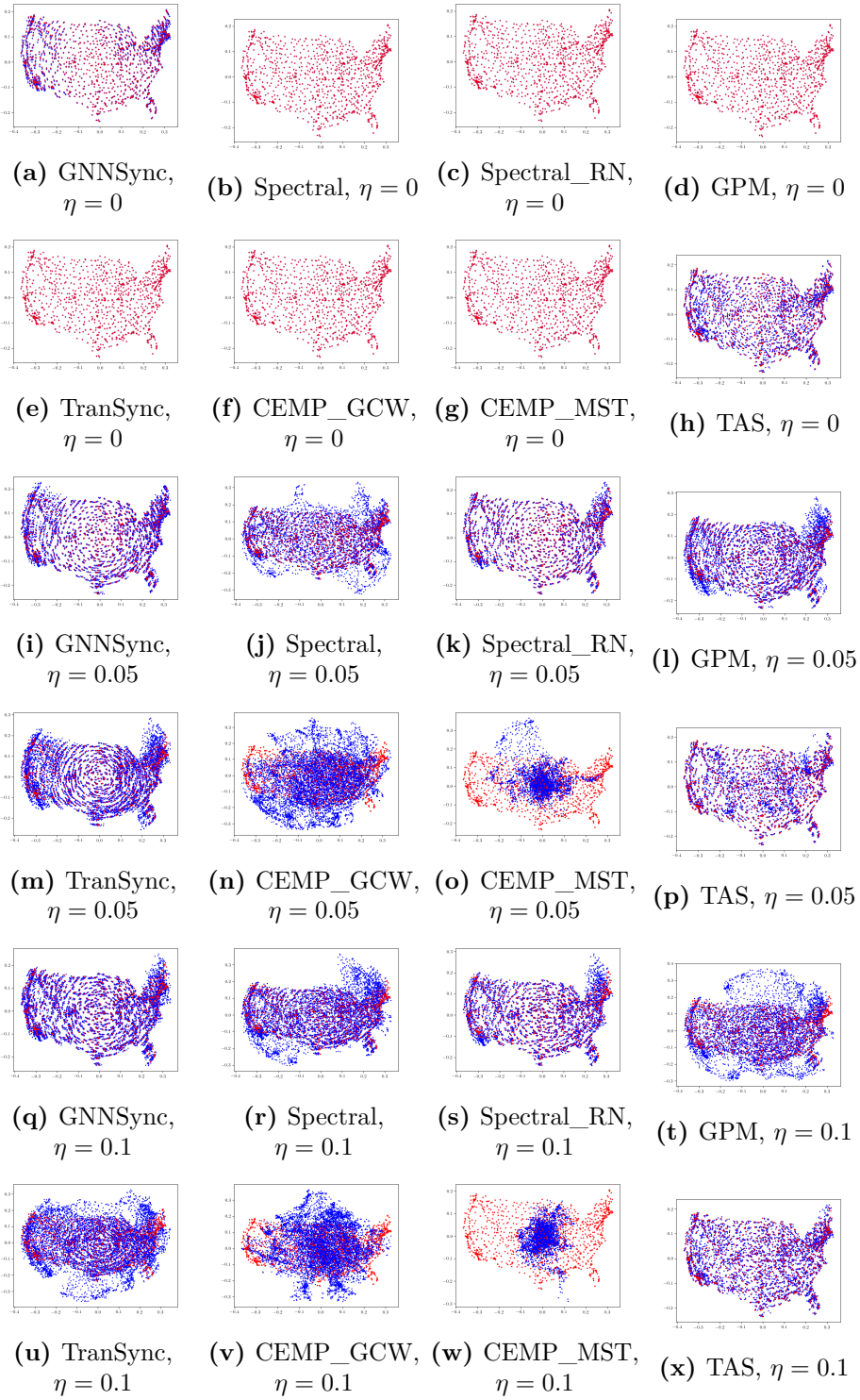
(i) GNNSync, $\eta = 0.2$ **(j)** Spectral, $\eta = 0.2$ **(k)** Spectral_RN, $\eta = 0.2$ **(l)** GPM, $\eta = 0.2$

(m) TranSync, $\eta = 0.2$ **(n)** CEMP_GCW, $\eta = 0.2$ **(o)** CEMP_MST, $\eta = 0.2$ **(p)** TAS, $\eta = 0.2$

(q) GNNSync, $\eta = 0.25$ **(r)** Spectral, $\eta = 0.25$ **(s)** Spectral_RN, $\eta = 0.25$ **(t)** GPM, $\eta = 0.25$

(u) TranSync, $\eta = 0.25$ **(v)** CEMP_GCW, $\eta = 0.25$ **(w)** CEMP_MST, $\eta = 0.25$ **(x)** TAS, $\eta = 0.25$

Figure E.15: Result visualization for the Sensor Network Localization task on the U.S. map using option “1" as ground-truth angles for high-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0$ **(b)** Spectral, $\eta = 0$ **(c)** Spectral_RN, $\eta = 0$ **(d)** GPM, $\eta = 0$

(e) TranSync, $\eta = 0$ **(f)** CEMP_GCW, $\eta = 0$ **(g)** CEMP_MST, $\eta = 0$ **(h)** TAS, $\eta = 0$

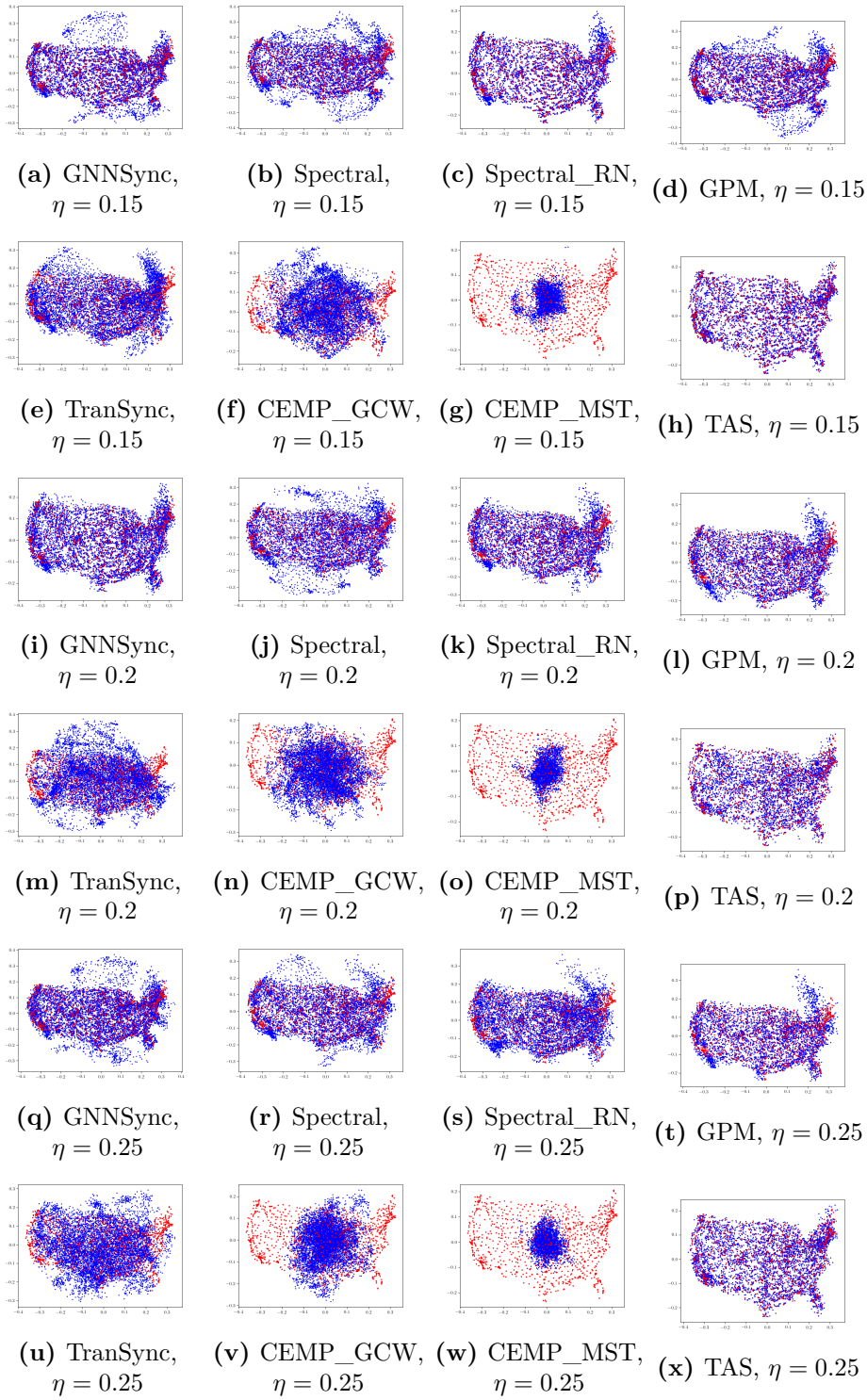
(i) GNNSync, $\eta = 0.05$ **(j)** Spectral, $\eta = 0.05$ **(k)** Spectral_RN, $\eta = 0.05$ **(l)** GPM, $\eta = 0.05$

(m) TranSync, $\eta = 0.05$ **(n)** CEMP_GCW, $\eta = 0.05$ **(o)** CEMP_MST, $\eta = 0.05$ **(p)** TAS, $\eta = 0.05$

(q) GNNSync, $\eta = 0.1$ **(r)** Spectral, $\eta = 0.1$ **(s)** Spectral_RN, $\eta = 0.1$ **(t)** GPM, $\eta = 0.1$

(u) TranSync, $\eta = 0.1$ **(v)** CEMP_GCW, $\eta = 0.1$ **(w)** CEMP_MST, $\eta = 0.1$ **(x)** TAS, $\eta = 0.1$

Figure E.16: Result visualization for the Sensor Network Localization task on the U.S. map using option “2" as ground-truth angles for low-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0.15$ **(b)** Spectral, $\eta = 0.15$ **(c)** Spectral_RN, $\eta = 0.15$ **(d)** GPM, $\eta = 0.15$

(e) TranSync, $\eta = 0.15$ **(f)** CEMP_GCW, $\eta = 0.15$ **(g)** CEMP_MST, $\eta = 0.15$ **(h)** TAS, $\eta = 0.15$

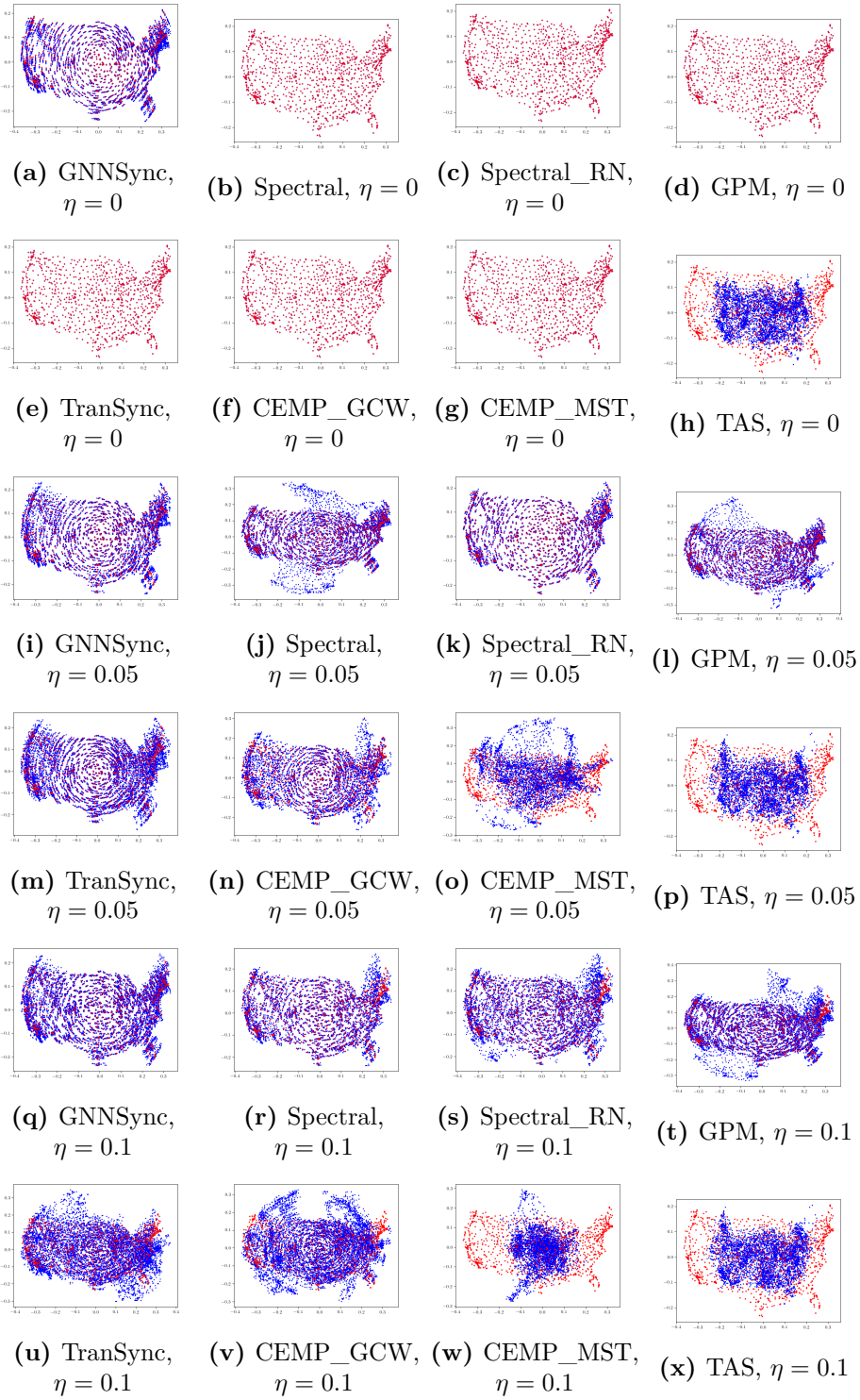
(i) GNNSync, $\eta = 0.2$ **(j)** Spectral, $\eta = 0.2$ **(k)** Spectral_RN, $\eta = 0.2$ **(l)** GPM, $\eta = 0.2$

(m) TranSync, $\eta = 0.2$ **(n)** CEMP_GCW, $\eta = 0.2$ **(o)** CEMP_MST, $\eta = 0.2$ **(p)** TAS, $\eta = 0.2$

(q) GNNSync, $\eta = 0.25$ **(r)** Spectral, $\eta = 0.25$ **(s)** Spectral_RN, $\eta = 0.25$ **(t)** GPM, $\eta = 0.25$

(u) TranSync, $\eta = 0.25$ **(v)** CEMP_GCW, $\eta = 0.25$ **(w)** CEMP_MST, $\eta = 0.25$ **(x)** TAS, $\eta = 0.25$

Figure E.17: Result visualization for the Sensor Network Localization task on the U.S. map using option “2" as ground-truth angles for high-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0$ **(b)** Spectral, $\eta = 0$ **(c)** Spectral_RN, $\eta = 0$ **(d)** GPM, $\eta = 0$

(e) TranSync, $\eta = 0$ **(f)** CEMP_GCW, $\eta = 0$ **(g)** CEMP_MST, $\eta = 0$ **(h)** TAS, $\eta = 0$

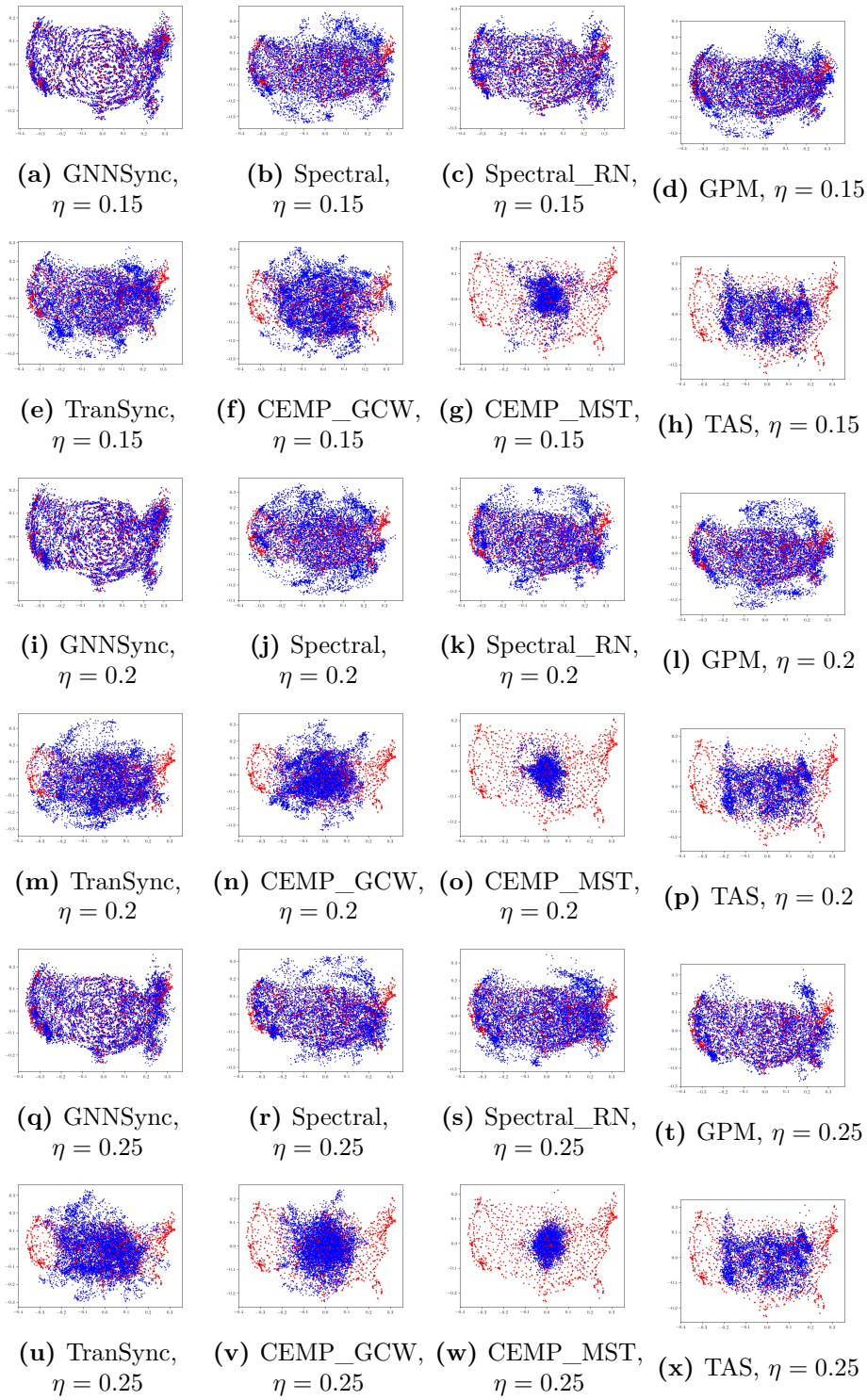
(i) GNNSync, $\eta = 0.05$ **(j)** Spectral, $\eta = 0.05$ **(k)** Spectral_RN, $\eta = 0.05$ **(l)** GPM, $\eta = 0.05$

(m) TranSync, $\eta = 0.05$ **(n)** CEMP_GCW, $\eta = 0.05$ **(o)** CEMP_MST, $\eta = 0.05$ **(p)** TAS, $\eta = 0.05$

(q) GNNSync, $\eta = 0.1$ **(r)** Spectral, $\eta = 0.1$ **(s)** Spectral_RN, $\eta = 0.1$ **(t)** GPM, $\eta = 0.1$

(u) TranSync, $\eta = 0.1$ **(v)** CEMP_GCW, $\eta = 0.1$ **(w)** CEMP_MST, $\eta = 0.1$ **(x)** TAS, $\eta = 0.1$

Figure E.18: Result visualization for the Sensor Network Localization task on the U.S. map using option “3" as ground-truth angles for low-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0.15$ **(b)** Spectral, $\eta = 0.15$ **(c)** Spectral_RN, $\eta = 0.15$ **(d)** GPM, $\eta = 0.15$

(e) TranSync, $\eta = 0.15$ **(f)** CEMP_GCW, $\eta = 0.15$ **(g)** CEMP_MST, $\eta = 0.15$ **(h)** TAS, $\eta = 0.15$

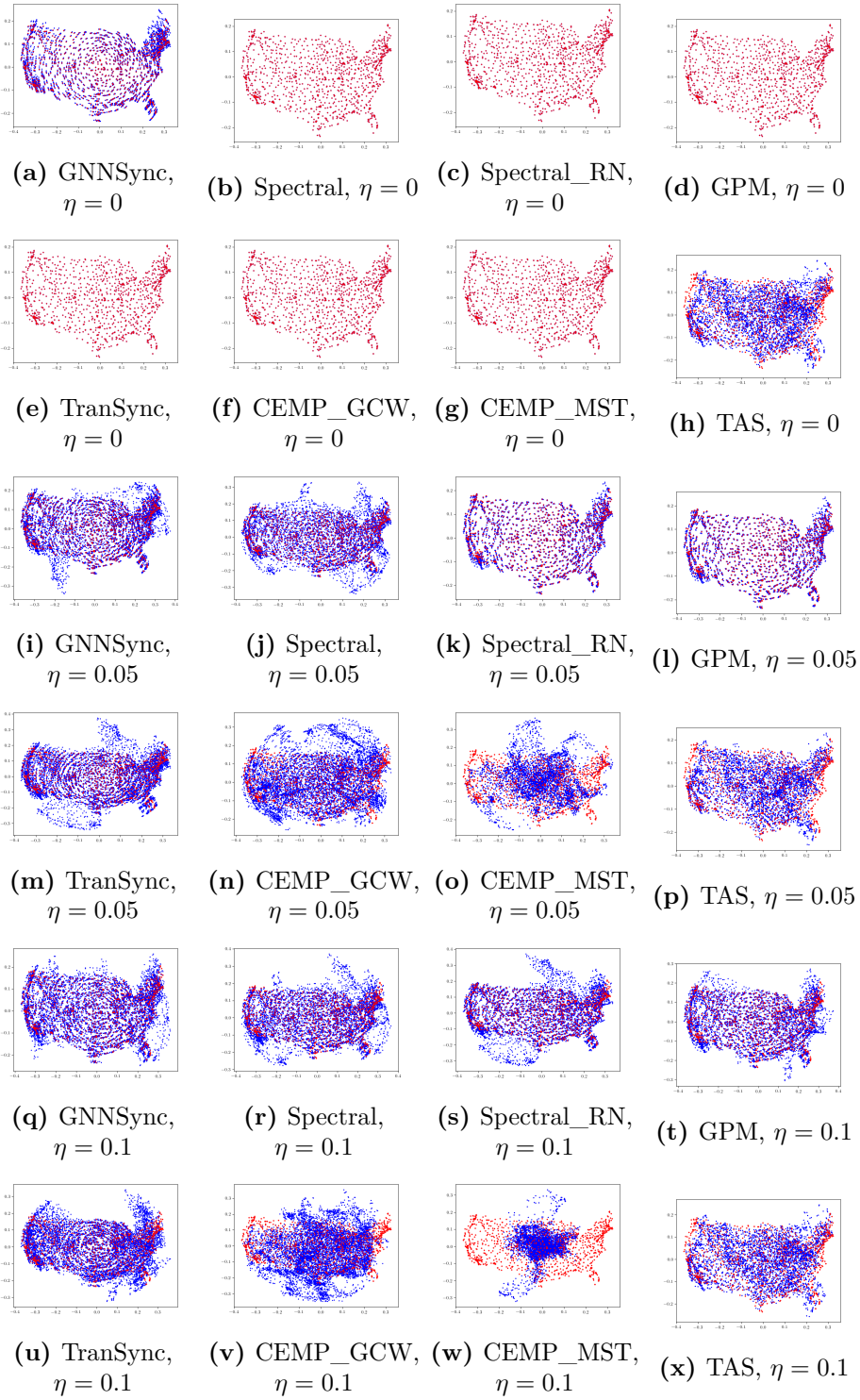
(i) GNNSync, $\eta = 0.2$ **(j)** Spectral, $\eta = 0.2$ **(k)** Spectral_RN, $\eta = 0.2$ **(l)** GPM, $\eta = 0.2$

(m) TranSync, $\eta = 0.2$ **(n)** CEMP_GCW, $\eta = 0.2$ **(o)** CEMP_MST, $\eta = 0.2$ **(p)** TAS, $\eta = 0.2$

(q) GNNSync, $\eta = 0.25$ **(r)** Spectral, $\eta = 0.25$ **(s)** Spectral_RN, $\eta = 0.25$ **(t)** GPM, $\eta = 0.25$

(u) TranSync, $\eta = 0.25$ **(v)** CEMP_GCW, $\eta = 0.25$ **(w)** CEMP_MST, $\eta = 0.25$ **(x)** TAS, $\eta = 0.25$

Figure E.19: Result visualization for the Sensor Network Localization task on the U.S. map using option “3" as ground-truth angles for high-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0$ **(b)** Spectral, $\eta = 0$ **(c)** Spectral_RN, $\eta = 0$ **(d)** GPM, $\eta = 0$

(e) TranSync, $\eta = 0$ **(f)** CEMP_GCW, $\eta = 0$ **(g)** CEMP_MST, $\eta = 0$ **(h)** TAS, $\eta = 0$

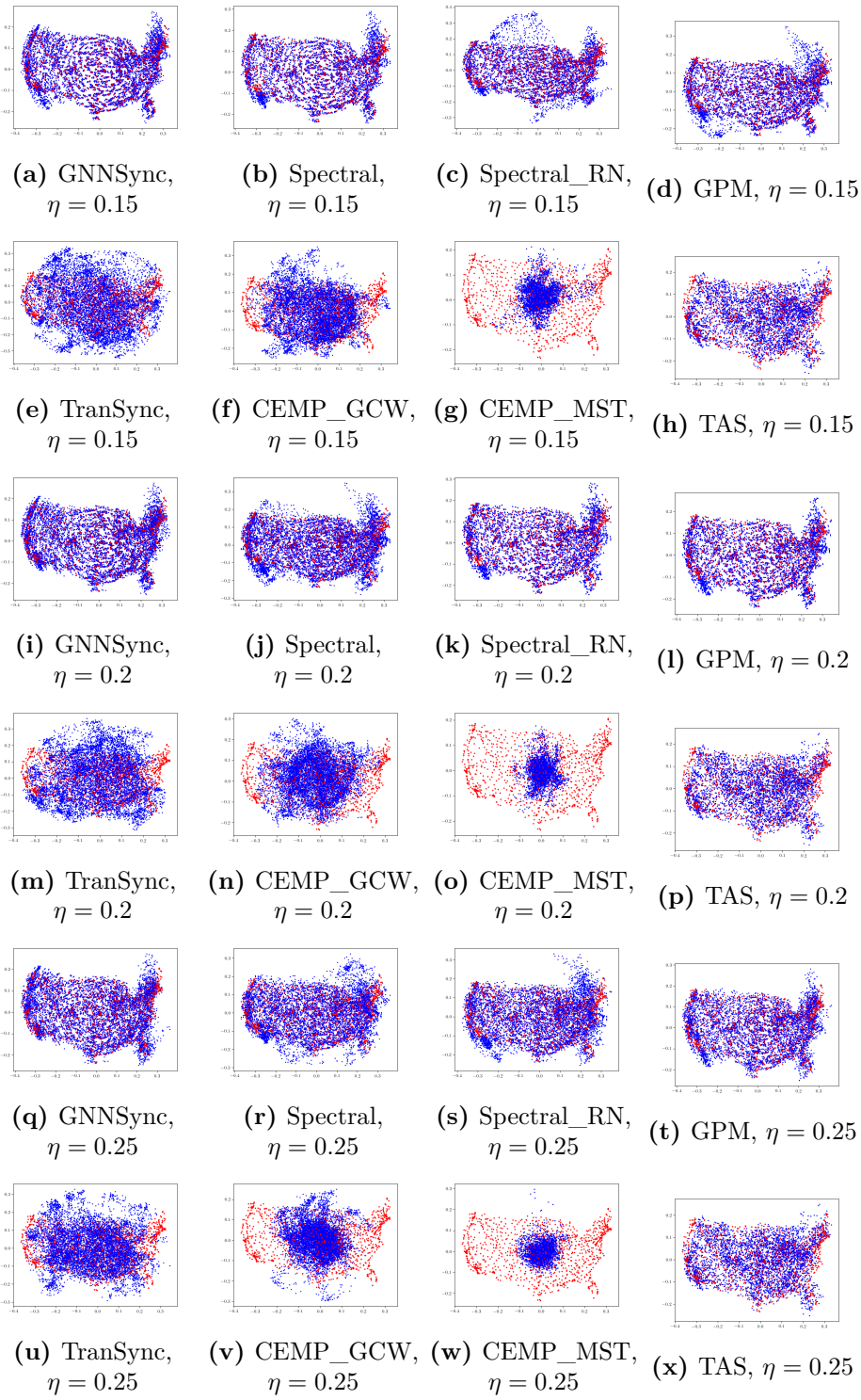
(i) GNNSync, $\eta = 0.05$ **(j)** Spectral, $\eta = 0.05$ **(k)** Spectral_RN, $\eta = 0.05$ **(l)** GPM, $\eta = 0.05$

(m) TranSync, $\eta = 0.05$ **(n)** CEMP_GCW, $\eta = 0.05$ **(o)** CEMP_MST, $\eta = 0.05$ **(p)** TAS, $\eta = 0.05$

(q) GNNSync, $\eta = 0.1$ **(r)** Spectral, $\eta = 0.1$ **(s)** Spectral_RN, $\eta = 0.1$ **(t)** GPM, $\eta = 0.1$

(u) TranSync, $\eta = 0.1$ **(v)** CEMP_GCW, $\eta = 0.1$ **(w)** CEMP_MST, $\eta = 0.1$ **(x)** TAS, $\eta = 0.1$

Figure E.20: Result visualization for the Sensor Network Localization task on the U.S. map using option “4" as ground-truth angles for low-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0.15$ **(b)** Spectral, $\eta = 0.15$ **(c)** Spectral_RN, $\eta = 0.15$ **(d)** GPM, $\eta = 0.15$

(e) TranSync, $\eta = 0.15$ **(f)** CEMP_GCW, $\eta = 0.15$ **(g)** CEMP_MST, $\eta = 0.15$ **(h)** TAS, $\eta = 0.15$

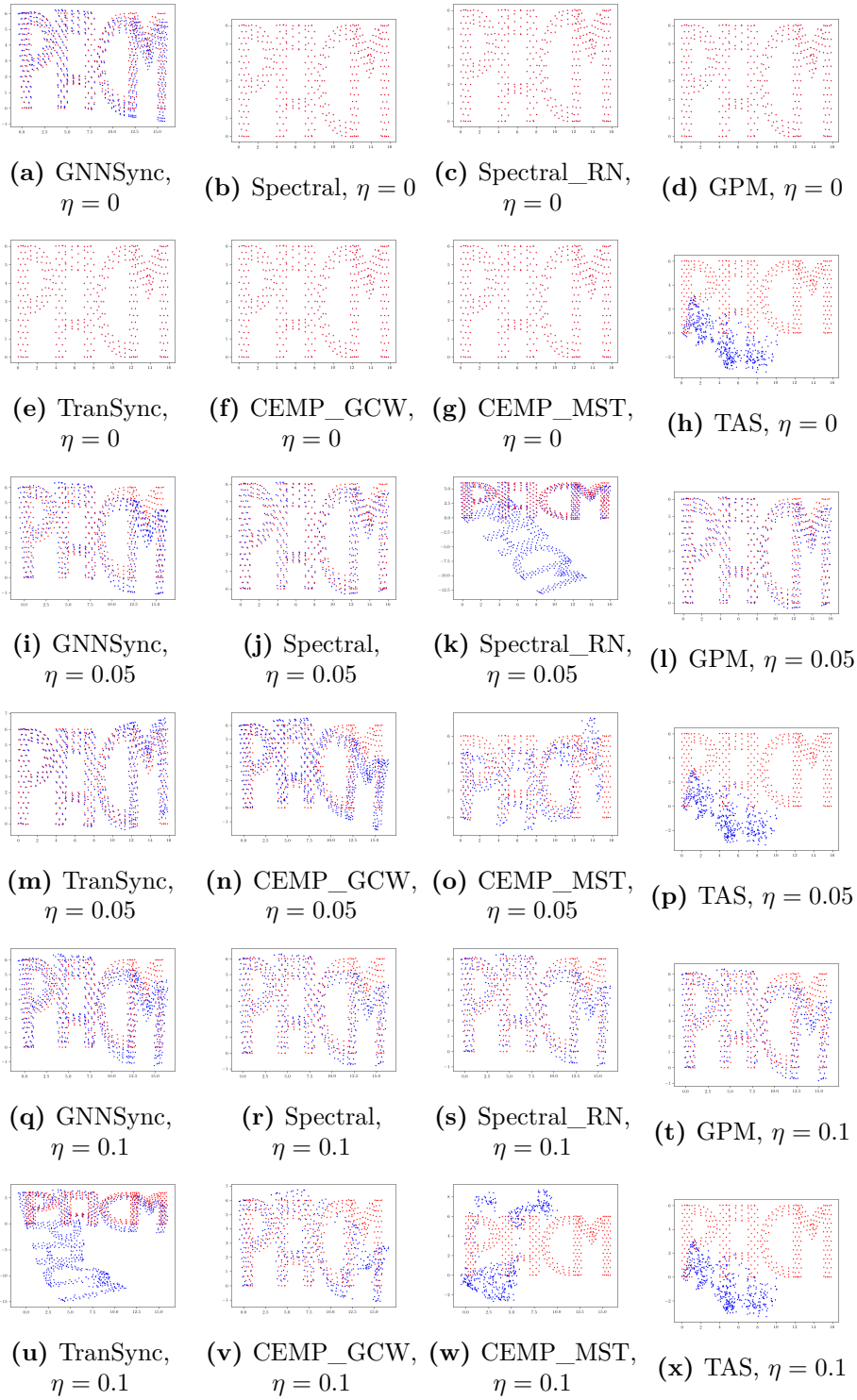
(i) GNNSync, $\eta = 0.2$ **(j)** Spectral, $\eta = 0.2$ **(k)** Spectral_RN, $\eta = 0.2$ **(l)** GPM, $\eta = 0.2$

(m) TranSync, $\eta = 0.2$ **(n)** CEMP_GCW, $\eta = 0.2$ **(o)** CEMP_MST, $\eta = 0.2$ **(p)** TAS, $\eta = 0.2$

(q) GNNSync, $\eta = 0.25$ **(r)** Spectral, $\eta = 0.25$ **(s)** Spectral_RN, $\eta = 0.25$ **(t)** GPM, $\eta = 0.25$

(u) TranSync, $\eta = 0.25$ **(v)** CEMP_GCW, $\eta = 0.25$ **(w)** CEMP_MST, $\eta = 0.25$ **(x)** TAS, $\eta = 0.25$

Figure E.21: Result visualization for the Sensor Network Localization task on the U.S. map using option “4" as ground-truth angles for high-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0$ **(b)** Spectral, $\eta = 0$ **(c)** Spectral_RN, $\eta = 0$ **(d)** GPM, $\eta = 0$

(e) TranSync, $\eta = 0$ **(f)** CEMP_GCW, $\eta = 0$ **(g)** CEMP_MST, $\eta = 0$ **(h)** TAS, $\eta = 0$

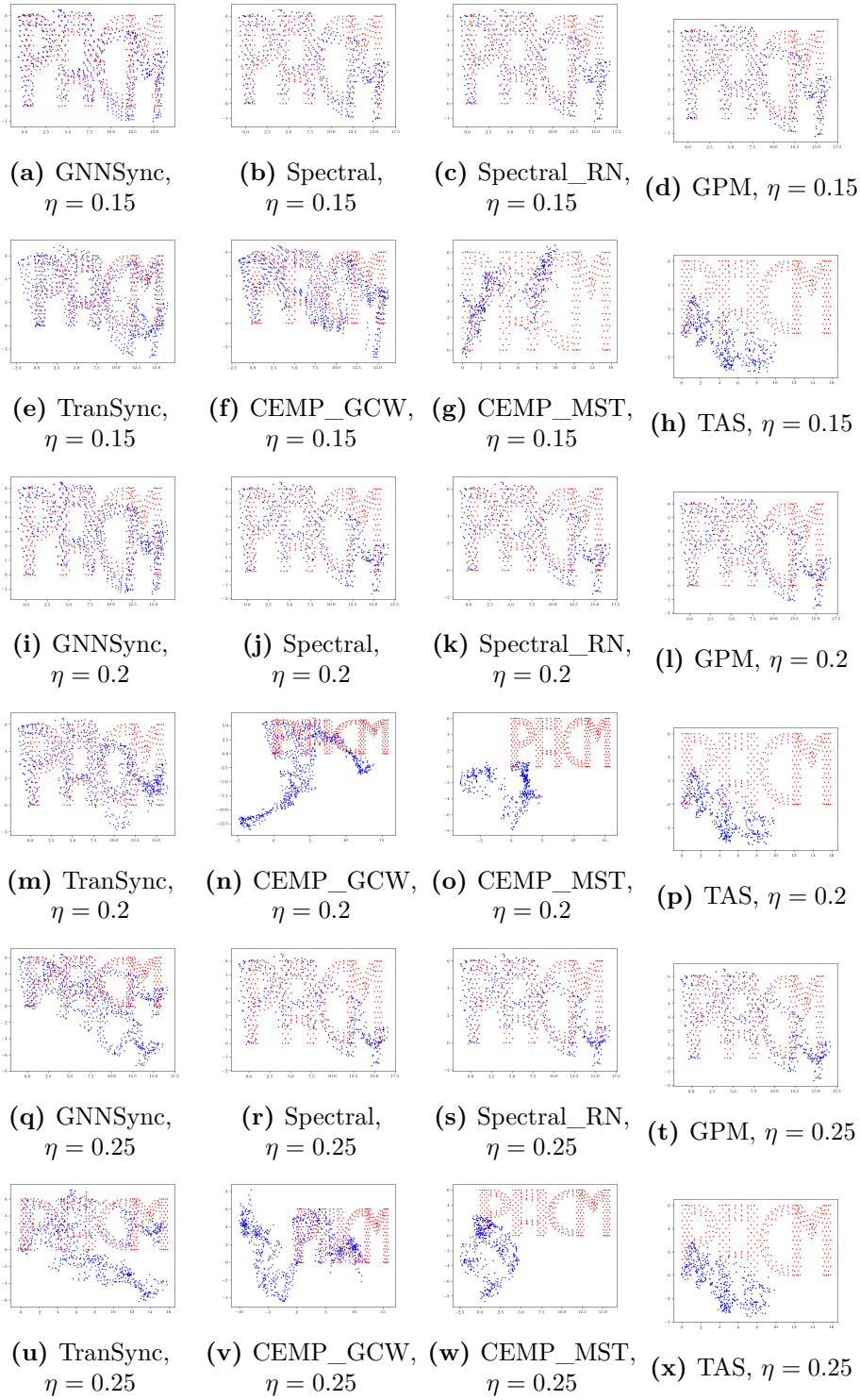
(i) GNNSync, $\eta = 0.05$ **(j)** Spectral, $\eta = 0.05$ **(k)** Spectral_RN, $\eta = 0.05$ **(l)** GPM, $\eta = 0.05$

(m) TranSync, $\eta = 0.05$ **(n)** CEMP_GCW, $\eta = 0.05$ **(o)** CEMP_MST, $\eta = 0.05$ **(p)** TAS, $\eta = 0.05$

(q) GNNSync, $\eta = 0.1$ **(r)** Spectral, $\eta = 0.1$ **(s)** Spectral_RN, $\eta = 0.1$ **(t)** GPM, $\eta = 0.1$

(u) TranSync, $\eta = 0.1$ **(v)** CEMP_GCW, $\eta = 0.1$ **(w)** CEMP_MST, $\eta = 0.1$ **(x)** TAS, $\eta = 0.1$

Figure E.22: Result visualization for the Sensor Network Localization task on the PACM point cloud using option “1" as ground-truth angles for low-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.



(a) GNNSync, $\eta = 0.15$ **(b)** Spectral, $\eta = 0.15$ **(c)** Spectral_RN, $\eta = 0.15$ **(d)** GPM, $\eta = 0.15$

(e) TranSync, $\eta = 0.15$ **(f)** CEMP_GCW, $\eta = 0.15$ **(g)** CEMP_MST, $\eta = 0.15$ **(h)** TAS, $\eta = 0.15$

(i) GNNSync, $\eta = 0.2$ **(j)** Spectral, $\eta = 0.2$ **(k)** Spectral_RN, $\eta = 0.2$ **(l)** GPM, $\eta = 0.2$

(m) TranSync, $\eta = 0.2$ **(n)** CEMP_GCW, $\eta = 0.2$ **(o)** CEMP_MST, $\eta = 0.2$ **(p)** TAS, $\eta = 0.2$

(q) GNNSync, $\eta = 0.25$ **(r)** Spectral, $\eta = 0.25$ **(s)** Spectral_RN, $\eta = 0.25$ **(t)** GPM, $\eta = 0.25$

(u) TranSync, $\eta = 0.25$ **(v)** CEMP_GCW, $\eta = 0.25$ **(w)** CEMP_MST, $\eta = 0.25$ **(x)** TAS, $\eta = 0.25$

Figure E.23: Result visualization for the Sensor Network Localization task on the PACM point cloud using option "1" as ground-truth angles for high-noise input data. Red dots indicate ground-truth locations and blue dots are estimated city locations.

E.4.2 Extended ablation study results

Ablation study results are reported in Fig. E.24 and E.25, while the rest are omitted but could lead to the same conclusion. Note that for $k > 1$, we ablation study results are based on using $\mathcal{L}_{\text{cycle}}$ as the training loss function.

Improvements over all possible baselines when taking their output as input features for $k = 1$ are reported in Fig. E.26, E.27 and E.28, where we omit results for $\eta = 0.9$ as in general all methods fall behind the trivial solution at $\eta = 0.9$. We find that in most cases GNNSync could improve over baselines, and could do worse often only when all methods fall behind the trivial baseline.

To show the effect of a linear combination of L_{cycle} and L_{upset}, we empirically test $L_{\text{cycle}} + \tau L_{\text{upset}}$, with τ varying from 0 to 0.9; see Fig. E.29 (the others are omitted but could lead to the same conclusion) for details. The performance for different choices of τ do not vary significantly, providing further evidence that it suffices to simply pay attention to either of the two loss functions instead of their linear combination. The experiments also show that as the problem becomes harder (e.g. as the noise level increases and the network becomes sparser), a smaller coefficient of L_{upset} (even zero) is preferred, which indicates that L_{cycle} plays a more essential role in the more challenging scenarios.

To assess the effect of fine-tuning (via projected gradient steps) over the baselines, we apply the same number of projected gradient descent steps as GNNSync to the comparative baselines and report the performance in Figures E.30 and E.31. We observe that even when applying these fine-tuning steps, the baselines are usually beaten by our end-to-end trainable GNNSync pipeline.

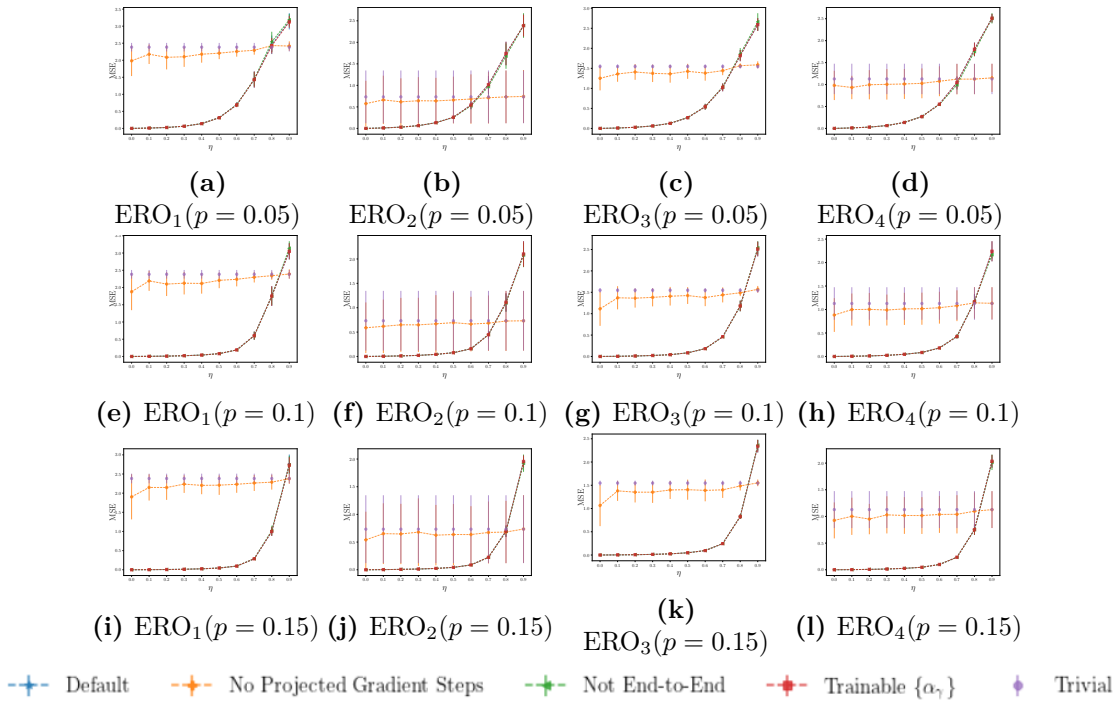


Figure E.24: MSE performance comparison on GNNSync variants on angular synchronization ($k = 1$) for ERO models. p is the network density and η is the noise level. Error bars indicate one standard deviation.

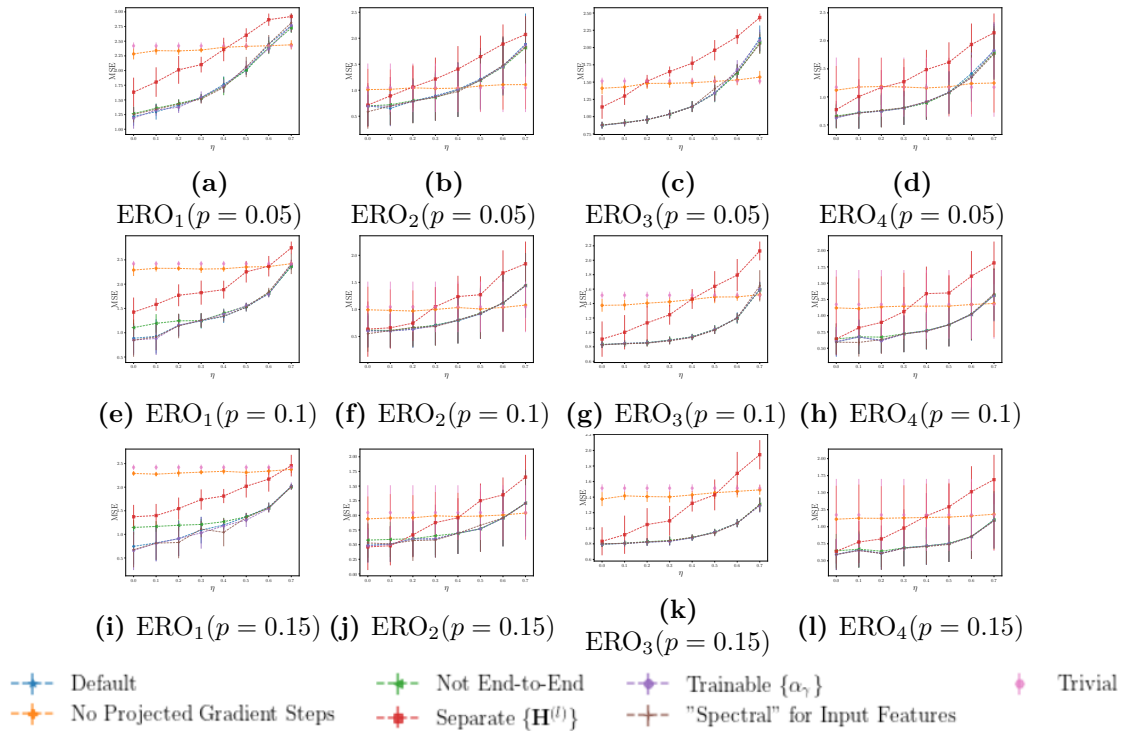


Figure E.25: MSE performance comparison on GNNSync variants on $k-$synchronization with $k = 2$ for ERO models. p is the network density and η is the noise level. Error bars indicate one standard deviation.

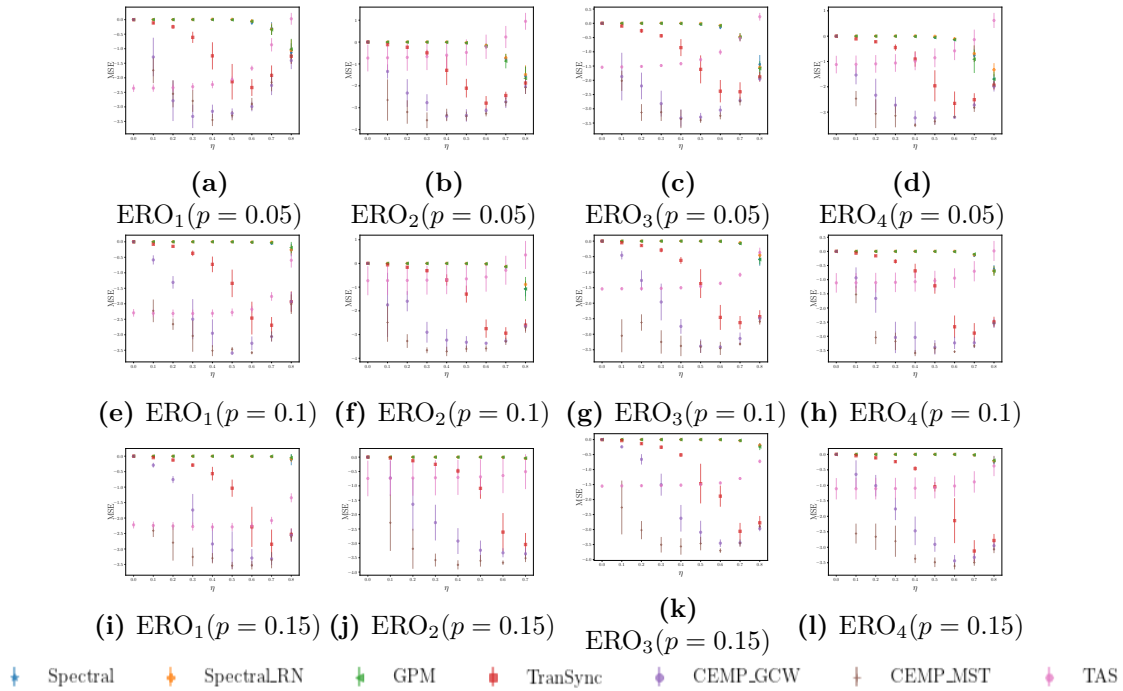


Figure E.26: MSE performance improvement on GNNSync over variants on angular synchronization ($k = 1$) for ERO models. Error bars indicate one standard deviation.

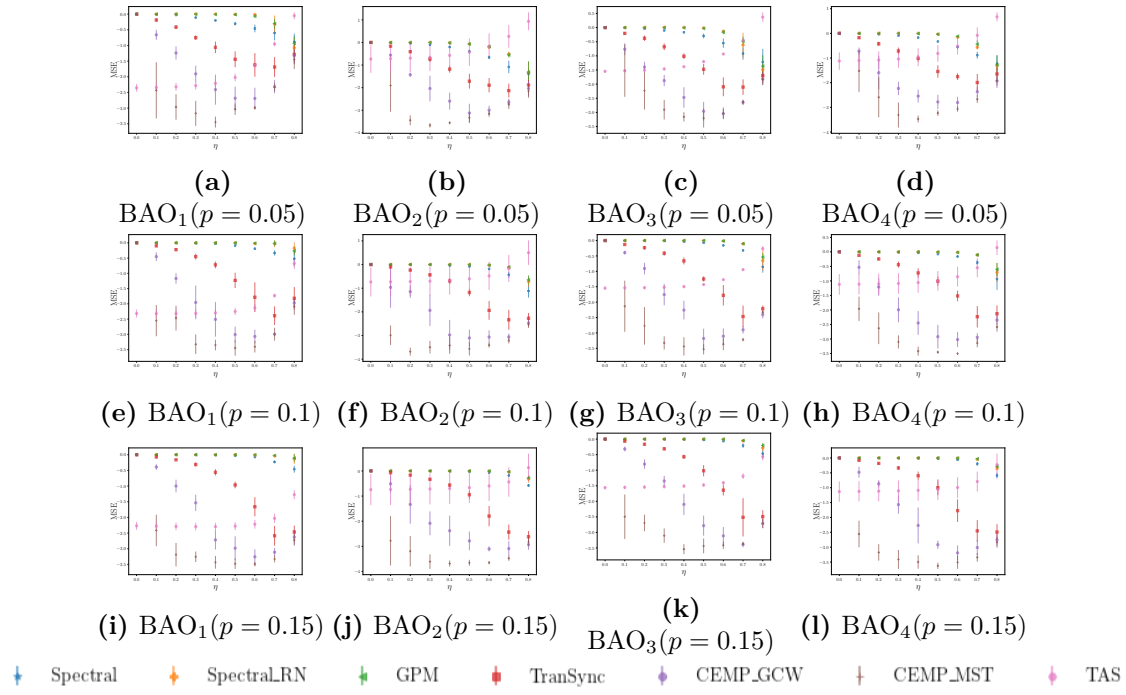


Figure E.27: MSE performance improvement on GNNSync over variants on angular synchronization ($k = 1$) for BAO models. Error bars indicate one standard deviation.

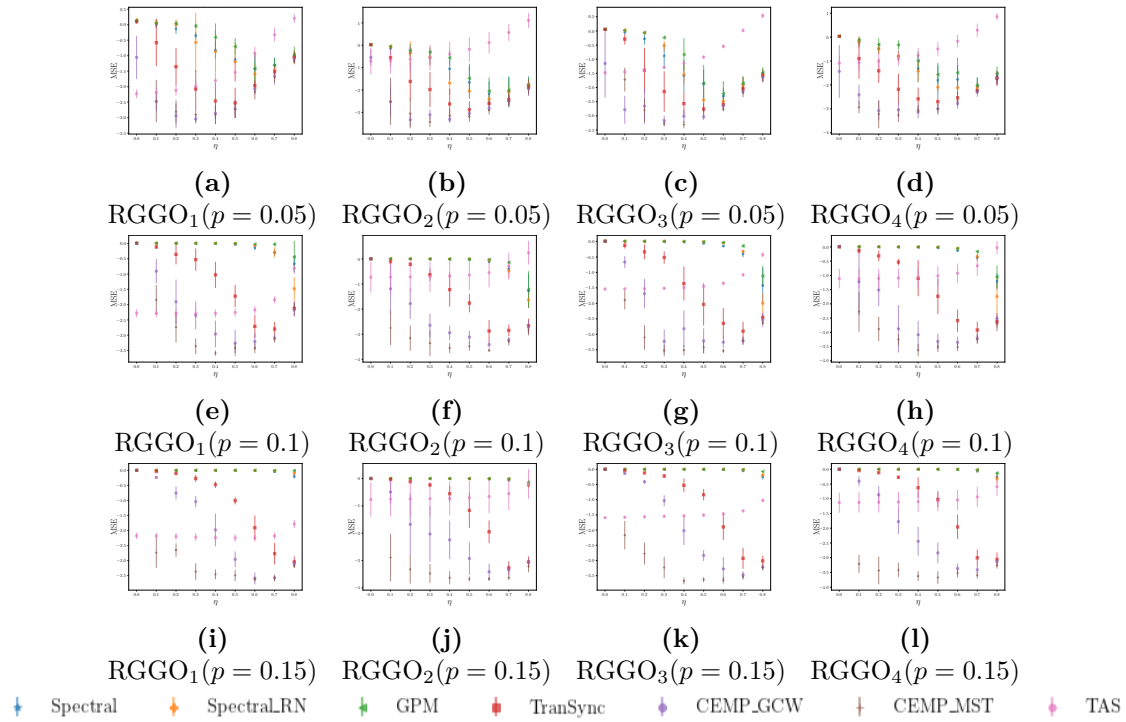


Figure E.28: MSE performance improvement on GNNSync over variants on angular synchronization ($k = 1$) for RGGO models. Error bars indicate one standard deviation.

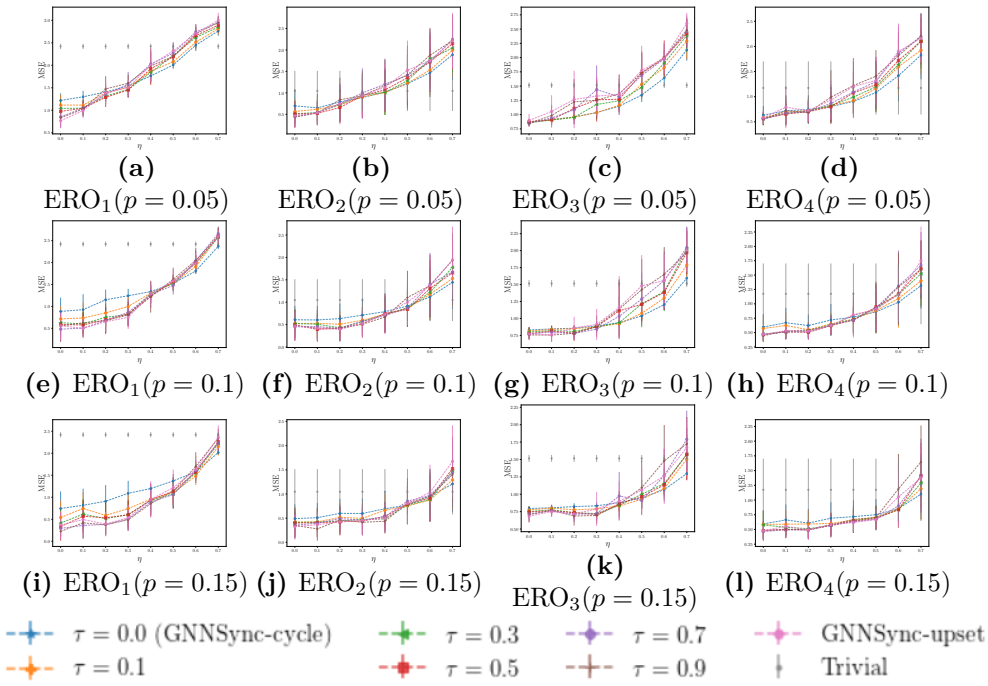


Figure E.29: MSE comparison on GNNSync variants using as loss $\tau\mathcal{L}_{\text{upset}} + \mathcal{L}_{\text{cycle}}$ with different coefficients τ, on $k-$synchronization with $k = 2$ for ERO models. p is the network density and η is the noise level. Error bars indicate one standard deviation.

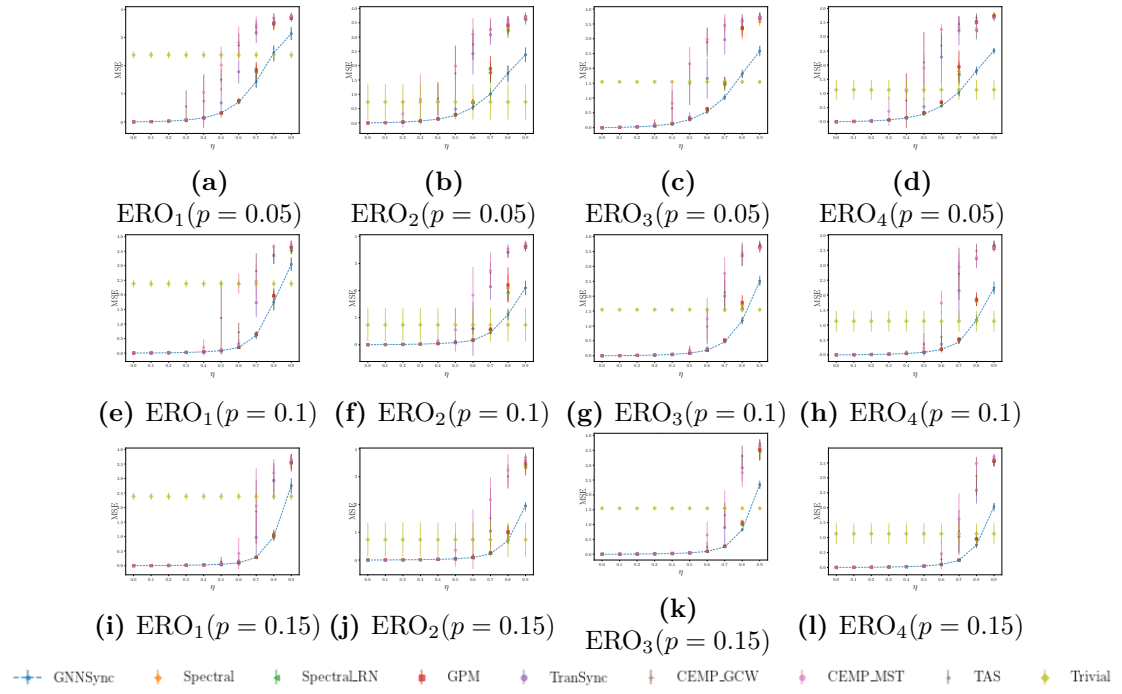


Figure E.30: MSE performance comparison on GNNSync against fine-tuned baselines on angular synchronization ($k = 1$) for ERO models. p is the network density and η is the noise level. Error bars indicate one standard deviation.

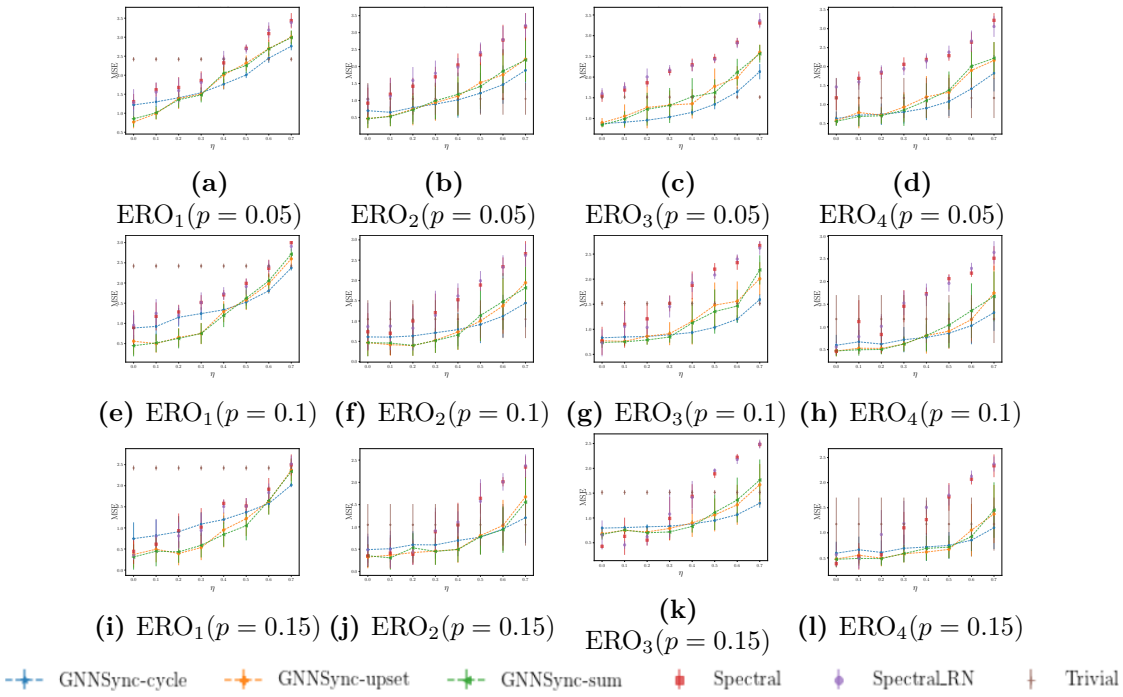


Figure E.31: MSE performance comparison on GNNSync against fine-tuned baselines on $k-$synchronization with $k = 2$ for ERO models. p is the network density and η is the noise level. Error bars indicate one standard deviation.