

© 2021 Aravind Sankar

SPARSITY-AWARE NEURAL USER BEHAVIOR MODELING IN ONLINE
INTERACTION PLATFORMS

BY

ARAVIND SANKAR

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Hari Sundaram, Chair
Professor Jiawei Han
Associate Professor Hanghang Tong
Dr. Neil Shah, Senior Research Scientist at Snap Inc.

ABSTRACT

Modern online platforms offer users an opportunity to participate in a variety of content-creation, social networking, and shopping activities. With the rapid proliferation of such online services, learning data-driven user behavior models is indispensable to enable personalized user experiences. Recently, representation learning has emerged as an effective strategy for user modeling, powered by neural networks trained over large volumes of interaction data. Despite their enormous potential, we encounter the unique challenge of data sparsity for a vast majority of entities, *e.g.*, sparsity in ground-truth labels for entities and in entity-level interactions (cold-start users, items in the long-tail, and ephemeral groups).

In this dissertation, we develop generalizable neural representation learning frameworks for user behavior modeling designed to address different sparsity challenges across applications. Our problem settings span transductive and inductive learning scenarios, where transductive learning models entities seen during training and inductive learning targets entities that are only observed during inference. We leverage different facets of information reflecting user behavior (*e.g.*, interconnectivity in social networks, temporal and attributed interaction information) to enable personalized inference at scale. Our proposed models are complementary to concurrent advances in neural architectural choices and are adaptive to the rapid addition of new applications in online platforms.

First, we examine two transductive learning settings: inference and recommendation in *graph-structured* and *bipartite user-item* interactions. In chapter 3, we formulate user profiling in social platforms as semi-supervised learning over graphs given sparse ground-truth labels for node attributes. We present a graph neural network framework that exploits higher-order connectivity structures (network motifs) to learn attributed structural roles of nodes that identify structurally similar nodes with co-varying local attributes. In chapter 4, we design neural collaborative filtering models for few-shot recommendations over user-item interactions. To address item interaction sparsity due to heavy-tailed distributions, our proposed meta-learning framework learns-to-recommend few-shot items by knowledge transfer from arbitrary base recommenders. We show that our framework consistently outperforms state-of-art approaches on overall recommendation (by 5% Recall) while achieving significant gains (of 60-80% Recall) for tail items with fewer than 20 interactions.

Next, we explored three inductive learning settings: modeling spread of *user-generated content* in social networks; item recommendations for *ephemeral groups*; and *friend ranking* in large-scale social platforms. In chapter 5, we focus on diffusion prediction in social net-

works where a vast population of users rarely post content. We introduce a deep generative modeling framework that models users as probability distributions in the latent space with variational priors parameterized by graph neural networks. Our approach enables massive performance gains (over 150% recall) for users with sparse activities, while being faster than state-of-the-art neural models by an order of magnitude. In chapter 6, we examine item recommendations for ephemeral groups with limited or no historical interactions together. To overcome group interaction sparsity, we present self-supervised learning strategies that exploit the preference co-variance in observed group memberships for group recommender training. Our framework achieves significant performance gains (over 30% NDCG) over prior state-of-the-art group recommendation models. In chapter 7, we introduce multi-modal inference with graph neural networks that captures knowledge from multiple feature modalities and user interactions for multi-faceted friend ranking. Our approach achieves notable higher performance gains for critical populations of less-active and low-degree users.

ACKNOWLEDGMENTS

There are so many people who contributed to my Ph.D. journey at the University of Illinois, Urbana-Champaign (UIUC).

First and foremost, I would like to thank my advisor, Professor Hari Sundaram, whose guidance and encouragement has been instrumental in shaping this dissertation. I am ever-grateful to Hari for taking me as a student, his kindness and genuine interest in my mental and physical well-being meant a lot to me during my initial Ph.D. years. Hari taught me invaluable art of technical writing – the ability to express sophisticated concepts in a simplified and accessible language without missing out on the essential details. His guidance has greatly influenced how I envision the big picture and impact of a research problem, and enhanced my ability to relate and connect ideas across different research areas. He has also been incredibly supportive of my independent research pursuits with other professors and external collaborators. Over the years, he has been integral in shaping me into the researcher that I have become, and I would be fortunate to work with him again in the future.

I also want to thank my other thesis committee members, Professor Jiawei Han, Professor Hanghang Tong, and Dr. Neil Shah. I would especially like to thank Jiawei for his kind words of encouragement during my uncertain moments. His courses laid the foundation for my foray into data mining, and my numerous collaborations with him and other students from his research group have contributed immensely to different threads of work in this thesis. I am thankful to Hanghang for his insightful questions and feedback during my thesis proposal, which were greatly instructive and helpful in the direction of this work. I am also immensely grateful to Neil Shah for his guidance during my internship at Snap, which motivated me to explore practically impactful implications of my research beyond publishing objectives. Despite the fully remote internship experience, I had an awesome time working with Neil, and got to know how much fun research in the industry can be.

I would also like to thank Yanhong Wu for his guidance and encouragement during my two summers that I spent at Visa Research. Yanhong went above and beyond his mandated responsibilities to mentor me, and spent countless hours outside of his regular work schedule to assist me with additional experiments and paper writing after the end of my internship.

I'm extremely fortunate to have been advised by my undergraduate thesis advisors Professors Sayan Ranu and Karthik Raman at Indian Institute of Technology Madras (IIT-Madras). Even after I moved to the U.S. for my Ph.D. studies, they actively continued mentoring me to ensure that our work reached its best possible conclusion with a publica-

tion in a top bioinformatics journal. Without their efforts in instructing and mentoring me at different levels, I could not have possibly had the opportunities that led me to UIUC.

I would also like to acknowledge my colleagues at Crowd Dynamics Lab for their numerous research discussions, constructive feedback, and collaborations that have shaped my work. A big shout out to my research collaborators Junting Wang, Adit Krishnan, and Xinyang Zhang, who have closely worked with me on several projects.

Finally, I am forever indebted to my parents, sister, and brother-in-law for their unwavering love and support through the ups and downs of my Ph.D. journey.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	Introduction	1
1.2	Technical Challenges	2
1.3	Transductive User Behavior Modeling	3
1.4	Inductive User Behavior Modeling	5
1.5	Thesis Organization	8
CHAPTER 2	LITERATURE REVIEW	10
2.1	Social Network Mining	10
2.2	Recommender Systems	14
2.3	Relevant Machine Learning Paradigms	18
CHAPTER 3	INFOMOTIF: HIGHER-ORDER STRUCTURES IN GRAPH-BASED INTERACTIONS	21
3.1	Introduction	21
3.2	Preliminaries	24
3.3	InfoMotif Framework	27
3.4	Model Details	34
3.5	Experiments	36
3.6	Discussion	48
3.7	Related Work	49
3.8	Conclusion	51
CHAPTER 4	PROTOCF: FEW-SHOT COLLABORATIVE FILTERING VIA META-TRANSFER	52
4.1	Introduction	52
4.2	Problem Definition	55
4.3	Related Work	56
4.4	ProtoCF Framework	58
4.5	Experiments	68
4.6	Conclusion	75
CHAPTER 5	INFVAE: SOCIAL REGULARIZATION TO PREDICT INFORMATION DIFFUSION	77
5.1	Introduction	77
5.2	Related Work	79
5.3	Problem Definition	80
5.4	INFVAE Framework	81
5.5	Experiments	92
5.6	Acknowledgements	101
5.7	Conclusion	102

CHAPTER 6	GROUPIM: SELF-SUPERVISED EPHEMERAL GROUP INTERACTION MODELING	103
6.1	Introduction	103
6.2	Related Work	105
6.3	Preliminaries	106
6.4	GROUPIM Framework	111
6.5	Experiments	117
6.6	Conclusion	128
CHAPTER 7	GRAFRANK: MULTI-FACETED FRIEND RANKING IN SOCIAL PLATFORMS	130
7.1	Introduction	130
7.2	Related Work	132
7.3	Preliminaries	133
7.4	Graph Neural Friend Ranking	137
7.5	Experiments	146
7.6	Conclusion	157
CHAPTER 8	CONCLUSIONS AND FUTURE DIRECTIONS	158
8.1	Research Contributions	158
8.2	Future Work	163
REFERENCES	168

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

In recent times, users participate in a multitude of content-creation, social networking and e-commerce platforms, with diverse interactional settings. User behavior is characterized by their activities and interactions with other users and functionalities within the platform. For instance, users in social networking platforms (such as Facebook and Snapchat) form friendships where they communicate with friends, post or re-share user-generated content, and interact in diverse groups. In e-commerce platforms (such as Amazon and Walmart), users browse, view, and purchase products and often create product reviews. With the rapid growth and pervasiveness of such online platforms, learning data-driven models of user behavior is indispensable to enable highly personalized user experiences.

Modeling user behavior in online interactional platforms poses exciting new challenges in handling behavioral data at the scale of over millions of entities (*e.g.*, users, content, groups, etc.). First, *data sparsity*, despite the enormous potential to learn nuanced behavioral patterns from massive interaction logs, a fundamental challenge is *data sparsity* for a vast majority of entities towards model learning. Second, *interaction heterogeneity*, users participate in diverse interactional scenarios, including structural *graph* connectivity in social networks, *bipartite* user-item interactions in e-commerce platforms, and *multipartite* user-group-item interactions in group activities. Sparsity challenges manifest in a variety of different ways, *e.g.*, a substantial fraction of users in social networks are inactive with limited *structural* and *engagement* information [1]; further, ground-truth *labels* are typically very limited and expensive to obtain in large-scale platforms. Similarly, we encounter *interaction* sparsity for a large proportion of the item inventory in rapidly expanding e-commerce platforms [2]. Modeling user behavior in the face of *sparsity* challenges across different interactional scenarios, is critical to personalize experiences for diverse user populations.

Recently, neural representation learning, powered by deep neural network architectures, has emerged as one of the most promising approaches for user modeling, with state-of-the-art performance in several established benchmarks and widespread adoption in various industrial settings, *e.g.*, video recommendations on YouTube [3], personalized search ranking and listing recommendations at AirBnb [4], related pin recommendations on Pinterest [5, 6] etc. Neural networks have the representational capacity to learn sophisticated behavioral patterns from extensive volumes of data without the need for manual feature engineering. However, developing sparsity-aware neural user modeling frameworks to enable personalized

inference at scale is important, challenging, and has remained largely beyond reach.

In this thesis, we develop neural representation learning frameworks to characterize user behavior in online interactional platforms, broadly divided into *transductive* and *inductive* learning scenarios. Transductive learning involves behavior modeling and prediction for entities seen during model training, while inductive learning concerns unseen entities that are only observed during inference. We design neural user modeling frameworks with two desirable properties, *sparsity-awareness*: we leverage different facets of information reflecting user behavior (*e.g.*, interconnectivity in social networks, temporal and attributed interaction information) to enable personalized behavior inference at scale in the face of interaction sparsity; *generalizability*: we present architecture-agnostic frameworks with generalizable learning strategies; our models are complementary to concurrent advances in neural architectural choices and are adaptive to the rapid addition of new applications in online platforms.

1.2 TECHNICAL CHALLENGES

Designing neural user modeling frameworks poses several technical challenges in handling the massive scale of behavioral data and the diversity of interaction types involving over millions of entities. Despite the potential to learn neural models that harness massive interaction logs, a central theme of this dissertation is addressing *data sparsity* challenges that manifest in different ways across applications, which are briefly summarized below:

Ground-truth Label Sparsity: In large-scale online platforms, we typically have access to very limited ground-truth labels for entity attributes, *e.g.*, age, gender of users in social networks or aspect ratings of products in e-commerce platforms.

Entity-level Interaction Sparsity: Heavy-tailed distributions in user interests and interaction patterns are commonly observed in user behavioral data [7] across several online platforms. Despite the availability of behavioral data at massive scales, we have very limited historical records at the granularity of individual entities due to the highly skewed interaction distribution. Thus, learning meaningful trends or insights for a vast majority of entities is challenging due to *entity-level interaction sparsity*.

Feature Diversity and Skew: In inductive learning applications, entities are often represented using a combination of diverse features, *e.g.*, user features in social networking platforms may include static profile attributes, dynamic communication and engagement activities, while item features in e-commerce platforms may include textual descriptions, product reviews, and other attributes (such as brand or price). Here, the

key modeling challenges are feature heterogeneity and skew: the features often belong to different modalities and exhibit non-trivial correlations; we also observe skewed occurrences of features across different entities.

In this dissertation, we primarily develop architecture-agnostic frameworks for user behavior modeling, with generalizable learning strategies that are targeted towards addressing the aforementioned sparsity challenges. Our learning frameworks enable personalized inference at scale for broad application scenarios across diverse platforms. Furthermore, we also present an example of a specific deep learning architecture design that effectively handles sparsity concerns in a large-scale industrial application.

Our problem settings span transductive and inductive learning scenarios, where transductive learning models behavior of entities seen during training and inductive learning targets unseen entities that are only observed during inference. Thus, our key research contributions are also organized below into *transductive* and *inductive* user behavior modeling applications.

1.3 TRANSDUCTIVE USER BEHAVIOR MODELING

In this part, we explore two fundamental *transductive* interactional scenarios with access to interactions/connectivity information of all participating entities during model training.

First, we consider a *graph-structured* data representation to model interactions between various entities, such as how users connect and communicate with each other in social networks, or how they purchase and rate products in e-commerce platforms. We examine transductive or semi-supervised learning given structural graph connectivity between the interacting entities and a few manually labeled examples. This is a natural way to formulate user profiling in online platforms given *sparse* ground-truth labels for a certain attribute (*e.g.*, age, gender), and the goal is to exploit the graph-structured interactions to infer the attribute for the remaining users. In Chapter 3, we leverage *higher-order network structures* to develop a semi-supervised learning framework (INFOMOTIF) over graphs.

We then investigate the *bipartite* user-item interactional setting, which is the foundation for a wide variety of Collaborative Filtering (CF) methods [8] that personalize item recommendations based on historical user interactions. A close examination of prior neural recommendation models reveals poor accuracy levels for the vast majority of items (with sparse interactions) in the item inventory [2, 9]. In Chapter 4, we introduce a *few-shot learning* framework (PROTOCF) that extracts and transfers meta-knowledge from data-rich head entities (such as popular items), to enable accurate personalized recommendations for data-poor tail entities (such as long-tail items) with severe interaction sparsity.

1.3.1 Higher-Order Structures in Graph-based Interactions

Graph Neural Networks (GNNs) have emerged as a popular paradigm for semi-supervised learning on graphs given a few labeled examples, and have recently enabled substantial advances due to their ability to learn node representations combining graph structure and node/link attributes. Localized *message passing* between neighboring nodes is the basis of node representation learning in graphs, including the popular GNN models. We identify two key imitations due to localized message passing: *over-smoothing* (poor resolution in distinguishing structural node neighborhoods) and *localization* (k -layer message-passing is restricted to the k -hop neighborhood of the labeled training nodes), which degrade performance of prior GNNs due to limited availability of labeled information.

To address the limitations of localized message passing, we propose the novel concept of *attributed structural roles*, grounded on *network motifs*, to regularize arbitrary GNN models for semi-supervised learning. Network motifs are a general class of *higher-order structures* (such as dense subgraphs and cliques) that indicate connectivity patterns between nodes, are crucial for understanding the organization and properties of complex networks [10]. Our key hypothesis is that leveraging the higher-order connectivity structures between nodes is critical to accurately label the nodes, effectively compensating the lack of sufficient labeled information in the local neighborhoods. Attributed structural roles identify structurally similar nodes with co-varying local attributes, independent of network proximity.

In this work, we propose an *architecture-agnostic* framework (INFOMOTIF) to capture attributed structural roles by dynamically prioritizing multiple motifs in the learning process without relying on distributional assumptions in the underlying graph or the learning task. Our experiments across a wide variety of assortative and disassortative networks, indicate significant gains (3-10% accuracy) for INFOMOTIF over prior approaches, with stronger gains for nodes with sparse training labels and diverse attributes in local neighborhood structures.

1.3.2 Few-shot Collaborative Filtering via Meta-Transfer

Neural Collaborative Filtering (NCF) methods have recently revolutionized modern recommender systems with impressive gains over conventional collaborative filtering approaches. However, a closer analysis of prior neural recommendation models reveals skewed performance gains towards popular items with abundant historical interactions, with poor accuracy levels for a significant chunk of *long-tail* (niche) items in the inventory [2, 9]. Moreover, we empirically show that prior neural recommenders lack the *resolution power* to accurately rank relevant items within the long-tail owing to severe interaction sparsity. This restricts

personalization and impedes suppliers of long-tail items in under-represented categories (genres or styles). Targeting long-tail items can enhance recommendation diversity and bring relatively larger revenues compared to popular items with competitive markets.

In this work, we focus on learning to accurately recommend few-shot items with interaction sparsity. We formulate few-shot item recommendation as a metric-based meta-learning problem of *learning-to-embed* items with few interactions. Our key insight is *episodic training* to eliminate the interaction distribution inconsistency between head items (with abundant interactions) and tail items (with sparse interactions); we enable few-shot generalization by sub-sampling interactions from head items to mimic tail items during model training.

We propose a meta-learning framework (PROTOCF) that *extracts, relates, and transfers* the meta-knowledge learned by a neural base recommender over head items to learn robust and discriminative latent representations for tail items. Our experimental results indicate significant gains (Recall@50 gains of 4-20%) for PROTOCF on few-shot personalized item recommendations (with over 70% of the item inventory in the tail) over state-of-art models.

1.4 INDUCTIVE USER BEHAVIOR MODELING

In this part, we consider a few diverse *inductive* interactional scenarios where modeling user behavior entails new entities that are only observed during model inference.

First, we examine *content re-sharing* behavior of users in social micro-blogging platforms (such as Twitter and Tumblr) where user-generated content is often noisy and transient with a short lifespan. The spread or diffusion of user-generated content in social platforms hinges primarily on the contextual influence of recent re-shares and the extent of social connectivity. In Chapter 5, we present a social regularization framework (INFVAE) to carefully model the co-variance of temporal context (recent posting activities) with structural graph connectivity information, for predicting spread of user-generated content in social networks.

Next, we study *multipartite* user interactions (users, groups, items) in *ephemeral group* activities where groups have dynamic memberships and *sparse* item interactions. Group interactions are becoming increasingly popular in a variety of contemporary social platforms such as Meetup, Facebook and Snapchat groups, etc. In Chapter 6, we introduce a self-supervised learning framework (GROUPIM) to generate personalized item recommendations tailored to ephemeral groups with no previous interactions together.

Finally, we explore *multi-faceted* user interactions in an industrial social networking setting with rich multi-modal individual (such as interests, demographics, etc.) and pairwise attributes (such as communications). This is common in modern online platforms that routinely track many aspects of user behavior across different functionalities to gain multi-

faceted insights. In Chapter 7, we present our inductive learning approach (GRAFRANK) that incorporates *multi-modal* features to successfully overcome sparsity challenges for *less-active* and *low-degree* users, in the industrially ubiquitous application of friend suggestion in social platforms. We briefly summarize our key technical contributions below.

1.4.1 Social Regularization to Predict Information Diffusion

Modeling the spread or *diffusion* of information through content posting or re-sharing on social media platforms (such as Twitter, Facebook, etc.) has widespread applications, including news feed ranking, limiting misinformation spread, viral marketing, etc. To predict the set of influenced users who re-share a piece of user-generated content, prior work mainly consider the impact of *temporal* user-user influence learned from historical posting activities; this results in poor predictions for the vast population of passive users who seldom post content. Incorporating knowledge of *social graph connectivity* structures is crucial to overcome sparsity challenges for users with sparse diffusion interactions.

Social regularization builds on correlation theories such as *homophily* and *influence*. Social homophily [11] suggests stronger ties between users with shared latent interests, inducing similar behaviors without direct causal influence; in contrast, temporal influence captures direct peer-to-peer effects, resulting in temporally correlated diffusion behaviors [12]. Homophily and Influence are fundamentally *confounded* in observational studies of diffusion processes [13], which makes it challenging to contextually model the impact of both factors.

In this work, we propose a *social regularization* framework (INFVAE) to model the co-variance of social homophily (indicated by graph connectivity) and temporal influence (recent posting activities), for predicting diffusion behaviors. We differentiate diffusion *roles* of users (influential versus susceptible) and adopt *variational* priors to regularize the latent space of user representations via graph autoencoders designed to preserve structural connectivity information. Our experiments show significant gains (22% MAP@10) for INFVAE over state-of-the-art models, with massive gains for users with sparse diffusion interactions.

1.4.2 Self-supervised Ephemeral Group Interaction Modeling

With the emergence of social networking platforms such as Meetup and Facebook Event, social group activities have gained popularity; it is essential to provide groups with relevant item recommendations (*e.g.*, restaurant or a concert). Existing studies target *persistent* groups which are fixed, stable groups (*e.g.*, families watching movies) where the members have interacted with numerous items together as a group. We examine interactions

of *ephemeral* groups casually formed by ad-hoc users with *limited or no* historical interactions together. Ephemeral group interactions are pervasive in several real-world scenarios, including dining with strangers, traveling in group tours, and attending social events.

To address group interaction *sparsity*, our key technical insight is to exploit the preference covariance among group members and contextually prioritize their individual preferences, to make group recommendations. In contrast to prior work that design customized group preference aggregators, we achieve *inductive generalization* to ephemeral groups by designing *self-supervised* learning objectives to *regularize* base neural group recommenders with arbitrary individual preference encoders and group preference aggregators.

In this work, we present a *self-supervised* learning framework (GROUPIM) that relies on *mutual information* estimation and maximization over group memberships, to overcome group interaction sparsity for ephemeral group recommendation. Empirically, GROUPIM achieves significant performance gains (31-62% NDCG@20) over state-of-the-art group recommendation models, and is particularly effective for *large* and *diverse* groups.

1.4.3 Multi-Faceted Friend Ranking in Social Platforms

Graph Representation Learning methods (including GNNs), have recently advanced graph learning in multiple prominent academic applications, such as link prediction, community discovery, and node classification. Most prior transductive learning approaches directly learn latent representations per node, resulting in *prohibitive model sizes* for large-scale social networks with over multiple millions of users. It is vital to develop inductive learning techniques for *large-scale social modeling applications* where users interact with different functionalities, communicate with diverse groups, and have multifaceted interaction patterns.

In this work, we consider the industrially ubiquitous application of *friend suggestions* (recommending new candidate users to befriend) in social platforms. Compared to academic datasets, sparsity challenges are also typically exacerbated in industrial settings owing to the significant fraction of inactive users with limited structural and engagement information. To overcome structural and interactional sparsity, we exploit the rich knowledge of in-platform user actions to formulate friend suggestion as *multi-faceted friend ranking* on an evolving friendship graph, with multi-modal user features and link communication features.

We design a graph neural network architecture (GRAFRANK) that captures knowledge from multiple correlated feature modalities and user-user interactions to learn multi-faceted user representations for friend ranking. GRAFRANK outperforms state-of-the-art baselines on friend candidate retrieval (by 30% MRR) and ranking (by 20% MRR) tasks. Notably, GRAFRANK achieves higher gains for the critical population of *less-active* and *low-degree*

Part	Chapter	Chapter Title
Transductive User Modeling (Part I)	Chapter 3	Higher-Order Structures in Graph-based Interactions
	Chapter 4	Few-shot Collaborative Filtering via Meta-Transfer
Inductive User Modeling (Part II)	Chapter 5	Social Regularization to Predict Information Diffusion
	Chapter 6	Self-supervised Ephemeral Group Interaction Modeling
	Chapter 7	Multi-Faceted Friend Ranking in Social Platforms

Table 1.1: *Thesis Organization into transductive and inductive user modeling.*

users, and is currently being implemented by Snapchat to enhance their quick-add feature.

1.5 THESIS ORGANIZATION

This thesis is organized in two parts: *transductive* and *inductive* user behavior modeling.

In the first part, we model user behavior in *transductive* interactional scenarios where we have access to historical interactions and connectivities among all participating entities during model training, and the goal is to infer their missing properties (*e.g.*, profile attributes) or predict future behaviors (*e.g.*, item purchases) during model inference.

In the second part, we examine *inductive* interactional scenarios with real-time behavior prediction for new entities that are only observed during model inference. This scenario is common when dealing with user-generated content and new users in social media platforms and ephemeral group interactions with dynamic memberships (whose members may be together for the first time). Our primary focus here is the design of efficient sparsity-aware models that inductively generalize to new transient entities with limited interaction data. Table 1.1 provides a brief overview of different chapters in this thesis.

Table 1.2 summarizes the interaction types of user interactions in each chapter and their associated technical challenges. In chapter 3, we examine semi-supervised learning over graphs with sparse ground-truth training labels. We handle entity-level interaction sparsity challenges in Chapter 4 (long-tailed items), Chapter 5 (long-tailed users), and Chapter 6 (ephemeral groups). In chapter 7, we further incorporate multi-faceted features and interactions across different modalities, with skewed distributions across different users.

In this thesis, our user modeling applications operate on a variety of interaction types, including bipartite user-item interactions (Chapter 4), multipartite user-group-item interactions (Chapter 6), and graph-structured interactions (Chapters 3, 5, and 7).

In particular, our graph-based user modeling frameworks, although primarily targeted at social networks, generalize to a broad class of graphs that satisfy two key structural

properties: scale-free graphs and homophily. These assumptions directly stem from the inherent nature of human interactions; we typically expect skewed distributions to be present in interaction data produced by human activity, resulting in graphs that exhibit a power law distribution (scale-free); in most user-user interactions networks, similar users are expected to form social connections (homophily).

However, we note that general graphs that arise in non-social contexts, (*e.g.*, transportation networks or brain networks) may exhibit different structural properties. In such a scenario, it is critical to revisit the inductive biases in designing graph learning models (*e.g.*, graph embedding methods and graph neural networks).

Part	Chapter	Data Type	Technical Challenge
Transductive User Modeling (Part I)	Chapter 3	Graph Connectivity	Ground-truth Label Sparsity
	Chapter 4	Bipartite Interactions	Entity Interaction Sparsity
Inductive User Modeling (Part II)	Chapter 5	Graph Connectivity	Entity Interaction Sparsity
	Chapter 6	Multipartite Interactions	Entity set Interaction Sparsity
	Chapter 7	Graph Connectivity	Entity Interaction Sparsity and Feature Diversity

Table 1.2: *Comparison of various technical challenges and data characteristics across the different thesis chapters.*

CHAPTER 2: LITERATURE REVIEW

In this chapter, we provide a broad overview of prior work in user inference tasks and recommendation methodologies and applications in different interaction contexts.

First, we discuss related work on user modeling approaches centered around addressing sparsity challenges in social network analysis and mining applications. Then, we discuss prior sparsity-aware approaches research directions in a variety of different recommender systems applications. Finally, we discuss related work on various learning paradigms, including semi-supervised learning, deep generative modeling, self-supervised learning, few-shot learning, and meta-learning, that tackle sparsity challenges in diverse machine learning domains.

2.1 SOCIAL NETWORK MINING

There has been a lot of interest in the past on analyzing, characterizing, and predicting user behavior in online social networking platforms. In this dissertation, we primarily focus on machine learning techniques for modeling (understanding and predicting) user behavior in social networks, which are based on recent advances in representation learning over graphs.

We first briefly review prior work on unsupervised and semi-supervised graph representation learning techniques, which are targeted at node-level property inference and link prediction (or completion) applications in static graphs. Then, we review related work on modeling different kinds of temporally evolving user behaviors in social networks, including friendship creation, inter-user interactions, and content-sharing activities.

2.1.1 Static Graph Representation Learning

Graph embedding methods learn latent node representations in graphs to capture the structural properties of a node and its local neighborhoods. Broadly, they fall into two categories: the first is unsupervised graph embedding models that learn universal node representations that capture different structural graph properties (*e.g.*, community, roles, etc.); the second is the class of semi-supervised learning models that further exploit supervision (*e.g.*, ground-truth labels for nodes or links) to learn task-driven graph embeddings.

Unsupervised Graph Representation Learning: There are a number of techniques that have been proposed to learn unsupervised latent node embeddings. The earliest techniques such as Principal component analysis [14] and Laplacian Eigenmaps [15] were based on dimensionality reduction, optimize an objective over an affinity matrix of the entire graph

to maximize the variance of the latent representations. Inspired by advances in neural language models such as word2vec [16], stochastic graph embedding techniques such as Deepwalk [17] and node2vec [18] learn unsupervised embeddings to maximize the likelihood of co-occurrence of nodes in fixed-length random walks. A related approach is the design of learning objectives to learn node embeddings that capture different orders of proximity between nodes in a graph [19, 20, 21].

As an alternative to proximity-preserving objectives to learn graph embeddings, some methods learn role-aware embeddings that embed structurally similar nodes close in the latent space, independent of network position [22, 23]; such methods typically utilize structural node features (*e.g.*, node degrees, motif count statistics) to extend classical proximity-preserving embedding methods based on random walks [24] and matrix factorization [25].

Neural graph embedding techniques were first introduced in SDNE [26] which utilized graph autoencoders to learn node embeddings that preserve both local and graph measures of graph proximity. More generally, the class of graph autoencoders [26, 27, 28] employ various encoding and decoding architectures to embed graph structure and learn node embeddings through a variety of unsupervised training objectives. A few methods extended the above techniques to also incorporate knowledge of node features (or attributes) in the embedding learning process [29, 30, 31].

Semi-supervised Graph Representation Learning: Graph-based semi-supervised learning is a well-studied problem, where the goal is to classify nodes in a graph given a small set of labeled examples [32]. Traditional methods propagate labels through linked nodes in the graph based on different smoothness assumptions [33]. Collective classification techniques [34] generalize label propagation by further utilizing node features and allowing more flexible local updates. Semi-supervised variants of graph embedding techniques include label information in the learning process through supervised training objectives [35, 36].

Graph Neural Networks: Graph Neural Networks are neural networks over graphs designed to learn node-level or graph-level representations towards a variety of learning tasks. Initial GNNs generalized Convolutional Neural Networks (CNNs) to graphs in the spectral domains; spectral GNNs [37, 38] defined graph convolutions through the Graph Fourier Transform described by eigenvectors of the Graph Laplacian. However, spectral methods (being a function of the graph Laplacian) cannot generalize to graphs with different structural properties.

Message-passing GNNs (spatial) generalize label propagation through localized message passing over node neighborhoods; they learn node representations by recursively propagating features (*i.e.*, message passing) from local neighborhoods through the use of aggregation and activation functions [39]. Graph Convolutional Networks (GCNs) [39] learn degree-weighted neighborhood aggregators, which can be interpreted as a form of Laplacian smoothing. Many

models generalize GCN with a variety of learnable neighborhood aggregators, *e.g.*, self-attentions [40], mean and max pooling functions [41]; these approaches have consistently outperformed proximity-preserving graph embedding techniques on several benchmarks.

In this dissertation, we identify two key limitations in prior message-passing GNN architectures: k-hop localization and over-smoothing. A few recent methods have attempted to address either of these issues. To handle localization, non-local GNN variants incorporate information from nodes in different localities via varying influence radii [42], shortest paths [43], and global summaries [44]. To address over-smoothing, a few methods design structural GCNs via degree-aware [45] and motif-based aggregators [46, 47]; however, such methods are highly localized. In chapter 3, we design a learning framework over GNNs to learn attributed structural roles based on the co-variance of features in motifs (higher-order connectivity structures), thus simultaneously enhancing the distinguishability of node representations and identifying structurally similar nodes independent of graph proximity.

Heterogeneous Graphs: Recent work generalize graph embedding techniques and graph neural networks to heterogeneous graphs containing nodes and edges of different types. The key idea is to incorporate structural and semantic information indicated by meta-path or meta-graph structures. A few popular heterogeneous graph embedding approaches capture proximity between nodes connected via meta-paths [48], meta-graphs [49, 50], and meta-path guided random walks [51, 52]. Heterogeneous graph neural networks conduct message-passing aggregation over local neighborhoods induced by specific node types [53, 54], meta-paths [55, 56, 57] and meta-graphs [58]. Despite advances in modeling rich heterogeneous semantics into message-passing, the key limitation of localization remains. In our work in chapter 3, we also demonstrate the benefits of structure role learning (via typed motif structures) in our framework for heterogeneous graph mining applications.

In addition to reviewing prior graph representation learning models designed for node property inference or link prediction applications in static graphs, we also discuss prior work related to modeling temporally evolving user behaviors in social networks.

2.1.2 Temporal User Behavior Modeling

In modern social platforms (such as Facebook, Twitter, and Snapchat), users participate in a wide variety of temporally evolving interactions, *e.g.*, form friendships and following connections, post or re-share user-generated content, communicate with other users, and interact with a variety of different functionalities within the platform.

In this dissertation, we examine two central applications: modeling temporal friendship formation (or broadly, temporal link evolution) and the spread (or diffusion) of user-

generated content in social networks. We now review prior work on these research directions.

Temporal Link Evolution: The earliest methods designed for modeling friendship formation in social networks were motivated by the principles of homophily [11] and triadic closure [59]. Initial methods include carefully designed heuristics to model user-proximity, *e.g.*, path-based Katz centrality [60] or common neighbor-based Adamic/Adar [61] and learning techniques that exploit such pairwise features for link ranking [62]. However, heuristic feature extraction for each potential link is infeasible in large-scale time-evolving social networks.

Recent methods introduced graph representation learning models over dynamic graphs, which are commonly represented as a sequence of evolving graph snapshots at multiple discrete time steps. Such techniques broadly fall into two categories: the first category includes temporal smoothness methods that ensure node embedding stability across consecutive time-steps, guided by triadic closure [63, 64] and incremental models that update embeddings from the previous time step [65]. However, these methods cannot capture long-range variations in graph structure, and are inadequate when nodes exhibit vastly differing evolutionary behaviors. The second category includes dynamic graph neural networks that model temporal graph evolution; popular approaches include recurrent methods [66, 67] that capture temporal dynamics via hidden states to summarize historical snapshots, and self-attentional models [68] that flexibly weight historical representations.

Despite the recent advances in dynamic GNN architectures, they remain unexplored for large-scale user-user social modeling applications (such as friend suggestion) where users exhibit multifaceted behaviors. In chapter 7, we design dynamic GNNs for the important application of friend suggestion, through a novel multi-faceted friend ranking formulation with multi-modal user features and link communication features.

Information Diffusion Modeling: Prior diffusion models in social networks were designed towards one or more objectives: *microscopic* diffusion prediction, which learns a model from social graph connectivity and diffusion cascade sequences, to predict the next (or final set of) influenced user(s) given a sequence of initial activated seed users; and *macroscopic* cascade prediction, which predicts global properties of information spreading in the social network, including size, growth, and shape of cascades over time [69]. In this dissertation, our main focus is modeling individual user behavior; thus, we focus on microscopic diffusion modeling.

Historically, modeling spread of user-generated content in social networks has been studied through two seminal diffusion models: Independent Cascade (IC) [70] and Linear Threshold (LT) [71]; IC models information spreading as cascades of activations over the network, while LT determines user activations according to thresholds on incoming neighbor influence. The earliest data-driven diffusion models extend IC and LT via probabilistic generative models to incorporate knowledge of topics [72], continuous timestamps [73], user profiles [74], and

community structures [75, 76]. A key limitation of such methods is the reliance on extensive feature engineering and the limited modeling capacity of the chosen probability distributions.

Recently, representation learning models have incorporated user influencing capability and susceptibility in diffusion to learn latent user embeddings from ordered pairwise co-occurrences in diffusion cascades and social graph connectivity structures [77, 78, 79, 80]. To handle time-stamped diffusion cascade sequences, neural diffusion models have generalized sequential models (such as RNNs and LSTMs) to consider DAG propagation structures by projecting cascades onto local social neighborhoods [81, 82, 83]. As such, prior diffusion models capture the temporal correlation in diffusion behaviors based on activations observed in the cascades, which may be inadequate for the vast majority of users that seldom post content and appear in very few cascades (sparse diffusion interactions).

In our work in chapter 5, we address the key challenge of diffusion interaction sparsity by utilizing variational priors to regularize latent user representations in diffusion models via variational graph autoencoders designed to preserve structural connectivity information. By parameterizing latent space priors through flexible deep generative models, our framework also generalizes classical probabilistic generative modeling approaches.

2.2 RECOMMENDER SYSTEMS

Recommender systems appear in a variety of e-commerce, advertising, and social networking applications. This has led to several recent advances in developing neural recommendation models designed to handle the massive scale and diversity in user interactions. Our key focus is motivated by the fundamental data sparsity challenge for the vast majority of entities towards model learning (owing to heavy-tailed interaction distributions).

In this dissertation, we examine few-shot recommendation settings targeted at entities with very few interactions. Specifically, we explore sparsity challenges in personalized item recommendations to users (collaborative filtering), and groups of users (group recommendation). Below, we review prior work on collaborative filtering and group recommendation.

2.2.1 Collaborative Filtering

Collaborative Filtering (CF) is one of the most popular techniques for user modeling in recommender systems. First, we briefly discuss classical CF models, followed by recent advances in neural CF models and techniques that address interaction sparsity challenges.

Classical Collaborative Filtering: Classical CF models operated on the explicit feedback setting by formulating a matrix completion problem to infer the unknown user-item ratings;

the key idea towards rating inference is to derive knowledge from those of “similar” items or “similar” users. There are two broad classes of techniques: neighborhood based collaborative filtering [84] and latent-factor matrix factorization [85] models. Neighborhood based CF models determine similar items (or users) based on their rating similarity or co-occurrence, and estimate unknown ratings by a weighted average of ratings over the similar items. Matrix factorization models decompose the user-item rating matrix into user and item specific latent factors, which are later used to infer missing ratings via inner products between the user and item latent factors. The earliest methods designed for implicit feedback data (commonly observed in massive online interactions), extended matrix factorization models with different pointwise and pairwise ranking objectives that were designed to handle missing data using sample re-weighting [86] or negative sampling [87].

Neural Collaborative Filtering: Neural CF models generalize matrix factorization to parameterize users and items with vectorized representations and interaction functions that are powered by deep neural networks. To enhance the embedding layer that generates user and item representations, much effort has been devoted to examine several modeling choices including latent embeddings [88], denoising [89] and variational autoencoders [90] and graph neural networks [5, 91]. Other recent efforts to improve the interaction function exploit deep learning techniques such as nonlinear neural networks [88], memory networks [92], factorization machines [93], and euclidean distance metrics [94], to capture non-linear feature interactions between users and items. While these neural models learn expressive models to significantly outperform conventional CF approaches, sparsity concerns owing to long-tail items and cold-start users remain a critical challenge.

Few-shot and Cold-start Recommendation: Few-shot and cold-start recommendation are two important and related sub-problems that have received attention in prior research. Most neural recommendation models do not learn meaningful latent item (or user) representations in the cold-start and few-shot settings, where new items (or users) provide either no interactions or a handful of interactions, respectively. We briefly review prior work in both the few-shot (commonly known as long-tail) and cold-start recommendation scenarios.

Long-tail challenges in recommender systems arise due to interaction sparsity for the vast majority of items (and users). Clustering is one popular way to address interaction sparsity by exploiting modeling group-level information to account for the lack of entity-level interactions; early methods generate recommendations over tail items at the granularity of item/user clusters, *e.g.*, cluster-based smoothing [95], user-item co-clustering [96] and joint clustering and collaborative filtering [97]. Other efforts leverage item-item co-occurrence statistics to regularize recommenders, *e.g.*, joint factorization of an item co-occurrence matrix that shares latent factors with a CF model. In a few neural recommenders, variational auto-

encoders (VAEs) [90, 98] employ distributional regularization via Gaussian priors on the latent space, and have effectively alleviated sparsity challenges. Another recent strategy to alleviate sparsity is data augmentation for items (or users) in the tail via adversarial regularization [9] or rating generation [99] techniques. However, adversarial learning is often computationally expensive and does not scale to massive online platforms. While the few-shot learning problem also appears in other prediction problems with few-shot classes [100, 101] (given a handful of samples), the massive scale of interaction data (note that each user/item is a few-shot instance, unlike few-shot classes) renders these solutions inapplicable.

In chapter 4, we formulate long-tail item recommendations in the most general user-item interaction setting, as a few-shot learning problem of learning-to-recommend few-shot items with very few interactions [102]. Our proposed meta-learning framework is complementary to advances in neural recommender architectures and enables flexible adaptation to the tail.

Cold-start recommendation is a related problem that specifically targets new users or new items with *no* historical interactions. In such a scenario, the key idea is to rely on auxiliary side information, *e.g.*, item content [103], textual reviews [104], contextual factors [105], social connections [106], and external knowledge graphs [107]. A wide variety of techniques have been proposed for each of the sub-problems, *i.e.*, content-based collaborative filtering, review-based recommendation, context-aware recommendation, social recommendation, and knowledge-aware recommendation respectively. Generalizable techniques include randomized feature dropouts to enable generalization to missing inputs [108, 109], and meta-learning frameworks for zero-shot (cold-start) recommendation. Meta-learning methods for cold-start recommendation utilize gradient-based parameter adaptation [110, 111, 112, 113, 114], hyper-parameter initialization [115], and shared layers with user-specific (or item) parameter adaptation [116, 117, 118]. In this dissertation, we focus on the few-shot setting without requiring access to any side information to ensure generalizability; a deeper exploration of different cold-start recommendation scenarios is an important future direction.

2.2.2 Group Recommendation

Group interactions have become increasingly prevalent in a variety of contemporary social networking platforms such as Meetup, Facebook, and Snapchat. Prior literature on developing recommender systems to suggest items relevant to groups of users, can be broadly divided into two categories based on group types: persistent and ephemeral. Persistent groups have stable members with rich activity history together, while ephemeral groups comprise users who interact with very few items together and may appear only during inference time.

Persistent Group Recommendation: Since persistent groups have stable groups with rich

historical interactions together, previous studies treat groups as virtual users, and adopt conventional personalized recommendation techniques to make group recommendations. A few key approaches include genetic algorithm-based group interaction modeling [119], power balance between group members towards content selection [120], and deep neural architectures modeling individual choices and group decisions [121]. applied. However, such methods cannot handle new groups that are only observed during inference time.

Ephemeral Group Recommendation: Prior efforts focus on investigating different strategies for aggregating individual preferences and modeling group interactions. To generate ephemeral group recommendations, they either aggregate recommendation results (or item scores) for each member, or aggregate individual member preferences, thus falling into two classes: score (or late) aggregation [122] and preference (or early) aggregation [123].

The score (or late) aggregation strategies first generate recommendation results for each group member, and then generate group recommendations by aggregating these individual results based on static predefined aggregation strategies. Popular *score aggregation* strategies include least misery [122], average [124], maximum satisfaction [125], and relevance and disagreement [126], inspired by social choice theories [127]. An empirical comparison of different heuristic strategies [122] has demonstrated the absence of a clear winner, especially with the variance in group sizes and coherence levels. However, the hand-crafted heuristics are static modeling hypotheses that overlook real-world group interactions.

The preference (or early) aggregation strategies first aggregate the preferences of group members, which are then utilized to produce group recommendations. The first approaches in this category fuse the profiles (raw item histories) of members into a group profile and utilize conventional recommenders to generate group recommendations [128]. Recent approaches introduced latent-factor models based on probabilistic generative models and deep neural networks. Probabilistic methods [123, 129] model the group generative process by considering both the personal preferences and relative influence of group members, to differentiate their contributions towards group decisions. Neural methods explore attention mechanisms [130] to learn data-driven preference aggregators [131, 132, 133] jointly from individual and group interactions; yet, these models severely suffer from the interaction sparsity challenges for ephemeral groups, which often results in degenerate solutions.

In chapter 6, we address interaction sparsity for ephemeral group recommendation by designing self-supervised learning strategies based on mutual information estimation and maximization, to regularize arbitrary neural base group recommenders. In the following section, we review machine learning paradigms that are relevant to the proposed user behavior modeling approaches in this dissertation.

2.3 RELEVANT MACHINE LEARNING PARADIGMS

In this dissertation, we address different sparsity challenges that manifest in online user interactions. Our proposed frameworks and models have deeper connections to broader machine learning paradigms and their applications in other domains, including computer vision and natural language. We briefly review relevant work in semi-supervised learning, deep generative modeling, self-supervised learning, and few-shot and meta-learning paradigms.

2.3.1 Semi-supervised Learning

To account for lack of labeled data in supervised learning tasks, semi-supervised learning leverages large amounts of unlabeled data to learn additional structure about the input distribution, with the goal of enhancing model performance on the learning task. Different modeling assumptions have been examined in prior work, including label smoothness, low-density (or clustering), and manifold regularization [32, 134]. One popular approach is transductive learning that constructs a large graph connecting neighboring objects (similar data points are connected by an edge), and propagates labels from the initial labeled set to the unlabeled objects (following label smoothness assumptions). More recently, graph neural networks [39] generalize classical label propagation techniques through trainable neighbor aggregation over feature-rich local node neighborhoods in attributed graphs.

Graph-based semi-supervised learning has enormous potential for social interaction modeling tasks where users are already connected and typically satisfy homophily assumptions along several attributes. In this dissertation, we have designed novel graph neural network frameworks for semi-supervised learning and ranking over graphs in chapters 3 and 7.

2.3.2 Deep Generative Modeling

Deep Generative Models are a powerful class of neural networks that aim to approximate any kind of data distribution. Variational Autoencoders (VAEs) [135] and Generative Adversarial Networks (GANs) [136] are the most popular frameworks for deep generative modeling, and have achieved tremendous success in several computer vision and natural language processing applications, including image and video generation [137, 138], editing [139], and enhancement [140], and text synthesis [141] and summarization [142]. Deep generative models have recently been examined in graph mining and recommender systems applications to address sparsity challenges via data augmentation [99], enhanced hard negative mining [143], and distributional priors on the latent space [90].

Flexible distribution-aware and data-driven prior regularization enables better characterization of long-tail and cold-start entities. In chapter 5, we explored deep generative models for robust behavior modeling in social networks with sparse user interactions.

2.3.3 Self-supervised Learning

To further alleviate sparsity challenges, the recently emerging paradigm of self-supervised learning defines auxiliary (pretext) tasks which are formulated using only unlabeled data; solving the pretext task learns effective latent representations that can benefit a variety of different learning tasks. The most effective framework is contrastive learning, which aims to maximize the agreement between different views of the data, by contrasting against negative samples. Contrastive learning techniques have benefited a broad range of computer vision tasks by formulating pretext tasks directly from local-global contrast [144] or by rotating, cropping, or colorizing images [145, 146]. Self-supervised learning has also been effectively utilized in recent natural language representation learning models by carefully designing pretext texts, *e.g.*, masked word prediction and next sentence prediction in BERT [147], sentence order prediction in ALBERT [148], sentence permutation in BART [149] etc.

Despite promising recent advances in self-supervised learning techniques across computer vision and natural language processing applications, this paradigm is relatively unexplored in the graph mining and recommendation domains. In chapter 3, we explore self-supervised learning strategies to train graph neural networks by exploiting higher-order structures (via network motifs) to formulate pretext tasks. In chapter 6, we presented a self-supervised learning framework that utilized group membership structure learning to overcome group interaction sparsity for ephemeral group recommendation.

2.3.4 Few-shot Learning and Meta-Learning

Few-shot learning is the paradigm of designing models capable of learning new tasks (*e.g.*, classification tasks) rapidly given a limited number of training examples; such techniques are useful when training examples are hard to find, or where the cost of data labeling is high [150]. Recently, meta-learning (learning to learn) framework has emerged as an effective approach for few-shot learning. In the meta-learning framework, we learn how to learn (*e.g.*, to classify) given a set of training tasks with the expectation of generalization to new tasks that have never been encountered during training time. The adaptation process is a mini learning session that occurs during inference time with limited exposure to the new task configurations. There are three common prior meta-learning modeling approaches:

(a) learning an efficient distance metric (metric-based) [101]; (b) deep neural network (*e.g.*, recurrent) with external or internal memory (model-based) [151]; (c) optimize the model parameters explicitly for fast learning (optimization-based) [152].

Designing meta-learning frameworks to handle interaction sparsity in recommendation settings poses unique scaling challenges. In chapter 4, we introduce metric-based meta-learning frameworks for few-shot item recommendations.

CHAPTER 3: INFOMOTIF: HIGHER-ORDER STRUCTURES IN GRAPH-BASED INTERACTIONS

3.1 INTRODUCTION

In this chapter, we propose a new class of motif-regularized graph neural networks (GNNs); GNNs have emerged as a popular paradigm for semi-supervised learning on graphs due to their ability to learn representations combining topology and attributes, without relying on expensive feature engineering. GNNs are typically formulated as a message passing framework [153], where the representation of a node is computed by a GNN layer aggregating features from its graph neighbors via learnable aggregators. Long-range dependencies are captured by using k layers to incorporate features from k -hop neighborhoods. GNNs have demonstrated promising results in several application domains spanning *homogeneous* graphs (*e.g.*, user-user friendship networks) comprising nodes and edges of a single type, and *heterogeneous* graphs (*e.g.*, academic citation networks) containing nodes and edges of different types. In particular, GNNs have achieved state-of-the-art results in several applications including node classification [39], link prediction [154], and personalized recommendation [5].

Localized message passing limitations: We illustrate two key limitations of prior k -layer GNN architectures: *k-hop localized* and *over-smoothed* representations (Figure 3.1).

1. GNNs, while highly expressive, are inherently *localized*: a k -layer GNN cannot utilize features of nodes that lie outside the k -hop neighborhood of the labeled training nodes. In Figure 3.1, nodes a and b belong to different classes. A 2-layer GNN sees unlabeled node c within the aggregation range of a (class 1) and outside the influence of b (class 2 and more than 2 hops away). Thus, a GNN will more likely label c with class 1 (than class 2). However, in reality, c and b display identical attributes (node color) in the local structure; a localized GNN fails to incorporate this factor.
2. GNNs with multiple layers learn *over-smoothed* node representations by iteratively aggregating neighbor features [155]. In Figure 3.1, nodes c and a share the same number of neighbors with blue and green attributes; however, green neighbors of node a form triangles, while blue neighbors of node b (and c) form triangles. Considering *local nodal attribute arrangements*, node c is more similar to b than to a . The over-smoothing effect in GNNs obscures this attribute co-variation difference when classifying node c .

Thus, we require a new learning framework over graphs, to overcome the above key limitations due to localized message passing in popular GNN models.

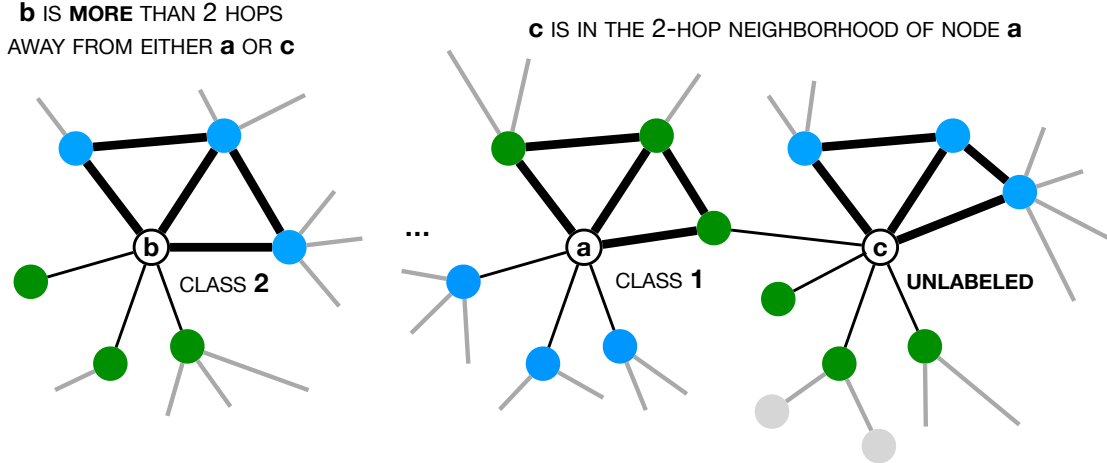


Figure 3.1: Localized message passing limitations: A stylized example in a homogeneous graph with a 2-layer GNN model (node colors indicate node attributes). Node a is in the 2-hop range of node c . Node c **does not receive** gradient updates from node b (class 2) since node b is more than 2 hops away. A 2-layer GNN will likely label node c as class 1. Notice that c is in class 2 since c and b have identical local structure and attribute co-variation.

One way to overcome these limitations is the paradigm of *role discovery* [156] that identifies nodes with structurally similar neighborhoods. In contrast to the notion of communities defined by network proximity, structural roles characterize nodes by their local connectivity and subgraph patterns independent of their location in the network [22]; thus, two nodes with similar roles may lie in different parts of the graph. Prior role-aware models learn similar representations for structurally similar nodes while ignoring nodal attributes [24], *i.e.*, they will assign the same role to nodes a and b in Figure 3.1 with topologically identical local structures; however, nodes a and b differ in their local attribute arrangements (blue vs. green attributes in triangles), and thus belong to different classes. Furthermore, structural role learning is relatively unexplored in heterogeneous graphs with typed nodes and edges.

Present Work: To enable the expressivity to distinguish attributed structures, we propose the concept of *attributed structural roles* that identify structurally similar nodes with co-varying attributes, independent of network proximity. We ground structural roles on *network motifs*¹, which are induced subgraph structures over a few nodes (*e.g.*, triangles). Network motifs are a broad class of *higher-order structures* that indicate connectivity patterns between nodes, and are crucial for understanding the organization and properties of complex networks [10]. In addition, network motifs can be easily generalized to capture type semantics in rich heterogeneous graphs through heterogeneous (typed) higher-order struc-

¹The terms network motif, graphlet, and induced subgraph are used interchangeably in literature

tures [157] (also known as metagraphs). Leveraging higher-order connectivity structures between nodes is extremely valuable to overcome the lack of sufficient training labels in local neighborhoods during semi-supervised learning. We define two nodes as sharing attributed structural roles if they participate in topologically similar motif instances over co-varying sets of attributes. We note that attribute co-variance permits for multiple discrete and continuous attributes, rather than stricter notions such as regular equivalence [22].

We propose INFOMOTIF, a GNN architecture-agnostic regularization framework that exploits the co-variance of attributes and motif structures. INFOMOTIF learns regularizers based on a set of network motifs, which vary in their task-specific significance. Specifically, across instances of the same motif (*e.g.*, a triangle structure), we learn discriminative attribute correlations to regularize the underlying GNN node representations; this encourages the GNN to learn statistical correspondences between distant nodes that participate in similarly attributed instances of that motif. We propose a novel training curriculum to integrate multiple motif regularizers while attending to motif types and skewed motif distributions. We summarize our key contributions below:

Attributed Structural Role Learning: We propose the novel concept of *attributed structural roles* to regularize GNN models for semi-supervised learning over graphs. In contrast to prior role discovery work that identify structurally similar nodes agnostic to attributes [24], we adopt the paradigm of *self-supervised learning* to regularize node representations to capture *attribute correlations* in motif structures. Our framework INFOMOTIF unifies the expressive local neighborhood aggregation power of message-passing GNNs with the paradigm of structural role discovery.

Architecture-agnostic Regularization Framework: To the best of our knowledge, INFOMOTIF is the first to address the limitations of localized message passing in GNNs through an architecture-agnostic framework. Unlike prior attempts that design new aggregators [42, 45], we achieve architecture independence by modulating the node representations learned by the base GNN through motif-based *mutual information maximization*, to capture attributed structural roles. We regularize three state-of-the-art GNNs within our framework to demonstrate significant performance gains.

Distribution-agnostic Multi-Motif Curriculum: We propose two learning progress indicators, *task-driven utility* and *distributional novelty*, to integrate multiple motif regularizers within our INFOMOTIF framework. Unlike prior strategies [33, 158] that incorporate regularizers via tunable hyper-parameters, our training curriculum dynamically prioritizes different motifs in the learning process without relying on any

distributional assumptions on the underlying graph or on the learning task.

We regularize three state-of-the-art GNN models in our INFOMOTIF framework for semi-supervised node classification. Our experiments are conducted on a wide variety of real-world datasets spanning homogeneous and heterogeneous networks. In *homogeneous* graphs, our experimental results indicate significant gains (3-10% accuracy) for INFOMOTIF over prior approaches on two diverse classes of datasets: assortative *citation* networks that exhibit strong homophily and dis-assortative *air-traffic* networks that depend on structural roles. We also demonstrate the utility of our framework in three *heterogeneous* graph datasets where INFOMOTIF outperforms a number of state-of-the-art methods with significant performance gains (5% accuracy) on average. Our qualitative analysis indicates stronger gains for nodes with *sparse training labels* and *diverse attributes* in local neighborhood structures.

We organize the rest of the chapter as follows. In Section 3.2, we present the problem formulation, and introduce preliminaries on graph neural networks and network motifs. We describe our proposed framework INFOMOTIF in Sections 3.3 and 3.4, present experimental results in Section 3.5, discuss limitations in Section 3.6, finally concluding in Section 3.8.

3.2 PRELIMINARIES

In this section, we formalize semi-supervised node classification on graphs via graph neural networks and introduce network motifs in homogeneous and heterogeneous graphs.

3.2.1 Problem Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an attributed graph, with nodes \mathcal{V} and edges $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$. Note, $\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_U$, the sets of labeled (\mathcal{V}_L) and unlabeled (\mathcal{V}_U) nodes in the graph. Let $\mathcal{N}(v)$ denote the neighbor set of node $v \in \mathcal{V}$ in \mathcal{G} , and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times F}$ denotes the attribute matrix with rows $\mathbf{x}_v \in \mathbb{R}^F$ for node $v \in \mathcal{V}$. In our work, the graph may be *heterogeneous* with multiple types of nodes and edges; in such a scenario, we have a node type mapping $\psi : \mathcal{V} \mapsto \mathcal{T}_V$ where \mathcal{T}_V is the set of T_V node types that identifies each node in \mathcal{V} with a type in \mathcal{T}_V , and a corresponding edge type mapping $\xi : \mathcal{E} \mapsto \mathcal{T}_E$ where \mathcal{T}_E is the set of T_E edge types. Each labeled node $v \in \mathcal{V}_L$ belongs to one of C classes, encoded by a one-hot vector $\mathbf{y}_v \in \mathbb{B}^C$ ($\mathbb{B} = \{0, 1\}$). Our goal is to predict the labels of the unlabeled nodes $v \in \mathcal{V}_U$. This is the familiar transductive or semi-supervised learning setup for node classification in large-scale graphs [33].

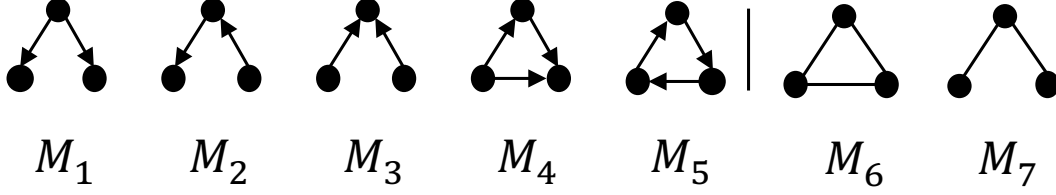


Figure 3.2: Topologically distinct, directed (M_1 to M_5) and undirected (M_6 to M_7) connected, network motifs over three nodes.

3.2.2 Graph Neural Networks

Graph Neural Networks (GNNs) use multiple *message-passing* layers to learn node representations. At each layer $l > 0$, where 0 is the input layer, GNNs compute a representation for node v by aggregating features from its local neighborhood, through a learnable aggregator function $f_{\theta,l}$ per layer. Using k layers allows for the k -hop neighborhood of a node to influence its representation. Let $\mathbf{h}_{v,l-1} \in \mathbb{R}^D$ denote the representation of node v in layer $l - 1$. The l -th layer of the GNN follows a message passing rule given by:

$$\mathbf{h}_{v,l} = f_{\theta,l}(\mathbf{h}_{v,l-1}, \{\mathbf{h}_{u,l-1}\}), \quad u \in \mathcal{N}_v \quad (3.1)$$

Equation (3.1) says that the node embedding $\mathbf{h}_{v,l} \in \mathbb{R}^D$ for node v at the l -th layer is a non-linear aggregation $f_{\theta,l}$ of the embeddings from layer $l - 1$ of node v and the embeddings of immediate network neighbors $u \in \mathcal{N}(v)$ of node v . The function $f_{\theta,l}$ defines the message passing mechanism at layer l and we can use a variety of aggregator architectures, including graph convolution [39], graph attention [40], and pooling [41] to instantiate $f_{\theta,l}$. The node representation for v at the input layer is denoted by $\mathbf{h}_{v,0}$ (*i.e.*, $l = 0$), where $\mathbf{h}_{v,0} = \mathbf{x}_v$ and $\mathbf{x}_v \in \mathbb{R}^F$. We designate the representation of node v at the final GNN layer $\mathbf{h}_v \in \mathbb{R}^D$, as its **base GNN representation**. In this work, we use GNNs as a collective term for neural networks that operate over graphs using localized message passing [159], as opposed to spectral methods [37] that learn convolutional filters from the entire graph.

3.2.3 Network Motifs

Network motifs are a general class of higher-order connectivity patterns, with a history of use in network science [10, 160]. A motif has several topologically equivalent appearances in the network called *motif instances*. Prior work [161, 162] shows how to efficiently compute motif instances for large graphs.

Definition 3.1 (Network Motif). A network motif $M_t = (\mathcal{V}_t, \mathcal{E}_t)$ is a connected, induced

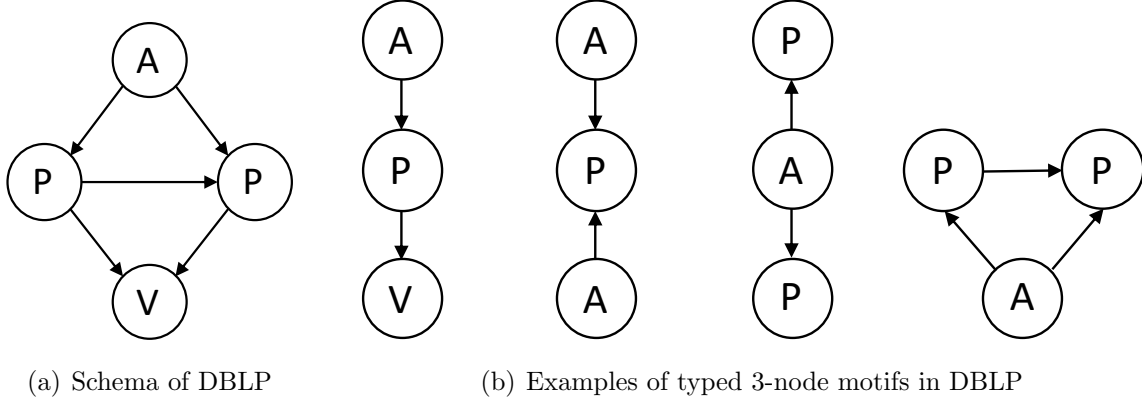


Figure 3.3: (a) *Heterogeneous network schema of academic citation network DBLP with three node types: Author (A), Paper (P) and Venue (V) and three edge types $A - P$, $P - V$ and $P - A$.* (b) *Examples of 3-node connected typed network motifs.*

subgraph consisting of a subset $\mathcal{V}_t \subset \mathcal{V}$ and $\mathcal{E}_t = \{e \in \mathcal{E} \mid e = (u, v), u, v \in \mathcal{V}_t\}$. Let k_t denote the number of nodes in network motif M_t ; that is, $k_t = |\mathcal{V}_t|$.

Here, we consider 3-node connected network motifs, *e.g.*, Figure 3.2 shows all 3-node, topologically distinct, directed (*e.g.*, citations) and undirected, connected network motifs.

In a heterogeneous graph, nodes/edges are of many different types which makes it essential to explicitly (and jointly) model the connectivity patterns and the participating types. We define *typed* network motifs that generalize network motifs through additional constraints on the types of participating nodes, which are formally described below as:

Definition 3.2 (Typed Network Motif). A typed network motif $M_t = (\mathcal{V}_t, \mathcal{E}_t, \psi_t, \xi_t)$ is a connected, induced subgraph consisting of a subset $\mathcal{V}_t \subset \mathcal{V}$ and $\mathcal{E}_t = \{e \in \mathcal{E} \mid e = (u, v), u, v \in \mathcal{V}_t\}$ such that node and edge type mappings $\psi_t = \psi|_{\mathcal{E}_t}$ and $\xi_t|_{\mathcal{V}_t}$ are restrictions of ψ and ξ to \mathcal{V}_t and \mathcal{E}_t respectively, and $k_t = |\mathcal{V}_t|$ is the number of nodes in M_t .

We assume that the given graph \mathcal{G} has a set of unique network motifs $\mathcal{M} = \{M_1, \dots, M_T\}$.

Definition 3.3 (Motif Instance). Let I_t be an induced subgraph of \mathcal{G} . We define I_t to be a motif instance of M_t if I_t is isomorphic to M_t . A motif M_t can have several motif instances in \mathcal{G} . While each such motif instance has a unique node set, two motif instances can share nodes. We denote the set of unique instances of M_t in \mathcal{G} that contain node v as $\mathcal{I}_v(M_t)$.

3.2.4 Model Regularization

We plan to use these local structural properties (*i.e.*, network motifs) to *regularize* the graph neural network model during training. Typically, we train GNNs by minimizing the

Symbol	Description
\mathcal{M}	Set $\{M_1, \dots, M_T\}$ of T network motifs
$\mathcal{I}_v(M_t)$	Set of instances of motif M_t in \mathcal{G} that contain node v
$\mathbf{h}_{v,l}$	Representation of node v at layer l of GNN
\mathbf{h}_v	Base GNN representation of node v (final layer)
\mathbf{h}_v^t	Motif-gated representation of node v for motif M_t
\mathbf{e}_{v,I_t}	Instance-specific representation of v in $I_t \in \mathcal{I}_v(M_t)$
$\mathbf{s}_{v,t}$	Motif-level representation of node v for motif M_t
\mathbf{z}_v	Final Representation of node v
α_{vt}	Task-specific importance of motif M_t to node v
β_v	Novelty score for training node $v \in \mathcal{V}_L$

Table 3.1: *Notation*

cross-entropy loss L_B , between model predictions $\hat{\mathbf{y}}_v \in \mathbb{R}^C$ and ground-truth labels $\mathbf{y}_v \in \mathbb{B}^C$ of labeled training nodes in $v \in \mathcal{V}_L$, which is defined below:

$$L_B = - \sum_{v \in \mathcal{V}_L} \sum_{c=1}^C y_{v,c} \log \hat{y}_{v,c} \quad (3.2)$$

where the c -th index of the one-hot vector $\hat{\mathbf{y}}_{v,c}$ is the probability that v belongs to the true class c . Notice that the loss L_B is agnostic to any local structural properties (*e.g.*, mixing patterns in social networks [163]) that may be indicative of the true node class. Thus, we develop a modified loss $L'_B = L_B + \lambda L_R$, where L_R is the regularization loss that incorporates attributed motif structure and λ is a constant. Our goal is to design L_R to overcome the two limitations of message-passing models: localized and over-smoothed node representations.

3.3 INFOMOTIF FRAMEWORK

In this section, we first discuss the structural properties of GNNs to motivate the notion of attributed structural roles. In section 3.3.2, we present our motif-based self-supervised learning framework INFOMOTIF to regularize GNNs based on a single motif. Finally, in section 3.3.3, we introduce our overall framework with a novel multi-motif training curriculum.

3.3.1 Motivating Insights: Attributed Structural Roles

A k -layer GNN computes a localized representation $\mathbf{h}_{v,k}$ for each node v that incorporates information from its k -hop neighborhood, denoted by $\mathcal{N}_k(v)$. For a node set $S \subseteq \mathcal{V}$, let

$\mathcal{N}_k(S) = \bigcup_{v \in S} \mathcal{N}_k(v)$ define its k -hop neighborhood, and $\mathbf{X}(S)$ denote its set of input node features. Let $\mathbf{Y}(\mathcal{V}_L)$ comprise the training labels of nodes in the labeled set \mathcal{V}_L . For a k -layer GNN trained on \mathcal{V}_L using loss L_B (Equation (3.2)), let $\Theta^* = \{\Theta_1, \dots, \Theta_k\}$ be the optimal parameters computed by its training algorithm. Now, we have the following proposition.

Proposition 3.1. The optimal parameter set Θ^* is a function of $\mathbf{X}(\mathcal{N}_k(\mathcal{V}_L))$ and $\mathbf{Y}(\mathcal{V}_L)$; further, any changes in inputs $\mathbf{X}(\mathcal{V} \setminus \mathcal{N}_k(\mathcal{V}_L))$ do not affect Θ^* .

Proof Sketch. By an induction argument, the supervised loss L_B can be written as $g(\Theta_1, \dots, \Theta_k, \mathbf{Y}(\mathcal{V}_L), \mathbf{X}(\mathcal{N}_k(\mathcal{V}_L)))$ for some function $g(\cdot)$. Thus, when the GNN is trained on L_B using gradient updates, the optimal Θ^* must be independent of $\mathbf{X}(\mathcal{V} \setminus \mathcal{N}_k(\mathcal{V}_L))$.

Notice that the addition of a standard regularization term (*e.g.*, L_1 or L_2 regularization) only impacts $\{\Theta_1, \dots, \Theta_k\}$; thus, the overall loss function still remains independent of $\mathcal{V} \setminus \mathcal{N}_k(\mathcal{V}_L)$, satisfying proposition 3.1.

Hence, the optimal parameters Θ^* of a k -layer GNN are only affected by node features in the k -hop neighborhood $\mathcal{N}_k(\mathcal{V}_L)$ of the labeled set \mathcal{V}_L , *i.e.*, the features and structural connectivities of nodes in $\mathcal{V} \setminus \mathcal{N}_k(\mathcal{V}_L)$ are ignored during the training process.

Let the k -hop neighborhood of class c be $\mathcal{N}_k(\mathcal{V}_L(c))$ where $\mathcal{V}_L(c) = \{v \in \mathcal{V}_L : y_{vc} = 1\}$ is the set of nodes labeled with class c . Let $L_B(c)$ denote the supervised loss term specific to class c . Now, the corollary directly follows from proposition 3.1:

Corollary 3.1. If node $v \notin \mathcal{N}_k(\mathcal{V}_L(c))$, the k -hop neighborhood of class c , then the supervised loss term $L_B(c)$ for class c is independent of v .

The above corollary states that gradients from the supervised loss $L_B(c)$ for class c cannot reach nodes that lie outside the k -hop neighborhood of class c , *i.e.*, $\mathcal{N}_k(\mathcal{V}_L(c))$. To illustrate its implications, we revisit Figure 3.1. Since node c lies beyond the 2-hop neighborhood of node b , node c does not affect the training loss at node b (which belongs to class 2). Thus, despite nodes c and b having identical co-variation of attributes and structure (blue neighbors form triangles), node c does not influence the training loss for nodes with class 2.

3.3.2 Self-supervised Single Motif Regularization

In this section, we introduce INFOMOTIF, a framework to regularize node representations of the base GNN by exploiting the co-variance of node attributes and motif structures. We define *attributed structural roles* by assigning the same role to nodes that participate in motif instances over *co-varying* sets of attributes. Compared to prior role-aware models [24] that discover structurally similar nodes *agnostic* to attributes, we define roles based on

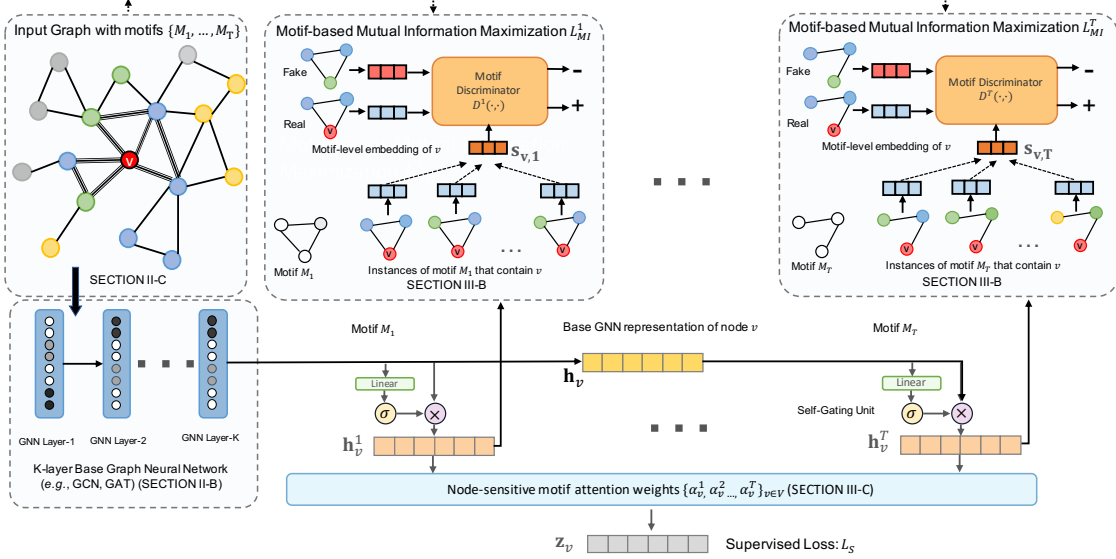


Figure 3.4: Neural architecture diagram of INFOMOTIF depicting the different model components: base GNN f_{θ_I} with k layers (bottom left), motif-based mutual information maximizing regularizers L_{MI}^I (top right), and motif-attention module to compute final node representations \mathbf{z}_v (bottom right). Instances of motif M_1 are shown in the graph (top left) with textured lines and colors indicate node attributes.

attribute occurrence in higher-order connectivity structures. In heterogeneous graphs, attributed structural roles further incorporate the semantics of node and edge types described by the connectivity structures of typed network motifs.

Now, we describe our self-supervised learning strategy to learn attribute co-variance for a single motif. In the next section, we extend these formulations to handle multiple motifs.

Motif-based Mutual Information

We first consider a single network motif type $M_t \in \mathcal{M}$ and a specific node $v \in \mathcal{V}$ to learn attribute co-variance across instances $\mathcal{I}_v(M_t)$ that contain v in the graph. To learn attributed structural roles, it is necessary to *contrast* the attributed instances of motif M_t against attributed node combinations that are not present in any instances of M_t .

We maximize the motif-based *mutual information* (MI) between a *motif-level* representation of v and corresponding *instance-specific* representations centered at v . By introducing motif-based MI maximization as a regularizer, the GNN is encouraged to learn *discriminative* statistical correspondences between nodes that participate in instances of the same motif. Motif-based MI maximization is an example of the broader paradigm of self-supervised learning that derives auxiliary supervision signals from the intrinsic structure (*e.g.*, connectivity

patterns in a network motif) of the underlying data.

We first adapt the base GNN representation \mathbf{h}_v (see Section 3.2.2), specific to motif M_t through a *motif gating* function $f_{\text{GATE}}^t : \mathbb{R}^D \mapsto \mathbb{R}^D$ resulting in a gated embedding \mathbf{h}_v^t . Then, we introduce a *motif instance encoder* $f_{\text{ENC}}^t : \mathbb{R}^D \times \mathbb{R}^{(k_t \times D)} \mapsto \mathbb{R}^D$ to compute the instance-specific representation $\mathbf{e}_{v,I_t} \in \mathbb{R}^D$ of node v conditioned on other co-occurring nodes in instance $I_t \in \mathcal{I}_v(M_t)$. Finally, the motif-level representation $\mathbf{s}_{v,t} \in \mathbb{R}^D$ of node v summarizes the set of instance-specific representations $\{\mathbf{e}_{v,I_t}\}_{I_t \in \mathcal{I}_v(M_t)}$ through a permutation-invariant *motif readout* function $f_{\text{READ}}^t(\cdot)$, *e.g.*, averaging or pooling functions.

For each node $v \in \mathcal{V}$, we maximize motif-based mutual information L_{MI}^t between its instance-specific representations $\{\mathbf{e}_{v,I_t}\}_{I_t \in \mathcal{I}_v(M_t)}$ and motif-level representation $\mathbf{s}_{v,t}$, by defining I_{ψ^t} as a mutual information estimator for motif M_t that is *shared* across all nodes. The resulting learning objective is given by:

$$L_{MI}^t(\theta, \phi^t, \psi_t) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \sum_{I_t \in \mathcal{I}_v(M_t)} I_{\psi^t}(\mathbf{e}_{v,I_t}; \mathbf{s}_{v,t}) \quad (3.3)$$

where θ and ϕ^t denote the parameters of the layers $\{f_{\theta,l}\}_{l=1}^k$, and motif-specific transforms $\{f_{\text{GATE}}^t, f_{\text{ENC}}^t, f_{\text{READ}}^t\}$ respectively. By maximizing MI across all instances of motif M_t in the graph through a shared MI estimator I_{ψ^t} , we enable the GNN to learn correspondences between a pair of potentially distant nodes that participate in instances of motif M_t .

Mutual Information Maximization

Following prior neural MI estimation methods [144, 164], we model the MI estimator I_{ψ^t} as a *discriminator* network that learns a decision boundary to accurately distinguish between *positive* samples drawn from the joint distribution and *negative* samples drawn from the product of marginal distributions. We train a *contrastive* discriminator network $\mathbf{D}_{\psi}^t : \mathbb{R}^D \times \mathbb{R}^D \mapsto \mathbb{R}^+$, where $\mathbf{D}_{\psi}^t(\mathbf{e}_{v,I_t}, \mathbf{s}_{v,t})$ denotes the probability score assigned to this instance-motif pair (higher scores for observed instances of motif M_t). The positive samples $(\mathbf{e}_{v,I_t}, \mathbf{s}_{v,t})$ for \mathbf{D}_{ψ}^t are the representations \mathbf{e}_{v,I_t} of observed instances $I_t \in \mathcal{I}_v(M_t)$ of motif M_t paired with the motif-level representation $\mathbf{s}_{v,t}$. The negative samples $(\mathbf{e}_{v,\tilde{I}_t}, \mathbf{s}_{v,t})$ are derived by pairing $\mathbf{s}_{v,t}$ with the representations $\mathbf{e}_{v,\tilde{I}_t}$ of negative instances \tilde{I}_t sampled from a negative sampling distribution $P_{\mathcal{N}}(\tilde{I}_t | M_t)$. The discriminator network \mathbf{D}_{ψ}^t is trained on a noise-contrastive objective L_{MI}^t with a binary cross-entropy loss between samples drawn from the joint distribution (positive pairs), and the product of marginals (negative pairs), which is defined as in the following objective below:

$$L_{MI}^t = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} L_{MI}^t(v) = -\frac{1}{2Q|\mathcal{V}|} \sum_{v \in \mathcal{V}} \sum_{i=1}^Q \left[\mathbb{E}_{I_t} \log \mathbf{D}_{\psi}^t(\mathbf{e}_{v,I_t}, \mathbf{s}_{v,t}) + \mathbb{E}_{\tilde{I}_t} \log(1 - \mathbf{D}_{\psi}^t(\mathbf{e}_{v,\tilde{I}_t}, \mathbf{s}_{v,t})) \right] \quad (3.4)$$

where Q is the number of observed motif instances sampled per node. The above objective Equation (3.4) maximizes MI between $\mathbf{s}_{v,t}$ and $\{\mathbf{e}_{v,I_t}\}_{I_t \in \mathcal{I}_v(M_t)}$ based on the Jensen-Shannon Divergence between their joint distribution and product of marginals [44].

We design the negative sampling distribution $P_{\mathcal{N}}(\tilde{I}_t | M_t)$ to learn attribute co-variance in instances of motif M_t . For each positive instance I_t , the generated negative instance \tilde{I}_t is topologically equivalent but contains attributes that do not occur in instances of M_t in \mathcal{G} . By contrasting the observed instances of M_t against fake instances with perturbed attributes, \mathbf{D}_{ψ}^t learns attributed structural roles with respect to network motif M_t .

3.3.3 Multi-Motif Regularization Framework

Now, we extend our framework for any graph that includes a set of motifs $\mathcal{M} = \{M_1, \dots, M_T\}$. A typical way to include regularizers (Equation (3.4)) from multiple motifs is given by:

$$L' = L_B + \lambda L'_{MI} = L_B + \lambda \cdot \frac{1}{T} \sum_{t=1}^T L_{MI}^t \quad (3.5)$$

where λ is a tunable hyper-parameter to balance the supervised classification loss L_B and motif regularizers. Intuitively, each motif $M_t \in \mathcal{M}$ is a connectivity pattern that can be viewed as defining one kind of structural role, *e.g.*, bridge nodes. Each motif has a different significance towards the learning task. Thus, a multi-motif framework should automatically identify the significance of different motifs without manual hand tuning.

In addition, real-world networks exhibit heavy-tailed degree and community distributions [165], which manifest as *skewed* (imbalanced) motif occurrences among nodes as well as across motif types. This further complicates the learning process of incorporating multiple motifs as regularizers. We identify three key aspects *task-oriented*, *node-sensitive*, and *skew-aware* that are critical to the design of a multi-motif learning framework:

- **Task:** Distinguish the significance of different motifs to compute node representations conditioned on the underlying semi-supervised learning task.

- **Node:** Expressive power to control the extent of regularization exerted by each motif at a node-level granularity.
- **Skew:** Adapt to varying levels of motif occurrence skew without any distributional assumptions on the input graph.

To address these objectives, we first describe our approach to compute final node representations conditioned on multiple motifs, followed by two novel online reweighting strategies.

Task-driven Representations

The base GNN is trained by a supervised task loss L_B (Equation (3.2)) over the labeled node set \mathcal{V}_L . We instead aggregate the set of motif-gated representations (\mathbf{h}_v^t for motif $M_t \in \mathcal{M}$), to compute the final representation $\mathbf{z}_v \in \mathbb{R}^D$ for node v . We learn attention weights α_{vt} to characterize the task-driven importance of motif M_t to node v and compute \mathbf{z}_v through a weighted average, given by:

$$\mathbf{z}_v = \sum_{t=1}^T \alpha_{vt} \mathbf{h}_v^t \quad \alpha_{vt} = \frac{\exp(\mathbf{p} \cdot \mathbf{h}_v^t)}{\sum_{t'=1}^T \exp(\mathbf{p} \cdot \mathbf{h}_v^{t'})} \quad (3.6)$$

where $\mathbf{p} \in \mathbb{R}^D$ defines the attention function and is learned by optimizing the final representations $\{\mathbf{z}_v\}_{v \in \mathcal{V}_L}$ of labeled nodes \mathcal{V}_L using the supervised loss L_B (Equation (3.2)). The final representation \mathbf{z}_v of each node $v \in \mathcal{V}$ is used for node classification.

Node-sensitive Motif Regularization

Instead of using static uniform weights to incorporate motif regularizers (Equation (3.5)), we contextually weight the contributions of different motif regularization terms $L_{MI}^t(v)$ from Equation (3.4) at a node-level granularity through the attention weights α_{vt} of motif M_t for node v . This results in the following motif-regularization loss:

$$L_{MI} = \frac{1}{nT} \sum_{t=1}^T \sum_{v \in \mathcal{V}} \alpha_{vt} L_{MI}^t(v) \quad (3.7)$$

The loss L_{MI} varies the extent of regularization per node in proportion to the task-specific importance α_{vt} of motif M_t to node v . Notice that the attention function is learned by training the final representations \mathbf{z}_v of labeled nodes $v \in \mathcal{V}_L$ on the supervised loss L_B ; in contrast,

Algorithm 3.1: *The framework of INFOMOTIF-GNN.*

Input: Graph \mathcal{G} , Labeled node set \mathcal{V}_L , Base GNN $\{f_{\theta,l}\}_{l=1}^k$

Output: Motif-regularized embedding \mathbf{z}_v for each node $v \in \mathcal{V}$

- 1: Initialize sample novelty weights $\beta_v = 1 \ \forall v \in \mathcal{V}_L$
 - 2: **while** *not converged* **do**
 - ▷ **Supervised loss over labeled node set \mathcal{V}_L**
 - 3: **for** each batch of nodes $\mathcal{V}_B \subseteq \mathcal{V}_L$ **do**
 - 4: Fix sample weights $\{\beta_v\}_{v \in \mathcal{V}_B}$ and optimize L_S on \mathcal{V}_B using mini-batch gradient descent (Equation 3.9).
 - 5: Compute motif attention weights $\{\alpha_v\}_{v \in \mathcal{V}}$ (Equation 3.6).
 - ▷ **Motif-based InfoMax loss over entire node set \mathcal{V}**
 - 6: **for** each batch of nodes $\mathcal{V}_B \subseteq \mathcal{V}$ **do**
 - 7: Fix motif weights $\{\alpha_v\}_{v \in \mathcal{V}}$ and optimize L_{MI} on \mathcal{V}_B using mini-batch gradient descent (Equation 3.7)
 - 8: Compute sample weights $\{\beta_v\}_{v \in \mathcal{V}_L}$ (Equation 3.8).
 - 9: Compute $\mathbf{z}_v \in \mathbb{R}^D \ \forall v \in \mathcal{V}$ (Equation 3.6)
-

the motif-regularization loss L_{MI} (which operates on all nodes) re-weights each motif loss term $L_{MI}^t(v)$ per node v with the estimated attention weight α_{vt} from Equation (3.6).

Skew-aware Sample Weighting

Prior work in curriculum and meta learning has shown the importance of re-weighting training examples to overcome training set biases [166]. In particular, re-weighting strategies that emphasize harder examples are effective at handling imbalanced data distributions [167]. We propose a *novelty-driven* re-weighting strategy to handle skew in motif occurrences across nodes and motif types.

The novelty β_v of node v is a function of its motif distribution, *i.e.*, novel nodes contain uncommon motif types in their neighborhood, which in turn reflects in their attention weight distribution over motifs. Let $\alpha_v \in \mathbb{R}^T$ denote the vector of attention weights for a labeled node v over the motif set \mathcal{M} . Now, the novelty β_v of node v is high if its motif distribution α_v significantly diverges from those of other nodes. We quantify β_v by the deviation (measured by euclidean distance) of α_v from the mean motif distribution of labeled nodes $v \in \mathcal{V}_L$.

$$\beta_v = \frac{\exp(\|\alpha_v - \mu\|^2)}{\sum_{u \in \mathcal{V}_L} \exp(\|\alpha_u - \mu\|^2)} \quad \mu = \frac{1}{|\mathcal{V}_L|} \sum_{v \in \mathcal{V}_L} \alpha_v \quad (3.8)$$

The novelty scores are normalized over \mathcal{V}_L using a softmax function, to give non-negative

sample weights $0 < \beta_v \leq 1$. We now define the novelty-weighted supervised loss L_S as:

$$L_S = - \sum_{v \in \mathcal{V}_L} \beta_v \sum_{c=1}^C y_{vc} \log \hat{y}_{vc} \quad (3.9)$$

In contrast to the original supervised loss L_B (Equation (3.2)), the re-weighted objective L_S induces a novelty-weighted training curriculum that progressively focuses on harder samples.

Model Training

The overall objective of INFOMOTIF is composed of two terms, the re-weighted supervised loss L_S (Equation (3.9)), and motif regularizers (Equation (3.7)), given by:

$$L = L_S + \lambda L_{MI} \quad (3.10)$$

We optimize L_S and L_{MI} alternatively at each training epoch, which removes the need to tune balance hyper-parameter λ . Algorithm 3.1 summarizes the training procedure.

Complexity Analysis

On the whole, the complexity of our model is $O(\mathbf{F}) + O(nTQD + nTD^2)$ where $O(\mathbf{F})$ is the base GNN complexity, T is the number of motifs, Q is sampled instance count per motif, and D the latent space dimensionality. Since $T \ll n$ and $Q \ll n$, the added complexity of our INFOMOTIF framework scales linearly with respect to the number of nodes.

3.4 MODEL DETAILS

We now discuss the architectural details of our framework: motif instance encoder, gating, readout, and discriminator.

3.4.1 Motif Gating

We design a pre-filter with *self-gating units* (SGUs) to regulate the flow of information from the base GNN embedding \mathbf{h}_v to the motif-based regularizer. The SGU for motif \mathbf{M}_t , denoted by $f_{\text{GATE}}^t(\cdot)$, learns a non-linear gating function to modulate the input \mathbf{h}_v at a feature-wise granularity through dimension re-weighting. In particular, we define the self-gating unit

for motif M_t as:

$$\mathbf{h}_v^t = f_{\text{GATE}}^t(\mathbf{h}_v) = \mathbf{h}_v \odot \sigma(\mathbf{W}_g^t \mathbf{h}_v + \mathbf{b}_g^t) \quad (3.11)$$

where $\mathbf{W}^t \in \mathbb{R}^{D \times D}, \mathbf{b}^t \in \mathbb{R}^D$ are learned parameters, \odot denotes the element-wise product operation, and σ is the sigmoid non-linearity. The self-gating mechanism effectively serves as a multiplicative skip-connection [168] that facilitates gradient flow from the motif-based regularizer to the layers of the base GNN.

3.4.2 Motif Instance Encoder

The encoder $f_{\text{ENC}}(\cdot)$ computes the instance-specific representation \mathbf{e}_{v,I_t} for node v conditioned on the gated representations $\{\mathbf{h}_u^t\}_{u \in I_t}$ of the nodes in instance I_t . We apply self-attentions [169] to compute a weighted average of the gated node representations $\{\mathbf{h}_u^t\}_{u \in I_t}$ in I_t . Specifically, f_{ENC} attends over each node $u \in I_t$ to compute attention weight α_u by comparing its gated representation \mathbf{h}_u^t with that of node v , \mathbf{h}_v^t , using an attention network that is parameterized by a single MLP layer, defined by:

$$\mathbf{e}_{v,I_t} = \sum_{u \in I_t} \alpha_u \mathbf{h}_u^t \quad \alpha_u = \frac{\exp(\mathbf{a}^t \cdot [\mathbf{h}_u^t || \mathbf{h}_v^t])}{\sum_{u' \in I_t} \exp(\mathbf{a}^t \cdot [\mathbf{h}_{u'}^t || \mathbf{h}_v^t])} \quad (3.12)$$

where $\mathbf{a}^t \in \mathbb{R}^{2D}$ is a weight vector parameterizing the attention function and $||$ denotes concatenation. We empirically find self-attentional encoders to outperform pooling alternatives.

3.4.3 Motif Readout

The motif readout function $f_{\text{READ}}^t(\cdot)$ summarizes the set of instance-specific representations $\{\mathbf{e}_{v,I_t}\}_{I_t \in \mathcal{I}_v(M_t)}$ to compute the motif-level representation $\mathbf{s}_{v,t}$ for motif M_t . Here, we use a simple averaging of the instance-specific representations to formulate the motif readout function $f_{\text{READ}}^t(\cdot)$ as defined below:

$$\mathbf{s}_{v,t} = f_{\text{READ}}^t(\{\mathbf{e}_{v,I_t}\}_{I_t \in \mathcal{I}_v(M_t)}) = \sigma\left(\sum_{I_t \in \mathcal{I}_v(M_t)} \frac{\mathbf{e}_{v,I_t}}{|\mathcal{I}_v(M_t)|}\right) \quad (3.13)$$

where σ is the sigmoid non-linearity. We adopt a batch-wise model training strategy with motif instance sampling (~ 20 per node) to compute the motif-level representation $\mathbf{s}_{v,t}$. While we find this simple readout function to be empirically effective, sophisticated architectures [170] are likely necessary to handle larger sample sizes.

3.4.4 Motif Discriminator

The discriminator D_ψ^t learns a motif-specific scoring function to assign higher likelihoods to observed instance-motif pairs relative to negative examples. Similar to prior work [44, 171], we use a bilinear scoring function defined by:

$$D_\psi^t(\mathbf{e}_{v,I_t}, \mathbf{s}_v^t) = \sigma(\mathbf{e}_{v,I_t} \cdot \mathbf{W}_d^t \mathbf{s}_v^t) \quad (3.14)$$

where $\mathbf{W}_d^t \in \mathbb{R}^{D \times D}$ is a trainable scoring matrix and σ is the sigmoid non-linearity to convert raw scores into probabilities of $(\mathbf{e}_{v,I_t}, \mathbf{s}_v^t)$ being a positive example for motif M_t .

3.5 EXPERIMENTS

We present extensive quantitative and qualitative analyses on multiple diverse datasets spanning homogeneous and heterogeneous graphs. We first introduce datasets, baselines, and experimental setup (Section 3.5.1, 3.5.2, and 3.5.3), followed by node classification results in Sections 3.5.4 and 3.5.5 by integrating three GNN models in our framework. We propose four research questions to guide our experiments:

- (RQ₁) How does INFOMOTIF compare with state-of-the-art graph neural networks and embedding learning methods on *node classification* over *homogeneous* graphs?
- (RQ₂) Is INFOMOTIF effective for *node classification* on *heterogeneous* information networks (multiple types of nodes and edges) compared to state-of-the-art approaches?
- (RQ₃) How do the different *architectural* design choices and *training* strategies in INFOMOTIF impact performance?
- (RQ₄) What is the impact of *node degree*, local training *label sparsity*, and local *attribute diversity*, on the classification performance of INFOMOTIF?
- (RQ₅) How do the motif-based *regularization* strategies and associated *hyper-parameters* in INFOMOTIF affect model training time and performance?

3.5.1 Datasets

Our experiments are designed towards semi-supervised node classification on nine real-world benchmark datasets, divided across homogeneous and heterogeneous networks. In

Dataset	Citation Networks			Air-Traffic Networks		
	Cora	Citeseer	Pubmed	Brazil	Europe	USA
# Nodes	2,485	2,110	19,717	131	399	1,190
# Edges	5,069	3,668	44,324	1,038	5,995	13,599
# Attributes	1,433	3,703	500	-	-	-
# Classes	7	6	3	4	4	4

Table 3.2: *Dataset statistics of homogeneous network benchmarks, including three assortative citation [34] networks and three dis-assortative air-traffic [24] networks. Ground-truth classes in citation networks exhibit attribute homophily; ground-truth classes in flight networks indicate node structural roles.*

Dataset	DBLP-A	DBLP-P	Movie
# Nodes	11,170	35,770	10,441
# Edges	24,846	131,636	99,509
# Attributes	4,479	11,680	4,577
# Node Types	3	3	4
# Classes	4	10	6

Table 3.3: *Dataset statistics of three heterogeneous graphs across bibliographic and movie information networks, with multiple node and edge types. Networks schemas are shown in Figures 3.3(a) and 3.5(a).*

homogeneous graphs, we experiment on two diverse types of datasets: *citation networks* that exhibit strong homophily and *air-traffic networks* that depend on structural roles. Table 3.2 presents detailed statistics for each dataset.

- **Citation Networks:** We consider three benchmark datasets: Cora, Citeseer, and PubMed [34], where nodes correspond to documents and edges represent citation links. Each document is associated with a bag-of-words feature vector and the task is to classify documents into different research topics.
- **Air-Traffic Networks:** We use three undirected networks: Brazil, Europe, and USA [24] where nodes correspond to airports and edges indicate the existence of commercial flights. Class labels are assigned based on activity level, measured by the cardinality of flights or people that passed the airports. We use one-hot indicator vectors as node attributes. Here, class labels are related to the role played by airports.

We conduct experiments on three real-world datasets over heterogeneous graphs whose statistics are shown in Table 3.3:

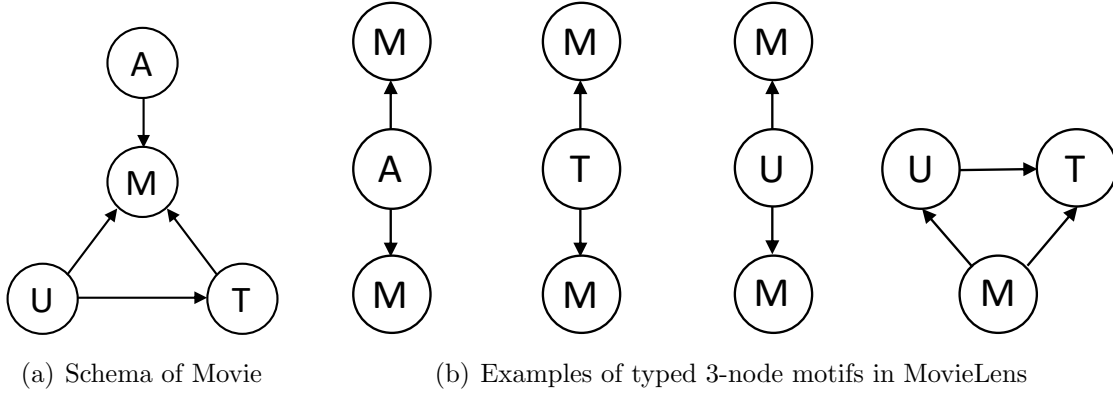


Figure 3.5: (a) *Heterogeneous network schema of movie network with four node types: Actor (A), Movie (M), User (U) and Term (T) and four edge types $A-M$, $U-M$, $T-M$, and $U-T$.* (b) *Examples of 3-node connected typed network motifs.*

- **DBLP-A:** This is a bibliographic network composed of 3 node types: author (A), paper (P) and venue (V), connected by three link types: $P \rightarrow P$, $A-P$ and $P-V$ (Figure 3.3(a)). We use a subset of DBLP [172] with text attributes of papers to classify authors based on their research areas.
- **DBLP-P:** This dataset has the same schema as DBLP-A, but the learning task is to classify research papers into ten categories, which are obtained from Cora [173].
- **Movie:** We use MovieLens [174] with four node types: movie (M), user (U), actor (A) and tag (T) linked by four types: $U-M$, $A-M$, $U-T$ and $M-T$, with attributes for actors and movies. The classification task is movie genre prediction.

3.5.2 Baselines

We first introduce baseline methods designed for learning over homogeneous graphs, organized into four categories based on whether they are *proximity-based vs. structural*; and the paradigm of *embedding learning vs. graph neural networks*:

- **Proximity-based embedding methods:** Conventional methods node2vec [18] that employs second-order proximity, and motif2vec [175] based on higher-order proximity.
 - node2vec [18]: Skip-gram model to learn unsupervised node representations that capture node co-occurrences in second-order random walk samples.

- motif2vec [175]: Higher-order extension of skip-gram embedding models to model higher-order proximity induced by motif structures.
- **Structural embedding methods:** Structural role-aware embedding learning models struc2vec [24], GraphWAVE [176], and DRNE [177].
 - struc2vec [24]: Learns node representations capturing structural identity by utilizing degree sequences at different local neighborhoods.
 - GraphWAVE [176]: Learns node embeddings that capture structural similarity by leveraging spectral graph wavelet diffusion patterns.
 - DRNE [177]: Deep recursive neural network to learn node embeddings that preserve the property of regular equivalence.
- **Standard Graph Neural Networks:** We compare against several state-of-the-art GNNs based on localized message passing:
 - GCN [39]: Graph convolutional networks operating on the graph Laplacian matrix using degree-weighted neighborhood aggregation.
 - GraphSAGE [41]: Scalable graph neural networks with neighborhood sampling and aggregation using a variety of pooling functions (max, mean, and LSTM)
 - GAT [40]: Graph attention networks with self-attentional neighbor aggregation.
 - JK-Net [42]: Jumping knowledge networks that use skip-connections to vary the influence radius (per node) for neighborhood aggregation.
 - DGI [44]: Deep Graph Infomax that learns node embeddings by maximizing mutual information between local and global representations of the same graph.
- **Structural Graph Neural Networks:** Motif-based GNN models Motif-CNN [46], MCN [47] and degree-specific DEMO-Net [45].
 - Motif-CNN [46]: Motif-based graph convolutional networks with attention-based cross-motif aggregation.
 - MCN [47]: Motif-based graph attentional model that uses weighted multi-hop motif adjacency matrices to capture higher-order neighborhood information.
 - DEMO-Net [45]: Degree-specific graph neural network that uses multi-task graph convolutions to learn degree-aware node embeddings.

Next, we introduce graph neural networks designed specifically for semi-supervised learning over heterogeneous graphs.

- **Heterogeneous Graph Neural Networks:** State-of-the-art *metapath*-based GNN models MAGNN [57], HAN [56], and *metagraph*-aware GNN model Meta-GNN [58].
 - MAGNN [57]: Learns heterogeneous node embeddings via intra-metapath aggregation to incorporate intermediate semantic nodes, and the inter-metapath aggregation to combine messages from multiple metapaths.
 - HAN [56]: Learns metapath-specific node embeddings from different metapath-based homogeneous graphs using an attention mechanism.
 - Meta-GNN [58]: Learns node embeddings to extract and aggregate features from local metagraph-structured neighborhoods in heterogeneous graphs.

3.5.3 Experimental Setup

We tested INFOMOTIF by integrating GCN, JK-Net and GAT as base graph neural network models within our framework. We only consider the largest connected component in each dataset and evaluate different train/validation/test splits to fairly compare different models [178]. We create 10 random data splits per training ratio and evaluate the mean test classification accuracy along with standard deviation.

All experiments were conducted on a Tesla K-80 GPU using PyTorch. Our implementation of INFOMOTIF is publicly available². For citation networks, we use two-layer base GNNs with layer sizes of 256 each, while using 64 for the smaller air-traffic networks. We train the base JK-Net model using 4 GCN layers and maxpool layer aggregation, while the base GAT model learns 8 attention heads per layer. The model is trained for a maximum of 100 epochs with a batch size of 256 nodes with Adam optimizer. We also apply dropout with a rate of 0.5, and tune the learning rate in the range $\{10^{-4}, 10^{-3}, 10^{-2}\}$.

3.5.4 Homogeneous Graphs (RQ₁)

In homogeneous graphs, we train INFOMOTIF using the set of all directed 3-node motifs in citation networks and undirected 3-node motifs in air-traffic networks (Figure 3.2).

²<https://github.com/CrowdDynamicsLab/InfoMotif>

Training Ratio	Data		Cora		Citeseer		PubMed	
	X	Y	20%	40%	20%	40%	20%	40%
PROXIMITY-BASED GRAPH EMBEDDING METHODS								
Node2Vec [18]			75.7 \pm 0.5	76.1 \pm 0.5	68.1 \pm 0.5	69.1 \pm 0.6	80.1 \pm 0.6	80.2 \pm 0.6
Motif2Vec [175]			79.0 \pm 0.4	79.2 \pm 0.4	66.6 \pm 0.4	67.1 \pm 0.3	79.8 \pm 0.2	79.8 \pm 0.4
STRUCTURAL GRAPH EMBEDDING METHODS								
Struct2Vec [24]			35.4 \pm 1.0	37.6 \pm 1.3	31.2 \pm 0.8	35.1 \pm 0.9	48.5 \pm 0.3	49.2 \pm 0.4
GraphWave [176]			39.5 \pm 2.1	41.1 \pm 1.5	38.5 \pm 1.2	40.6 \pm 0.9	43.0 \pm 2.0	43.3 \pm 1.3
DRNE [177]			34.9 \pm 1.5	36.5 \pm 1.5	30.8 \pm 1.2	32.2 \pm 1.2	40.4 \pm 0.7	41.6 \pm 0.4
STANDARD GRAPH NEURAL NETWORKS								
GCN [39]	✓	✓	81.6 \pm 0.5	82.0 \pm 0.4	75.8 \pm 0.5	76.6 \pm 0.3	85.7 \pm 0.7	86.1 \pm 0.5
GAT [40]	✓	✓	80.9 \pm 0.7	81.4 \pm 0.2	74.5 \pm 0.7	75.5 \pm 0.7	83.3 \pm 0.3	84.2 \pm 0.3
GraphSAGE [41]	✓	✓	81.3 \pm 0.3	83.5 \pm 0.3	72.9 \pm 0.3	73.8 \pm 0.2	86.6 \pm 0.2	87.2 \pm 0.3
JKNet [42]	✓	✓	81.3 \pm 0.8	83.6 \pm 0.8	71.5 \pm 0.8	72.5 \pm 0.7	82.2 \pm 0.4	83.8 \pm 0.5
DGI [44]	✓		76.2 \pm 0.8	77.3 \pm 0.9	74.5 \pm 0.7	74.7 \pm 0.7	78.2 \pm 0.9	78.5 \pm 0.9
STRUCTURAL GRAPH NEURAL NETWORKS								
DemoNet [45]	✓	✓	81.0 \pm 0.6	82.4 \pm 0.5	67.9 \pm 0.7	68.5 \pm 0.6	79.5 \pm 0.4	80.5 \pm 0.4
Motif-CNN [46]	✓	✓	81.6 \pm 0.5	82.8 \pm 0.5	73.4 \pm 0.3	76.8 \pm 0.3	87.3 \pm 0.1	87.5 \pm 0.1
MCN [47]	✓	✓	81.1 \pm 0.9	82.4 \pm 0.8	73.2 \pm 0.4	75.9 \pm 0.7	85.2 \pm 0.6	85.9 \pm 0.5
MOTIF-REGULARIZED GRAPH NEURAL NETWORKS (InfoMotif)								
InfoMotif-GCN	✓	✓	85.7 \pm 0.4	87.4 \pm 0.4	77.7 \pm 0.5	78.5 \pm 0.5	87.5 \pm 0.2	88.3 \pm 0.2
InfoMotif-JKNet	✓	✓	85.5 \pm 0.3	86.5 \pm 0.5	74.5 \pm 0.8	76.7 \pm 0.9	87.0 \pm 0.2	87.9 \pm 0.3
InfoMotif-GAT	✓	✓	85.5 \pm 0.3	87.2 \pm 0.7	76.5 \pm 0.5	77.0 \pm 0.4	85.9 \pm 0.4	86.2 \pm 0.5

Table 3.4: Node classification results (% test accuracy) on assortative citation networks. **X** and **Y** denote the use of attributes and labels respectively. We report mean classification accuracy and standard deviation over 5 trials. We show GraphSAGE results with the best performing aggregator (among Max, Mean and LSTM). INFO-MOTIF improves results of all base GNNs by 3.5% on average across datasets.

We evaluate different train/validation/test splits (training ratios of 20%, and 40%) and report experimental results comparing INFOMOTIF with three base GNNs, against competing baselines on citation and air-traffic networks, in Tables 3.4 and 3.5 respectively.

In citation network datasets, GNN models generally outperform conventional embedding methods. Moreover, attribute-agnostic structural embedding methods perform poorly and structural GNNs perform comparably to standard message-passing GNNs. Citation networks exhibit strong attribute homophily in local neighborhoods; thus, structural GNNs do not provide much performance benefits over state-of-the-art message-passing GNNs. In contrast, our framework INFOMOTIF regularizes GNNs to discover distant nodes with similar attributed structures across the entire graph. INFOMOTIF achieves consistent average

Training Ratio	Data		USA		Europe		Brazil	
	X	Y	20%	40%	20%	40%	20%	40%
PROXIMITY-BASED GRAPH EMBEDDING METHODS								
Node2Vec [18]			24.6 \pm 0.9	24.8 \pm 0.9	36.5 \pm 1.0	37.4 \pm 1.1	26.3 \pm 1.4	30.4 \pm 1.3
Motif2Vec [175]			51.3 \pm 1.1	54.8 \pm 1.1	37.1 \pm 1.2	38.1 \pm 1.2	27.2 \pm 1.5	33.9 \pm 1.5
STRUCTURAL GRAPH EMBEDDING METHODS								
Struct2Vec [24]			50.4 \pm 0.8	51.3 \pm 0.8	42.5 \pm 0.7	45.6 \pm 0.8	45.8 \pm 1.1	51.8 \pm 1.1
GraphWave [176]			45.2 \pm 1.4	48.0 \pm 1.4	38.1 \pm 1.9	41.1 \pm 1.6	40.2 \pm 2.0	43.1 \pm 1.8
DRNE [177]			51.3 \pm 1.1	52.4 \pm 1.1	43.1 \pm 1.7	47.6 \pm 1.3	46.5 \pm 2.7	50.2 \pm 2.3
STANDARD GRAPH NEURAL NETWORKS								
GCN [39]	✓	✓	51.9 \pm 0.9	56.0 \pm 0.9	37.4 \pm 0.9	40.1 \pm 0.8	36.5 \pm 1.5	38.9 \pm 1.6
GAT [40]	✓	✓	52.7 \pm 1.0	53.5 \pm 0.9	31.5 \pm 1.0	34.3 \pm 1.0	37.3 \pm 1.6	37.9 \pm 1.6
GraphSAGE [41]	✓	✓	45.3 \pm 1.2	49.4 \pm 1.2	28.8 \pm 1.0	32.5 \pm 1.0	36.1 \pm 1.6	37.5 \pm 1.6
JKNet [42]	✓	✓	53.8 \pm 1.2	56.1 \pm 1.0	49.7 \pm 1.1	53.8 \pm 1.1	55.9 \pm 1.5	58.4 \pm 1.8
DGI [44]	✓		46.4 \pm 1.3	47.3 \pm 1.2	37.5 \pm 1.5	39.9 \pm 1.5	41.4 \pm 1.6	45.2 \pm 1.7
STRUCTURAL GRAPH NEURAL NETWORKS								
DemoNet [45]	✓	✓	58.6 \pm 1.2	58.8 \pm 1.1	40.4 \pm 1.3	46.2 \pm 1.2	46.1 \pm 1.4	48.9 \pm 1.5
Motif-CNN [46]	✓	✓	53.6 \pm 1.0	54.2 \pm 1.0	37.9 \pm 1.0	41.1 \pm 1.1	28.9 \pm 1.6	35.7 \pm 1.7
MCN [47]	✓	✓	54.8 \pm 1.4	54.9 \pm 1.3	36.8 \pm 1.2	39.6 \pm 1.5	42.9 \pm 1.6	43.6 \pm 1.4
MOTIF-REGULARIZED GRAPH NEURAL NETWORKS (InfoMotif)								
InfoMotif-GCN	✓	✓	59.5 \pm 0.9	62.9 \pm 0.7	53.5 \pm 0.6	56.9 \pm 0.6	56.6 \pm 1.2	60.7 \pm 1.2
InfoMotif-JKNet	✓	✓	61.8 \pm 1.6	64.3 \pm 1.2	53.1 \pm 1.2	56.9 \pm 0.6	62.7 \pm 1.8	67.9 \pm 1.5
InfoMotif-GAT	✓	✓	58.0 \pm 0.4	60.4 \pm 0.3	46.0 \pm 1.5	50.0 \pm 2.0	50.6 \pm 1.3	56.3 \pm 1.1

Table 3.5: Node classification results (% test accuracy) on dis-assortative air-traffic networks. **X** and **Y** denote the use of attributes and labels respectively. Structural embedding methods and structural GNNs typically outperform proximity-based models. INFOMOTIF JK-Net achieves significant performance gains of 4% to 14% across different datasets.

classification accuracy gains of 3% for all three base GNN variants, with the GCN model variant achieving best empirical results overall.

In air-traffic networks, structural embedding methods outperform their proximity-based counterparts. We also observe a similar trend for structural GNNs over standard message-passing GNNs. Class labels rely more on node structural roles than the labels of neighbors in air-traffic networks. JK-net outperforms other competing GNNs, signifying the importance of long-range dependencies in air-traffic networks. Our framework INFOMOTIF enables GNNs to learn structural roles agnostic to network proximity, and achieves significant gains of 10% on average across all datasets.

Training Ratio	Data		DBLP-A		DBLP-P		Movie	
	X	Y	10%	20%	10%	20%	10%	20%
PROXIMITY-BASED GRAPH EMBEDDING METHODS								
Node2Vec [18]			63.9 \pm 0.4	65.3 \pm 0.6	68.5 \pm 0.4	70.1 \pm 0.5	54.1 \pm 0.3	56.7 \pm 0.3
Motif2Vec [175]			62.7 \pm 0.7	64.5 \pm 0.6	68.9 \pm 0.9	71.3 \pm 1.1	55.0 \pm 0.3	57.4 \pm 0.5
STRUCTURAL GRAPH EMBEDDING METHODS								
Struct2Vec [24]			34.2 \pm 0.4	36.1 \pm 0.4	34.9 \pm 0.2	35.7 \pm 0.3	32.7 \pm 0.4	34.9 \pm 0.1
GraphWave [176]			34.8 \pm 0.5	37.0 \pm 0.6	35.6 \pm 0.2	36.3 \pm 0.3	33.5 \pm 0.4	36.0 \pm 0.4
DRNE [177]			33.9 \pm 0.3	36.5 \pm 0.3	35.1 \pm 0.5	35.5 \pm 0.4	31.0 \pm 0.2	35.1 \pm 0.2
STANDARD GRAPH NEURAL NETWORKS								
GCN [39]	✓	✓	65.3 \pm 1.1	69.6 \pm 0.9	71.3 \pm 0.7	73.4 \pm 0.8	55.7 \pm 1.0	57.3 \pm 0.7
GAT [40]	✓	✓	67.5 \pm 0.8	71.7 \pm 0.8	71.9 \pm 0.5	73.0 \pm 0.7	58.6 \pm 0.9	59.9 \pm 1.0
GraphSAGE [41]	✓	✓	65.3 \pm 0.7	69.0 \pm 0.6	70.9 \pm 0.8	72.7 \pm 0.8	55.6 \pm 0.4	56.4 \pm 0.8
JKNet [42]	✓	✓	69.6 \pm 1.0	73.2 \pm 1.2	69.8 \pm 1.1	72.0 \pm 1.3	58.3 \pm 0.7	60.5 \pm 0.9
DGI [44]	✓		64.7 \pm 0.5	68.5 \pm 0.7	41.9 \pm 0.8	61.1 \pm 0.6	38.6 \pm 0.9	40.4 \pm 0.9
STRUCTURAL GRAPH NEURAL NETWORKS								
DemoNet [45]	✓	✓	70.7 \pm 1.3	72.3 \pm 1.1	72.6 \pm 0.9	73.5 \pm 0.6	59.5 \pm 0.8	61.2 \pm 0.8
Motif-CNN [46]	✓	✓	66.4 \pm 1.1	70.1 \pm 1.3	71.5 \pm 0.8	72.2 \pm 0.6	54.3 \pm 0.3	56.9 \pm 0.5
MCN [47]	✓	✓	67.1 \pm 1.2	71.2 \pm 1.1	71.9 \pm 0.9	72.5 \pm 0.6	54.7 \pm 0.4	57.2 \pm 0.6
HETEROGENEOUS GRAPH NEURAL NETWORKS								
HAN [56]	✓	✓	68.2 \pm 1.0	72.0 \pm 1.3	73.1 \pm 0.9	74.0 \pm 0.7	60.7 \pm 1.1	62.1 \pm 0.8
MAGNN [57]	✓	✓	68.9 \pm 0.7	72.5 \pm 0.7	74.7 \pm 0.6	75.8 \pm 0.7	62.1 \pm 0.9	63.0 \pm 0.5
Meta-GNN [58]	✓	✓	71.3 \pm 1.2	73.9 \pm 1.4	74.6 \pm 0.6	75.8 \pm 0.6	61.7 \pm 0.5	63.7 \pm 0.7
MOTIF-REGULARIZED GRAPH NEURAL NETWORKS (InfoMotif)								
InfoMotif-GCN	✓	✓	73.7 \pm 1.2	77.4 \pm 1.1	78.8 \pm 0.4	79.0 \pm 0.7	64.7 \pm 0.8	65.0 \pm 1.1
InfoMotif-JKNet	✓	✓	75.6 \pm 0.9	79.9 \pm 0.8	75.5 \pm 1.0	76.3 \pm 1.2	60.7 \pm 0.6	62.0 \pm 0.6
InfoMotif-GAT	✓	✓	72.4 \pm 1.0	75.3 \pm 1.4	77.1 \pm 0.7	77.9 \pm 0.6	62.8 \pm 0.5	64.2 \pm 0.5

Table 3.6: Node classification results (% test accuracy) on heterogeneous graphs from bibliographic and movie networks, with schemas shown in figures 3.3(a) and 3.5(a). **X** and **Y** denote the use of node attributes and training labels respectively. Heterogeneous GNN models (MAGNN, Meta-GNN) typically outperform structural GNNs (DemoNet, Motif-CNN, MCN) and type-agnostic message-passing GNNs (GCN, GraphSAGE, GAT). Our framework INFOMOTIF (with typed network motif regularization) achieves consistent and significant performance gains of 5% on average across the three heterogeneous network datasets.

3.5.5 Heterogeneous Graphs (RQ₂)

In heterogeneous graphs, we train INFOMOTIF and other baselines that use motifs/meta-graphs using typed 3-node network motifs (shown in Figures 3.3(b) and 3.5(b)) that are defined based on the heterogeneous type schema (Figure 3.3(a) and 3.5(a)). Our experimental results comparing our framework INFOMOTIF with baselines on DBLP and Movie networks, are shown in Table 3.6.

We find that message-passing GNNs (such as GCN, GAT) generally outperform conventional embedding methods (such as node2vec). Structural embedding methods (*e.g.*, struc2vec) perform poorly; this reveals their inability to capture structural aspects relevant to heterogeneous graphs with rich type semantics. Heterogeneous GNNs such as MAGNN and Meta-GNN outperform homogeneous GNNs owing to their type-aware semantic neighbor aggregation via metapaths and metagraphs respectively. Our framework INFOMOTIF further learns type-aware attributed structural roles which results in significant performance gains of 5% on average over prior approaches.

Dataset		Cora	Citeseer	Pubmed
InfoMotif-GCN ($L_S + \lambda L_{MI}$)		87.4 \pm 0.4	78.5 \pm 0.5	88.3 \pm 0.2
w/o novelty weights ($\beta_v = 1$ in eq. (3.9))		86.4 \pm 0.5	77.6 \pm 0.5	87.8 \pm 0.3
w/o task weights ($\alpha_{vt} = 1$ in eq. (3.7))		84.6 \pm 0.4	77.3 \pm 0.4	87.3 \pm 0.2
w/o novelty and task weights		84.0 \pm 0.5	76.4 \pm 0.6	87.3 \pm 0.2
Base model GCN (L_B)		82.0 \pm 0.4	76.6 \pm 0.3	86.1 \pm 0.5

Table 3.7: Ablation study results with 40% training ratio on citation networks. The novelty and task weighting strategies improve classification accuracies by 2% on average.

3.5.6 Model Ablation Study (RQ₃)

We present an ablation study on two citation networks Cora and Citeseer, to analyze the importance of major components in INFOMOTIF (Table 3.7). In our experiments, we choose GCN as the base GNN model due to its consistently high performance.

- **Remove novelty-driven sample weighting.** We set the novelty weight $\beta_v = 1$ (Equation (3.9)) for each labeled node $v \in \mathcal{V}_L$ to test the importance of addressing motif occurrence skew. We observe consistent 1% gains due to our novelty-driven sample weighting.

- **Remove task-driven motif weighting.** We remove the node-sensitive motif weights from the motif regularization loss (Equation (3.7)) by setting $\alpha_{vt} = 1$ for every node-motif pair. Contextually weighting the different motif regularizers at a node-level granularity results in 2% average accuracy gains across both datasets.
- **Remove both novelty and task driven weighting.** This variant applies a uniform motif regularization over all nodes without distinguishing the node-sensitive relevance of each motif; this significantly degrades classification accuracy.

3.5.7 Qualitative Analysis (RQ₄)

We qualitatively examine the source of INFOMOTIF’s gains over the base GNN (GCN due to its consistent performance). by analyzing *node degree*, *label sparsity* and *attribute diversity* in local node neighborhoods, on the Cora and Citeseer networks.

Node Degree

To evaluate performance variance with node degree, we divide the set of test nodes into bins based on four degree ranges. Figure 3.6 depicts the variation in classification accuracy for GCN and INFOMOTIF-GCN across degree segments, on Cora and Citeseer datasets.

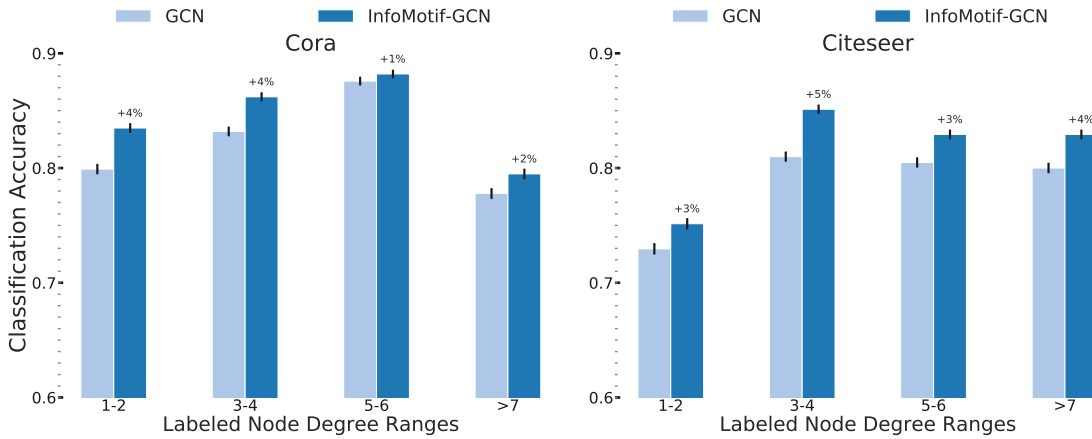


Figure 3.6: *Classification accuracy with respect to node degree. INFOMOTIF has consistent gains across all segments with higher gains for low-to-medium degree nodes (quartiles Q1 & Q2).*

INFOMOTIF has consistent performance improvements over GCN across all degree segments, with notably higher gains for low-to-medium degree nodes (quartiles Q1 and Q2).

Learning structural roles through self-supervised motif regularization is beneficial for nodes with limited local structural information.

Label Sparsity

We define the *label fraction* for a node as the fraction of labeled training nodes in its 2-hop neighborhood, *i.e.*, a node exhibits label sparsity if it has very few or no labeled training nodes within its 2-hop aggregation range. We separate the set of test nodes into four quartiles by their label fraction. Figure 3.7 depicts classification results for GCN and INFOMOTIF-GCN under each quartile (Q1 has nodes with small label fractions).

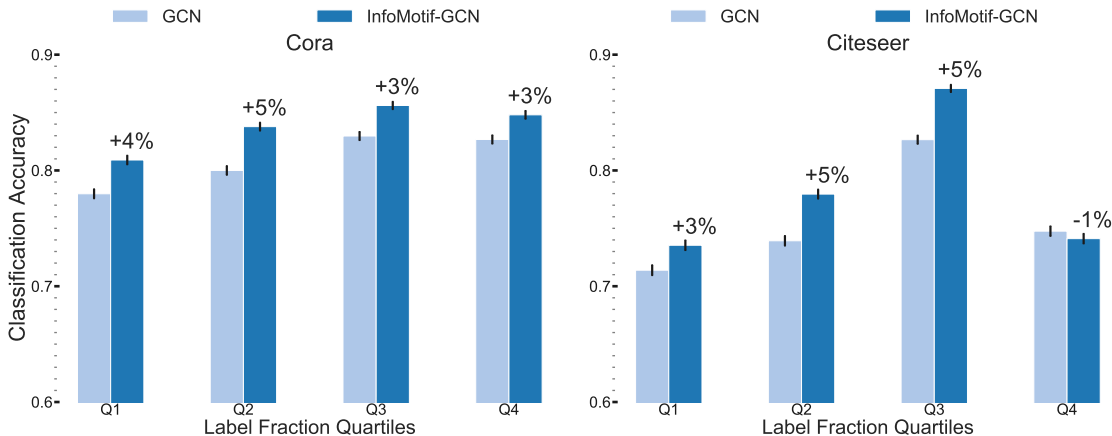


Figure 3.7: Classification accuracy over label fraction quartiles. (Q1: smaller label fraction). INFOMOTIF has larger gains over GCN in Q1 & Q2 (nodes that exhibit label sparsity)

INFOMOTIF has stronger performance gains over GCN for nodes with smaller label fractions (quartiles Q1 and Q2); this empirically validates the efficacy of our motif-based regularization framework in addressing the key limitation of k -hop localization in message-passing GNNs (Section 3.3.1), *i.e.*, INFOMOTIF benefits nodes with very few or no labeled nodes within their k -hop aggregation ranges, thus effectively addressing *label sparsity* challenges.

Attribute Diversity

We measure the local *attribute diversity* of a node by the mean pair-wise attribute dissimilarity (computed by cosine distance) of itself with other nodes in its 2-hop neighborhood, *i.e.*, a node that exhibits strong homophily with its neighbors has low attribute diversity. We report classification results across attribute diversity quartiles in Figure 3.8.

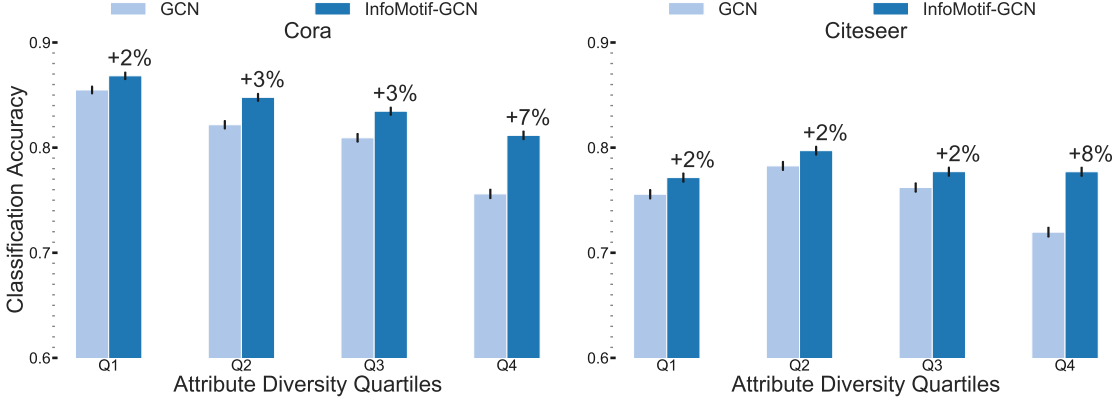


Figure 3.8: Classification accuracy across attribute diversity quartiles. (Q_4 : high attribute diversity). INFOMOTIF has stronger gains in Q_3 & Q_4 (nodes with diverse attributed neighborhoods).

Nodes with diverse attributed neighborhoods are typically harder examples for classification. Regularizing GNNs to learn attributed structures via motif occurrences can accurately classify diverse nodes, as evidenced by the higher relative gains of INFOMOTIF for diverse nodes (quartiles Q_3 and Q_4).

3.5.8 Parameter Sensitivity

We examine the effect of hyper-parameter Q that controls the number of motif instances sampled per node to train our motif-based discriminators (Equation (3.4)). Figure 3.9 shows variation in accuracies of our three GNN variants with the number of sampled instances (5 to 30), on Cora and Citeseer networks.

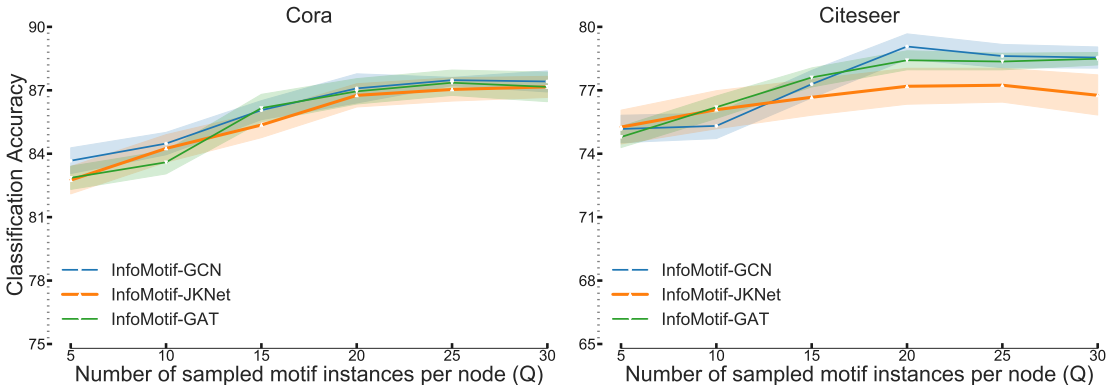


Figure 3.9: Classification accuracy increases slowly with the number of sampled motif instances and stabilizes around 15 to 20. Variance bands indicate 95% confidence intervals over 10 runs.

Performance of all GNN variants stabilize with 20 instances across both datasets. Since the complexity of our framework scales linearly with Q , we fix $Q = 20$ across datasets to provide an effective trade-off between compute-cost and performance

3.5.9 Efficiency Analysis

We empirically evaluate the added complexity of INFOMOTIF on two GNN models, GCN and GAT. We report the time per epoch on synthetically generated Barabasi-Albert networks [1] with 5000 nodes and increasing link density (Figure 3.10).

INFOMOTIF adds a small fraction of the base GNN runtime, and the added complexity scales linearly with the number of nodes, as evidenced by its nearly constant runtime gap over increasing link density (Figure 3.10). Furthermore, our GCN variant INFOMOTIF-GCN is significantly more efficient than GAT.

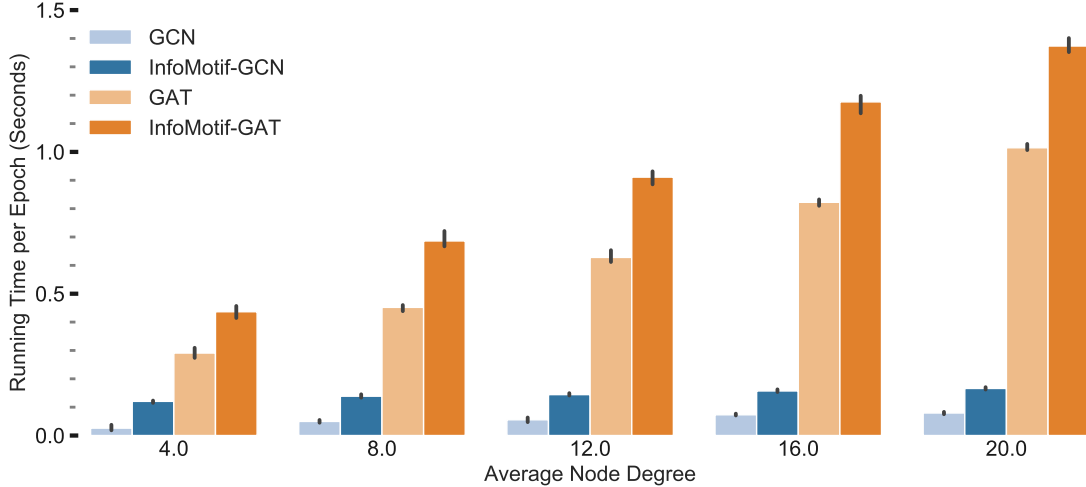


Figure 3.10: Runtime comparison of INFOMOTIF variants with its base GNNs. INFOMOTIF has minimal computational overheads; notice the nearly constant runtime gap with increasing node degree.

3.6 DISCUSSION

Our framework is designed to address the limitations of *oversmoothing* and *localization* in prior message-passing GNNs.

INFOMOTIF is orthogonal to advances in GNN architectures that improve the *structural distinguishability* of node representations through carefully designed neighborhood aggregators. In contrast, we enhance the structural resolution of node representations by regularizing

base GNNs through self-supervised learning objectives designed to capture connectivity in higher-order motif structures. By training contrastive discriminators to discover attribute correlations among motif instances across the entire graph, our approach learns generalizable *global* roles in addition to modeling local connectivity patterns. The enhanced quality of our learned node representations is further evidenced by our superior empirical performance for nodes with diverse attributed neighborhoods (Section 3.5.7).

INFOMOTIF addresses the challenge of *localization* by regularizing base GNNs to learn attributed structural roles through self-supervised training objectives. Specifically, our approach statistically relates distant nodes in the graph with co-varying attributed structures, to effectively overcome label sparsity in local neighborhoods (Section 3.5.7). Instead of adopting deeper GNNs that directly expand neighborhood aggregation ranges, we demonstrate the effectiveness of regularizing shallow base GNNs to learn attributed structural roles. Compared to deeper GNNs that scale poorly with neighborhood sizes, our regularization strategy enables efficient model inference.

In our work, we choose network motif structures as the central basis to formulate structural roles. In contrast to alternative approaches to quantify structural similarity based on coarse properties like degree sequences [24] or rigid notions of structural equivalence [22], network motifs are fundamental higher-order connectivity structures that enable flexible generalization to complex heterogeneous graphs with rich semantics. We empirically demonstrate the utilities of untyped motifs in homogeneous graphs and typed motifs in heterogeneous graphs.

Our key modeling hypothesis is the importance of attribute co-variance in local structures towards the learning application (*e.g.*, classification in social networks). Our substantial gains on two diverse classes of datasets indicates broad applicability for INFOMOTIF across graphs with varied structural characteristics. However, the performance gains may diminish in application scenarios (*e.g.*, learning in regular mesh graphs) where modeling such co-variance is not beneficial or even necessary.

3.7 RELATED WORK

Our work is related to semi-supervised learning approaches over *homogeneous* and *heterogeneous* graphs, and recent advances in the paradigm of *self-supervised* learning.

Homogeneous Graphs: Semi-supervised learning over graphs is a well-studied problem, where the goal is to classify nodes in a graph given a small set of labeled examples. The most popular label spreading [32] techniques propagate labels through linked nodes in the graph based on smoothness assumptions. Graph Neural Networks (GNNs) generalize label spreading through localized message passing over feature-rich node neighborhoods and

have achieved state-of-the-art results in several benchmarks [39]. GNNs learn node representations by recursively aggregating features from local neighborhoods in an end-to-end manner, with diverse applications, including information diffusion prediction [179], social recommendation [180], and community question answering [181]. Graph Convolutional Networks (GCNs) [39] learn degree-weighted aggregators, which can be interpreted as a special form of Laplacian smoothing [155]. Many models generalize GCN with a wide range of neighborhood aggregators, *e.g.*, self-attentions [40, 68], mean and max pooling functions [41], etc. However, all these models learn node representations that inherently overfit to the k -hop neighborhood around each node.

There are two broad categories of prior graph representation learning approaches that aim to overcome the key limitations of *oversmoothing* and *localization* in GNNs: *non-local GNNs* that capture contributions from distant nodes in the graph; and *structural role learning* techniques that enhance the structural distinguishability of the learned node representations.

Non-local methods expand the propagation range of GNNs to aggregate node representations of differing localities, *e.g.*, JKNet [42] uses skip-connections to vary the influence radius per node, PGNN [43] captures global network positions via shortest-paths, and DGI [44] maximizes MI between node representations and a summary representation of the entire graph. However, they either operate on a local scale [182], or learn coarse structural properties, which limits their ability to capture features from distant yet structurally similar nodes.

Role-aware models embed structurally similar nodes close in the latent space, independent of network position [22, 23]. A few approaches [177] employ strict definitions of structural equivalence to embed nodes with identical local structures to the same point in the latent space, while others utilize structural node features (*e.g.*, node degrees, motif count statistics) to extend classical proximity-preserving embedding methods, *e.g.*, feature-based matrix factorization [25] and random walk methods [24]. Notably, a few methods design structural GCNs via motif adjacency matrices [46, 47, 58]. However, all these methods model structural roles without considering node attributes. A related direction is higher-order network representation learning that models proximity via network motifs [175]. However, such representations are highly localized and cannot identify structurally similar nodes independent of network proximity. In contrast, we regularize GNNs to learn attributed structural roles based on the co-variance of attributes in motifs, thus simultaneously enhancing the distinguishability of node representations and identifying correspondences between distant nodes.

Heterogeneous Graphs: Representation learning techniques over heterogeneous graphs primarily focus on preserving structural information indicated by the type semantics in meta-path or meta-graph structures. A few popular approaches include node representation learning by capturing proximities between node pairs connected via meta-graphs [49, 50] and

meta-path guided random walks [51, 52]. Recently, graph neural networks have been generalized to heterogeneous graphs through message passing aggregation over local neighborhoods induced via specific node types [53, 54], meta-paths [55, 56, 57] and meta-graphs [58]. While these advances effectively incorporate rich heterogeneous semantics into message-passing GNNs, the key limitation due to localization remains. To our knowledge, structure role learning in heterogeneous graphs is unexplored and ours is the first to examine structural role learning in GNNs with rich type semantics and attributes.

Self-supervised Learning: The emerging paradigm of self-supervised learning [144] aims to alleviate the need for large volumes of labeled examples by extracting supervision signals from the intrinsic structure of the raw data. For instance, auxiliary supervision signals for images are created by rotating, cropping and colorizing images, followed by new training objectives to facilitate representation learning [183]. One empirically effective strategy is mutual information maximization [184] to maximize agreement across different views of the data. A few recent advances extend self-supervised learning to graph representation learning by exploiting structural properties such as node degree, proximity [185], and attributes [186], for model pre-training [187, 188, 189]. In our work, we design self-supervised learning objectives to regularize graph neural networks for node classification by learning attribute correlations in higher-order connectivity patterns (typed and untyped motif structures).

3.8 CONCLUSION

This chapter presents a new class of motif-regularized GNNs with an architecture-agnostic framework INFOMOTIF for semi-supervised learning on graphs. To overcome limitations of prior GNNs due to localized message passing, we introduce attributed structural roles to regularize GNNs by learning statistical dependencies between structurally similar nodes with co-varying attributes, independent of network proximity. INFOMOTIF maximizes motif-based mutual information, and dynamically prioritizes the significance of different motifs. Our experiments on nine real-world datasets spanning homogeneous and heterogeneous networks, show substantial consistent gains for INFOMOTIF over state-of-the-art methods.

In this chapter, we have explored a transductive learning scenario of semi-supervised learning over graph-structured entity interactions given sparse training labels. We leveraged higher-order connectivity structures (network motifs) between nodes to compensate for the lack of sufficient labeled information in the local neighborhoods. In the next chapter, we examine another central transductive scenario of collaborative filtering in the bipartite user-item interaction setting. Our key focus is to enable accurate personalized recommendations for data-poor tail entities (such as long-tail items) with severe interaction sparsity.

CHAPTER 4: PROTOCF: FEW-SHOT COLLABORATIVE FILTERING VIA META-TRANSFER

4.1 INTRODUCTION

Neural Collaborative Filtering (NCF) methods have recently enabled substantial advances in modern recommender systems that are critical to diverse e-commerce applications. Collaborative Filtering (CF) methods generate personalized recommendations by learning patterns from historical user-item interactions. However, a close examination of neural recommenders’ performance reveals a *paradox*: while the overall recommendation accuracy is high, accuracy levels are poor for most items in the inventory. A majority of recommendations are biased towards popular items [2], while ignoring *long-tail* items in under-represented categories. *Popularity bias* restricts personalization and impedes suppliers of long-tail items, who struggle to attract consumers given the low exposure. Targeting long-tail items can enhance recommendation diversity and bring relatively larger marginal profits, compared to popular items with competitive markets. The increasing impact of recommendations (*e.g.*, 80% of Netflix activity is a result of recommendations [190]), also raises *ethical* questions: can items that are never recommended by a system be considered an instance of *discrimination* [191]? Thus, we focus on learning robust models for long-tail item recommendations.

Empirical evidence of long-tail challenges: We highlight two key observations on prior neural recommendation models [88, 89, 90]:

- Performance gains are *skewed* towards popular items with abundant historical interactions. Figure 4.1 depicts Item Recall@ K (fraction of correctly ranked items within top- K) of three neural recommenders for different item-groups ordered by increasing item interaction count. Model performance is considerably lower for long-tail items (low popularity), indicating a clear *popularity bias*.
- Prior recommenders lack the *resolution power* to accurately rank relevant items within the long-tail. To empirically illustrate this observation, we split the item inventory into two equal-sized sets by interaction count (popular and long-tail), and evaluate personalized ranking performance independently within each set. Table 4.1 shows a big performance gap between ranking within the popular and long-tail item sets, which reflects on the poor quality of their long-tail item representations.

Prior efforts towards the long-tail have been two-fold. First, regularizing an existing recommender via inter-item associations (derived from item co-occurrences) [9, 192, 193, 194]

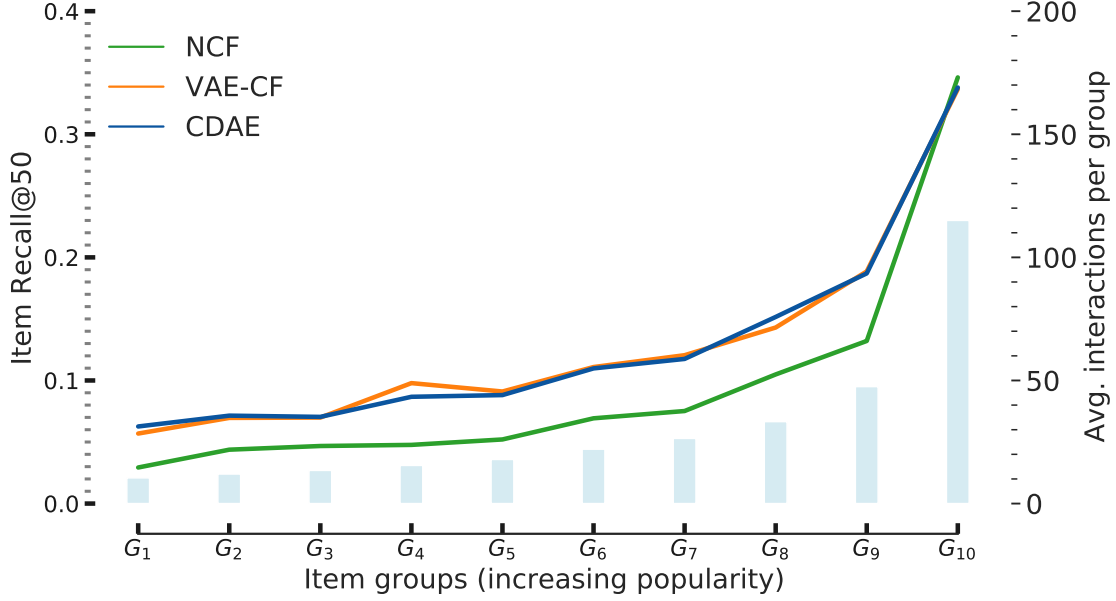


Figure 4.1: *Item Recall@50 of three popular neural recommendation models across different item-groups (increasing popularity) in Epinions. Neural recommenders overfit to popular items with considerably lower performance for long-tail items.*

or distributional priors over the latent space [90, 98]. While regularization strategies can partially alleviate overfitting, they typically involve static hypotheses that may be counter-productive for massive item inventories with a diverse collection of long-tail items. Second, incorporating external side information [103], *e.g.*, item attributes [195], social networks [106], and knowledge graphs [107] to overcome interaction sparsity. Note that exploiting side information is typically application-dependent, thus lacking generalizability to diverse scenarios. In contrast, we exploit the capabilities of any neural base recommender for long-tail item recommendation without any requirement of side information.

Item Subset	Top 50% Items		Bottom 50% Items	
	N@50	R@50	N@50	R@50
NCF [88]	0.0906	0.1874	0.0352	0.0973
VAE-CF [90]	0.1055	0.2106	0.0457	0.1125
CDAE [89]	0.1050	0.2102	0.0471	0.1149

Table 4.1: *Recommendation performance within top-50% head and bottom-50% tail items (ordered by item popularity) on Epinions. R@50 and N@50 denote Recall@50 and NDCG@50 metrics respectively. Compared to head items, we observe poor item ranking resolution within the long-tail.*

Learning a latent space tailored to long-tail items poses two key challenges: first, *sparsity and heterogeneity*: although tail items have sparse interactions, they belong to diverse item categories, *e.g.*, Figure 4.2 shows long-tailed distributions along different dimensions described by item attributes, location and category of Yelp businesses. Thus, we need sufficient resolution power to learn discriminative representations for tail items, while being careful to avoid overfitting. Second, *distribution mismatch* between head items (substantial interactions) and tail items (sparse interactions). Specifically, neural recommenders typically sample user-item pairs from the overall interaction distribution (biased to head items) for model training; this results in overfitting and degenerate representations for tail items.

Present Work: To overcome item interaction sparsity, we *extract, relate, and transfer* the knowledge learned by a neural base recommender over *head* items, to learn robust and discriminative tail item representations. We eliminate the distribution inconsistency between head and tail items with an *episodic few-shot learning* setup [151] to *simulate* the distribution of tail items during model training by *sub-sampling* interactions from data-rich head items.

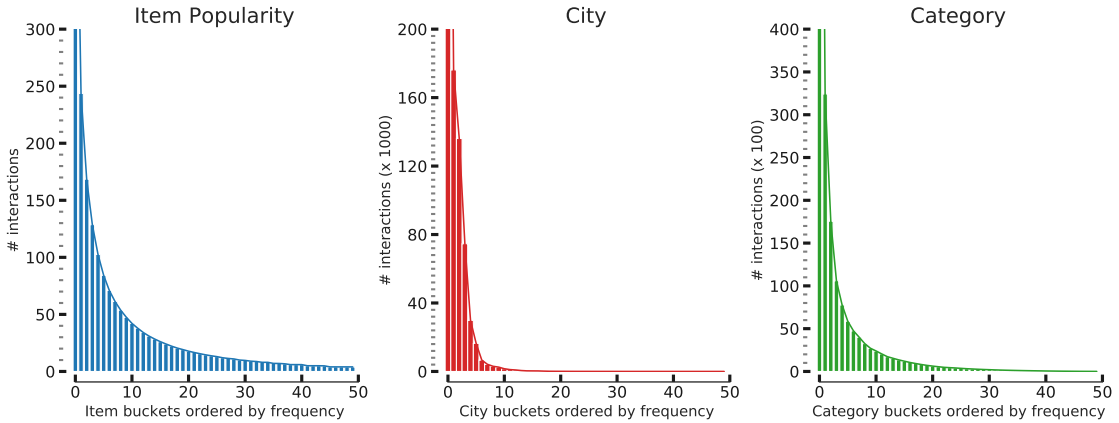


Figure 4.2: Multiple long-tailed distributions along different item dimensions. (a) shows interaction frequency distribution over unique items. (b) and (c) show frequency distributions over unique cities and categories of Yelp businesses.

In this work, we present a novel *metric-based* few-shot learning framework PROTOCF for long-tail item recommendation. First, we pretrain a *base recommender* to extract preferences over head items and item-item relationships. Then, we design a *few-shot recommender* that learns a *shared metric space* of users and items by extracting *meta-knowledge* across a collection of meta-training tasks designed to *mimic* long-tail item recommendation. PROTOCF *learns-to-compose* a representative *prototype* for each item from a small set of user interactions and recommends relevant items by finding the nearest item prototypes to each user. To transfer knowledge from the base recommender, we introduce a *knowledge distillation* strategy to distill the discovered *item-to-item* relationships into a *compact* set of *group*

embeddings; we compose *discriminative* item prototypes via learnable mixtures of group embeddings. We summarize our key contributions below:

Few-shot Item Recommendation: To our knowledge, ours is the first to formulate long-tail item recommendation as *learning-to-recommend* items with few interactions. We eliminate the distribution mismatch between head and tail items via episodic few-shot learning. Our problem formulation applies to diverse recommendation scenarios without requiring any auxiliary side information.

Discriminative Prototype Learning: We learn to compose *discriminative prototypes* for tail item *only* from their *sparse* user interactions. Unlike prior gradient based meta-learning for cold-start recommendation scenarios [110, 111, 112], we learn a metric space where the item prototypes directly cluster the interacted users in a *metric space*, which further eliminates expensive online adaptation costs.

Architecture-agnostic Knowledge Transfer: We enhance item prototypes by knowledge transfer from neural base recommenders. In contrast to layer transfer or adaptation methods [105], our *knowledge distillation* strategy extracts a compact representation of the item relationships discovered by arbitrary base recommenders. Significantly, PROTOCF is complementary to architectural advances in neural base recommenders and enables flexible adaptation to items in the long tail.

We instantiate PROTOCF by transferring meta-knowledge from three base recommenders. Our experiments show that PROTOCF outperforms state-of-the-art baselines (by 5% Recall@50) in *overall* recommendation, with notably significant *few-shot* gains (of 60-80% Recall@50) on tail items with less than 20 training interactions.

We organize the rest of the chapter as follows. In Section 4.2, we formally define the problem of long-tail item recommendation. In Section 4.4, we first outline the components of a neural base recommender, and then introduce our proposed framework PROTOCF for few-shot item recommendation. We then present our experimental results in Section 4.5, and finally conclude in Section 4.6.

4.2 PROBLEM DEFINITION

We consider the implicit feedback setting (only clicks, no explicit ratings) with a user set $\mathcal{U} = \{u_1, \dots, u_U\}$, an item set $\mathcal{I} = \{i_1, \dots, i_I\}$, and a binary $U \times I$ user-item interaction matrix \mathbf{X} . Let N_i denote the set of all users who have interacted with item $i \in \mathcal{I}$. Prior neural recommenders learn a scoring function $f(i | u, \mathbf{X}), i \in \mathcal{I}$ personalized to each user $u \in \mathcal{U}$ for

Symbol	Description
$p_B(i, j)$	proximity induced by \mathbf{R}_B for items $i, j \in \mathcal{I}$
$p(\mathbb{T})$	meta-training task distribution over items \mathcal{I}
\mathcal{T}	meta-training task sampled from $p(\mathbb{T})$
$\mathcal{I}_{\mathcal{T}, N}$	set of N items sampled from \mathcal{I} for task \mathcal{T}
\mathcal{S}_i	support set for item $i \in \mathcal{I}_{\mathcal{T}, N}$ in task \mathcal{T}
\mathcal{Q}_i	query set for item $i \in \mathcal{I}_{\mathcal{T}, N}$ in task \mathcal{T}
$G_U(\cdot \mid \theta)$	user encoder in few-shot recommender \mathbf{R}_F
\mathcal{Z}_M	set of M group embeddings $\{\mathbf{z}_m \in \mathbb{R}^D\}_{m=1}^M$
sim_m	similarity metric for meta-recommender \mathbf{R}_F
\mathbf{p}_i	initial item prototype for item $i \in \mathcal{I}$
\mathbf{g}_i	group-enhanced item prototype for item $i \in \mathcal{I}$
\mathbf{e}_i	final gated item prototype item for item $i \in \mathcal{I}$

Table 4.2: *Notations*

item ranking over the item set \mathcal{I} . From Table 4.1, we find their performance for tail items (bottom 50%) to be considerably inferior to head items (upper 50%). Thus, our focus is to develop personalized recommenders tailored to the *long-tail* items (with sparse interactions), while ensuring reasonable performance over the entire item set \mathcal{I} .

Problem 4.1 (Long-Tail Item Recommendation). Given user-item interaction matrix \mathbf{X} , learn a scoring function $f_T(i \mid u, \mathbf{X}), i \in \mathcal{I}$ to generate a ranked list of items personalized to each user $u \in \mathcal{U}$ that improves recommendation quality on the long-tail items without compromising overall model performance on the entire item set \mathcal{I} .

4.3 RELATED WORK

We briefly review a few related lines of prior work on neural collaborative filtering and long-tail item recommendations with interaction sparsity.

Neural Collaborative Filtering: The core paradigm of latent-factor CF models is to parameterize users and items with latent representations learned from historical user-item interactions. Recent neural recommenders enhance the representational capacity of CF models via non-linear latent representations [89, 90], neural interaction modeling [88, 94], and graph-based representation learning [91]. While these neural recommenders learn expressive models to significantly outperform conventional CF approaches, sparsity concerns owing to long-tail items remain a critical challenge.

Sparsity-aware Recommendation: Clustering is one popular way to address interaction sparsity by modeling group-level purchasing behaviors; early methods generate recommendations for tail items at the granularity of item clusters [95], *e.g.*, cluster-based smoothing [196], user-item co-clustering [96] and joint clustering and collaborative filtering [97]. However, clustering in the presence of distributional skew can lead to uninformative results [197] and degrades the extent of personalization.

Recent techniques leverage *item-item co-occurrence* statistics and *distributional priors* to regularize latent-factor CF methods. One common approach is *regularization* of recommendation models, *e.g.*, factorization of an item co-occurrence matrix that shares latent factors with a CF model [193, 194]. Variational auto-encoders (VAEs) [90, 98, 179] employ multivariate Gaussian priors as regularizers on the representational space to handle sparsity. Another strategy to alleviate sparsity is *data augmentation* for items (or users) in the tail via *adversarial* regularization [9] or rating generation [99, 198] techniques. However, regularization techniques typically impose static hypotheses with restrictive assumptions, while adversarial learning is computationally expensive and does not scale to massive item inventories.

Cold-start recommendation is a related problem that targets *new* users or *new* items with *no* historical interactions. In such a scenario, models rely on auxiliary information, *e.g.*, user profiles [110], item content [195], social connections [106, 171, 199], and knowledge graphs [107]. An effective strategy is randomized feature dropouts to enable generalization to missing inputs [108, 109]. In contrast, we focus on *few-shot* recommendations over long-tail items with very few interactions, *without* any auxiliary information.

Few-shot Learning: this learning paradigm involves designing models capable of learning new tasks rapidly given a limited number of training examples. Here, the *meta-learning* (learning to learn) paradigm has achieved considerable success in several domains including computer vision [100, 101], natural language [200], and data mining [110, 116].

A few recent methods adapt meta-learning to cold-start recommendation [201]. Prior meta-learning work have examined gradient-based parameter [110, 111, 112, 113, 114] and hyper-parameter initialization [115], and layer sharing with user-specific (or item) parameter adaptation [116, 117, 118]. However, these methods mostly address the cold-start (zero-shot) scenario with side information, thus inapplicable to our general setting of few-shot recommendation given limited interactions.

Our work is conceptually related to *metric-learning* approaches [100, 101, 202] for few-shot learning, which *learn-to-learn* a metric space that generalizes to new tasks without any need for adaptation. A few recent explorations address task heterogeneity [203] in few-shot classification settings, by designing architectures for explicit knowledge transfer from data-

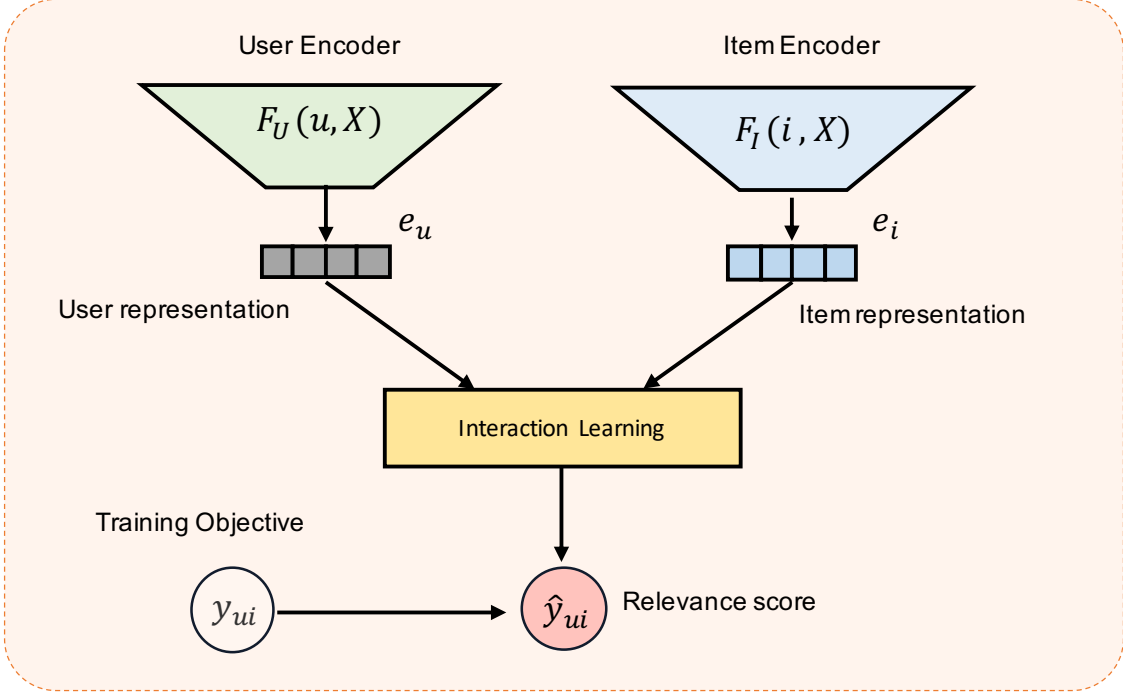


Figure 4.3: Outline of the different components of a neural base recommender based on collaborative filtering.

rich classes [204, 205, 206] to construct few-shot classifiers. However, such techniques are suited to learning classifiers from a *limited* number of classes, hence cannot scale to handle *massive* item inventories in recommendation applications. To our knowledge, ours is the *first* investigation of metric-based few-shot learning for long-tail item recommendations.

4.4 PROTOCF FRAMEWORK

Learning informative representations for tail items poses the key challenge of *interaction sparsity*, *i.e.*, each tail item only has a few observed interactions. Without access to side information (*e.g.*, item content or contextual attributes), we critically remark that prior neural recommenders [89, 90] learn informative representations for head items with abundant training interactions. Thus, to learn discriminative tail item representations, we propose a novel few-shot learning framework PROTOCF with two key steps:

First, we train a *base recommender* \mathbf{R}_B to learn high-quality user representations that mainly capture preferences over head items and infer item-item relationships (Section 4.4.1). Then, we present a *few-shot recommender* \mathbf{R}_F that *extracts and transfers* knowledge from the base recommender \mathbf{R}_B to *learn-to-recommend* few-shot items (Sections 4.4.2, 4.4.3).

4.4.1 Neural Base Recommender

In this section, we outline the architecture of a differentiable base recommender \mathbf{R}_B to learn a ranking $f_\phi(i | u), i \in \mathcal{I}$ (with parameters ϕ) over the interactions \mathbf{X} . Neural CF models [88, 89, 90] typically learn latent representations for users and items, followed by an interaction function and learning objective for model training.

User and Item Representations

Latent-factor CF methods adopt a variety of representation learning strategies, including matrix factorization [85, 87], autoencoders [89, 90] and graph neural networks [91]. We define preference encoders $F_U(\cdot | \phi)$ and $F_I(\cdot | \phi)$ to learn user $\mathbf{h}_u \in \mathbb{R}^D$ and item $\mathbf{h}_i \in \mathbb{R}^D$ embeddings for user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, which can be described by:

$$\mathbf{h}_u = F_U(u, \mathbf{X} | \phi) \quad \mathbf{h}_i = F_I(i, \mathbf{X} | \phi) \quad (4.1)$$

Training Objective

We define a user-item interaction function $F_{\text{INT}}(\cdot | \phi)$ to compute a prediction score $\hat{y}_b(u, i)$ that indicates the relevance of item i to user u . The training objective L_B of the base recommender \mathbf{R}_B is given by:

$$\hat{y}_b(u, i) = F_{\text{INT}}(\mathbf{h}_u, \mathbf{h}_i | \phi) \quad L_B = l(\hat{y}_b(u, i), y_{ui}) \quad (4.2)$$

where $y_{ui} = 1$ for observed (u, i) pairs and $l(\cdot)$ is a pairwise [87] or pointwise [88] loss function. over the user-item interactions. The interaction function F_{INT} measures user-item relevance and is typically modeled using an inner product [87, 89, 90, 91].

We train the base recommender \mathbf{R}_B using cosine similarity since normalized representations generalize better to few-shot settings [207] (compared to unnormalized inner products), and facilitate unified recommendation of head and tail items during model inference. We denote the *item-item proximity* $\text{sim}_b(\cdot)$ in the latent space of \mathbf{R}_B by:

$$p_B(i, j) \propto \text{sim}_b(\mathbf{h}_i, \mathbf{h}_j) = \cos(\mathbf{h}_i, \mathbf{h}_j) \quad i, j \in \mathcal{I} \quad (4.3)$$

We pre-train \mathbf{R}_B to extract knowledge of user preferences over head items via encoder $F_U(\cdot)$ and item-item proximities among different items via $p_B(i, j)$. Below, we present our proposed framework PROTOCF that transfers knowledge from the *data-rich* head to the *data-poor* tail to enable robust few-shot recommendations.

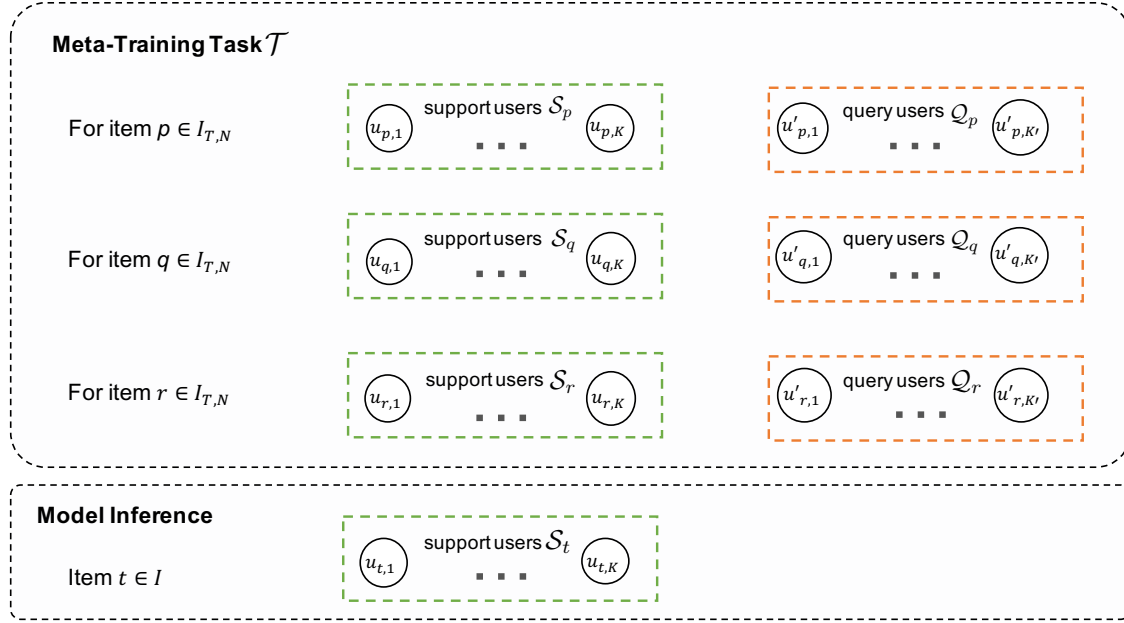


Figure 4.4: Episodic few-shot learning with meta-training task \mathcal{T} and item embedding inference at meta-testing.

4.4.2 Few-shot Item Recommendation

In this section, we formulate long-tail item recommendation as *few-shot item representation learning* for items given a small *support set* of upto K interacted users for each tail item (typically $K \approx 5$ to 20 interactions).

Few-shot Task Formulation

Our framework is grounded on *episodic learning* [100] with a collection of *meta-training* tasks or episodes. One approach [101, 207] is to construct meta-training tasks from the interactions of data-rich head items; however, excluding the diverse collection of tail items may impede generalization. Thus, our meta-training tasks operate on the entire item set \mathcal{I}_T .

Each meta-training task \mathcal{T} is a personalized ranking problem over a subset of N items $\mathcal{I}_{T,N}$ randomly sampled from \mathcal{I} . We *simulate the interaction distribution of few-shot items during meta-training* by sampling K training interactions (from N_i) for each item $i \in \mathcal{I}_{T,N}$ in task \mathcal{T} . During inference, we generate *few-shot* item recommendations from samples of upto K training interactions per item. By ensuring consistency between meta-training and inference, we bridge the *distribution mismatch* between head and tail items. The meta-knowledge extracted across diverse meta-training tasks benefits tail items with sparse interactions.

The episodic few-shot training process operates over a collection of meta-training tasks

$\{\mathcal{T}_1, \mathcal{T}_2, \dots\}$ sampled from a task distribution $p(\mathbb{T})$ over the item set \mathcal{I} . Specifically, a K -shot, N -item meta-training task \mathcal{T} sampled from $p(\mathbb{T})$ consists of support \mathcal{S} and query \mathcal{Q} user sets over N items $\mathcal{I}_{\mathcal{T},N} \subset \mathcal{I}$ (analogous to the usual sense of training and testing sets respectively). Each meta-training task $\mathcal{T} \sim p(\mathbb{T})$ is defined by:

$$\begin{aligned} \mathcal{T} &= \{\mathcal{I}_{\mathcal{T},N}, \mathcal{S}, \mathcal{Q}\} & \mathcal{I}_{\mathcal{T},N} &\subset \mathcal{I} & (4.4) \\ \mathcal{S} &= \{\mathcal{S}_i : i \in \mathcal{I}_{\mathcal{T},N}\} & \mathcal{S}_i &= \{u_{i,1}, \dots, u_{i,K}\} & u_{i,k} \in N_i \\ \mathcal{Q} &= \{\mathcal{Q}_i : i \in \mathcal{I}_{\mathcal{T},N}\} & \mathcal{Q}_i &= \{u'_{i,1}, \dots, u'_{i,K'}\} & u'_{i,k'} \in N_i \end{aligned}$$

where the support set \mathcal{S}_i contains K interacted users sampled for each item $i \in \mathcal{I}_{\mathcal{T},N}$ and the corresponding query set \mathcal{Q}_i includes K' interacted users sampled for each of the N items.

We learn a *few-shot recommender* \mathbf{R}_F that takes as input the support users \mathcal{S} to *learn-to-compose* representations for items $i \in \mathcal{I}_{\mathcal{T},N}$ in task \mathcal{T} . The few-shot recommender \mathbf{R}_F is trained by optimizing a learning objective designed to match the item recommendations generated by \mathbf{R}_F for its query users \mathcal{Q} with their corresponding ground-truth interactions over the item $\mathcal{I}_{\mathcal{T},N}$ in each meta-training task \mathcal{T} .

Initial Item Prototype

We learn a shared *metric space* of users and items to synthesize a representation for each item $i \in \mathcal{I}_{\mathcal{T},N}$ based on its support set \mathcal{S}_i (with K interactions) in task \mathcal{T} . The metric space (with similarity $\text{sim}_m(\cdot)$) compactly clusters the interacted users \mathcal{S}_i of each item $i \in \mathcal{I}_{\mathcal{T},N}$ around a *prototype* representation $\mathbf{p}_i \in \mathbb{R}^D$ [101].

We first define a user preference encoder $G_U(\cdot \mid \theta)$ that maps each user $u \in \mathcal{U}$ into a latent embedding space. To directly transfer knowledge from the base recommender \mathbf{R}_B , the few-shot user encoder $G_U(\cdot \mid \theta)$ has model parameters initialized from its pre-trained encoder $F_U(\cdot \mid \phi)$ in the base recommender \mathbf{R}_B (Equation 4.1), but is parameterized with learnable parameters θ . The prototype \mathbf{p}_i for item $i \in \mathcal{I}_{\mathcal{T},N}$ is computed as the mean vector of the embedded support user set \mathcal{S}_i , defined by:

$$\mathbf{p}_i = \frac{1}{S_i} \sum_{u_{i,k} \in \mathcal{S}_i} G_U(u_{i,k}, \mathbf{X}_H \mid \theta) \quad i \in \mathcal{I}_{\mathcal{T},N} \quad (4.5)$$

Equation 4.5 encourages the prototype \mathbf{p}_i to learn a representative vector summarizing its cluster of interacted users. However, we face two critical challenges in handling tail items: first, due to sparse support sets, the prototypes are often *noisy* and sensitive to *outliers*;

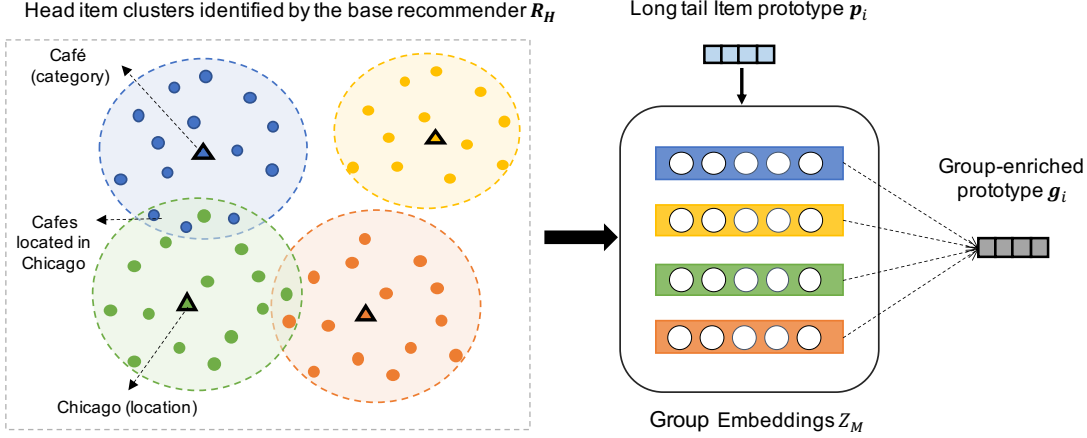


Figure 4.5: Group embeddings can be viewed as centroids of overlapping item clusters, e.g., Cafe (category) or Chicago (location); items (Cafes located in Chicago) may often belong to overlapping item clusters.

second, due to substantial *heterogeneity* in the tail, simplistic averaging may lack the *resolution* to discriminate across diverse tail items. Thus, the few-shot recommender \mathbf{R}_F needs a strong *inductive bias* during prototype learning to avoid overfitting, and yet have sufficient expressivity to learn *discriminative* prototypes.

Head-Tail Meta Knowledge Transfer

We now exploit the *item-to-item* relationship knowledge acquired by the neural base recommender \mathbf{R}_B (Section 4.4.1) as an inductive bias to enhance the item prototype \mathbf{p}_i (Equation 4.5). Given an item $i \in \mathcal{I}_{\mathcal{T},N}$ with limited support users, extracting knowledge from the most related items to item i (based on the item-to-item proximities) can provide valuable guidance to synthesize robust item prototypes.

However, direct knowledge transfer from the base recommender \mathbf{R}_B to the few-shot recommender \mathbf{R}_F is technically challenging: dynamically identifying related items for each few-shot item during prototype construction is not scalable due to the arbitrary complexity of \mathbf{R}_B and the high cardinality of item sets in massive inventories. Thus, we extract a *compact* representation of the *item-item proximity* knowledge (Equation 4.3) discovered by \mathbf{R}_B and transfer this knowledge to enhance item prototypes.

Group-Enhanced Item Prototype Learning. We learn a set of M group embeddings \mathcal{Z}_M as *basis vectors* modeling *item-item proximity* in the latent space of the base recommender \mathbf{R}_B , i.e., if $p_B(i, j)$ is high for a pair of items $i, j \in \mathcal{I}$, then items i and j are likely to exhibit similar characteristics. Formally, the set of group embeddings \mathcal{Z}_M are defined by

the following equation below:

$$\mathcal{Z}_M = \{z_m \in \mathbb{R}^D : 1 \leq m \leq M\} \quad M \ll |\mathcal{I}| \quad (4.6)$$

As depicted in Figure 4.5, we intuitively visualize the group embeddings as *discriminative centroids* of *overlapping* clusters of items identified by \mathbf{R}_B , and conversely view items as *mixtures* over the group embeddings, *e.g.*, each centroid may represent a contextual factor such as a restaurant type (Cafe) or location (Chicago), while restaurants are mixture of multiple centroids (Cafe located in Chicago) and belong to overlapping item clusters.

To enhance the prototype \mathbf{p}_i of item $i \in \mathcal{I}_{\mathcal{T},N}$, we synthesize a *group-enhanced prototype* $\mathbf{g}_i \in \mathbb{R}^D$ as a mixture over the M group embeddings. The mixture coefficients are estimated by a learnable attention mechanism [130] to measure *compatibility* α_{im} between the prototype \mathbf{p}_i and each centroid $z_m \in \mathcal{Z}_M$. We parameterize the attention function with a lightweight network comprised of an auxiliary set $\mathcal{K}_M = \{\mathbf{k}_m \in \mathbb{R}^D : 1 \leq m \leq M\}$ of trainable keys to index the group embeddings. We implement the attention function with an inner product followed by softmax normalization, as:

$$\mathbf{g}_i = \sum_{m=1}^M \alpha_{im} \mathbf{z}_m \quad \alpha_{im} = \frac{\exp(\mathbf{W}_q \mathbf{p}_i \cdot \mathbf{k}_m)}{\sum_{m'=1}^M \exp(\mathbf{W}_q \mathbf{p}_i \cdot \mathbf{k}_{m'})} \quad (4.7)$$

where $\mathbf{W}_q \in \mathbb{R}^{D \times D}$ is projects the prototype \mathbf{p}_i into a query to index the centroids. The group-enhanced item prototype \mathbf{g}_i relates the different centroids via attention, transferring knowledge to few-shot items with sparse support sets.

Task-level Stochastic Knowledge Distillation. We present a *knowledge distillation* strategy [208] to learn compact *group embeddings* \mathcal{Z}_M that capture *item-item relationships* in \mathbf{R}_B . We transfer knowledge from a high-capacity *teacher* model (base recommender \mathbf{R}_B) to a compact *student* model (group embeddings \mathcal{Z}_M) by encouraging the student to emulate predictions of the teacher; \mathcal{Z}_M is trained to emulate the item proximity distribution in \mathbf{R}_B .

Aligning pairwise item proximities over all items in \mathcal{I} is not scalable; thus, we operate at the granularity of each meta-training task \mathcal{T} . For each item $i \in \mathcal{I}_{\mathcal{T},N}$, we compute a soft probability distribution $p_B(j | i, \mathbf{R}_B)$ for the teacher model \mathbf{R}_B (based on equation 4.3) over other items $j \in \mathcal{I}_{\mathcal{T},N}$ in the task \mathcal{T} , described by:

$$p_B(j | i, \mathbf{R}_B) = \frac{\exp(p_B(i, j)/T)}{\sum_{k \in \mathcal{I}_{\mathcal{T},N}} \exp(p_B(i, k)/T)} \quad i, j \in \mathcal{I}_{\mathcal{T},N} \quad (4.8)$$

where $T > 0$ is a temperature scaling hyper-parameter to regulate the rate of knowledge

transfer. We analogously define the item similarity distribution $p_F(j \mid i, \mathbf{Z}_M)$ for the student model \mathbf{Z}_M based on the metric space proximity $\text{sim}_m(\cdot)$ of group-enhanced prototypes \mathbf{g}_i and \mathbf{g}_j for items $i, j \in \mathcal{I}_{\mathcal{T},N}$, defined by:

$$p_F(j \mid i, \mathbf{Z}_M) = \frac{\exp(\text{sim}_m(\mathbf{g}_i, \mathbf{g}_j))}{\sum_{k \in \mathcal{I}_{\mathcal{T},N}} \exp(\text{sim}_m(\mathbf{g}_i, \mathbf{g}_k))} \quad i, j \in \mathcal{I}_{\mathcal{T},N} \quad (4.9)$$

We align the two distributions by minimizing cross-entropy between their task-level similarities [208]. Since each item is typically related to very few items within the task \mathcal{T} , our *stochastic knowledge distillation* loss L_G minimizes distribution divergence over the top- n ($n \approx 10$) related items (out of N) identified by the teacher \mathbf{R}_B , by:

$$L_G = -\frac{1}{nN} \sum_{i \in \mathcal{I}_{\mathcal{T},N}} \sum_{j \in \pi_{B,n}(i)} p_B(j \mid i, \mathbf{R}_B) \log p_F(j \mid i, \mathbf{Z}_M) \quad (4.10)$$

where $\pi_{B,n}(i) = \text{Top}_n(p_B(\cdot \mid i, \mathbf{R}_B))$ denotes the top- n most related items to item i within $\mathcal{I}_{\mathcal{T},N}$ based on the teacher \mathbf{R}_B . Distinct from prior distillation approaches [208, 209] that match student and teacher predictions over a fixed set of classes, L_G is stochastic and thereby more efficient since it matches the top- n item proximity distributions over different sets of sampled items in each task. The distillation loss L_G transfers knowledge from \mathbf{R}_B to the group embeddings \mathbf{Z}_M and is trained jointly with the rest of the framework.

Item Prototype Fusion via Neural Gating

The initial prototype \mathbf{p}_i for item $i \in \mathcal{I}_{\mathcal{T},N}$ direct encodes its support users \mathcal{S}_i , while the group-enhanced prototype \mathbf{g}_i captures the knowledge transferred from related items (via base recommender \mathbf{R}_B). We design a *gating mechanism* [210] that adaptively selects salient feature dimensions from \mathbf{p}_i and \mathbf{g}_i to infer the final *gated item prototype* $\mathbf{e}_i \in \mathbb{R}^D$. Specifically, we introduce a neural gating layer to merge \mathbf{p}_i and \mathbf{g}_i by learning a non-linear gate to flexibly modulate information flow via dimension re-weighting, defined by:

$$\begin{aligned} \text{gate} &= \sigma(\mathbf{W}_{g_1} \mathbf{p}_i + \mathbf{W}_{g_2} \mathbf{g}_i + \mathbf{b}_g) \quad i \in \mathcal{I}_{\mathcal{T},N} \\ \mathbf{e}_i &= \text{gate} \odot \mathbf{p}_i + (\mathbf{1} - \text{gate}) \odot \mathbf{g}_i \end{aligned} \quad (4.11)$$

where $\mathbf{W}_{g_1} \in \mathbb{R}^{D \times D}$, $\mathbf{W}_{g_2} \in \mathbb{R}^{D \times D}$, and $\mathbf{b}_g \in \mathbb{R}^D$ are learnable weight parameters in the neural gating layer, \odot denotes the element-wise (or hadammard) product operation, and σ is the sigmoid non-linearity function.

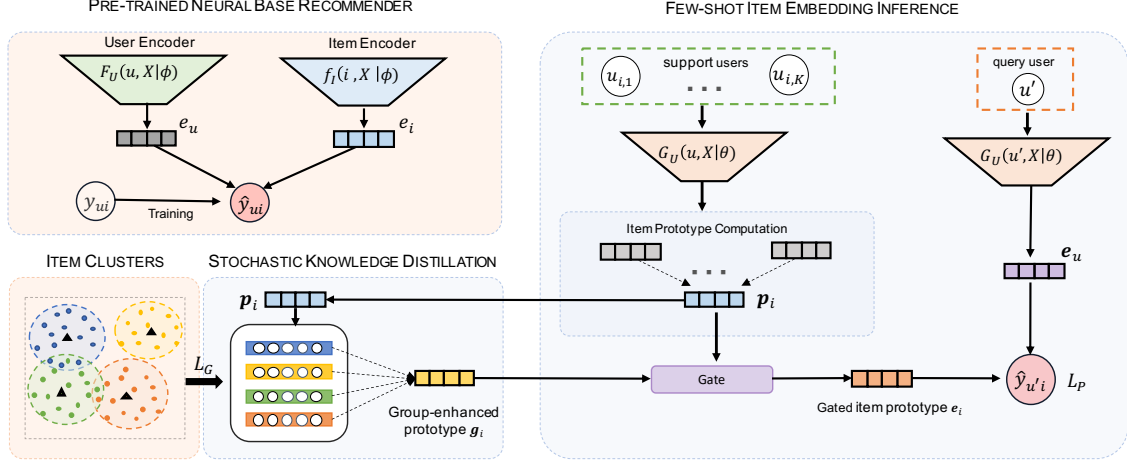


Figure 4.6: Architecture diagram of PROTOCF depicting the different model components: pre-trained neural base recommender \mathbf{R}_B (top left), group embedding learning via stochastic knowledge distillation L_G (bottom left), initial item prototype construction via support set averaging followed by group-enrichment and adaptive gating to construct gated item prototype \mathbf{e}_i (right).

Few-shot Recommender Training

We induce few-shot item rankings for the query users \mathbf{Q} using the gated item prototypes $\{\mathbf{e}_i : i \in \mathcal{I}_{\mathcal{T},N}\}$ in meta-training task \mathcal{T} . We generate item recommendations for each query user $u' \in \mathbf{Q}$ (over the task items $\mathcal{I}_{\mathcal{T},N}$) by measuring the similarity $\text{sim}_m(\cdot)$ with each item prototype \mathbf{e}_i . Each meta-training task \mathcal{T} minimizes a negative log-likelihood L_P between the few-shot item recommendations for query users \mathbf{Q} and their ground-truth interactions in \mathcal{T} , which is defined by:

$$L_P = -\frac{1}{KN} \sum_{i \in \mathcal{I}_{\mathcal{T},N}} \sum_{u'_{i,k'} \in Q_i} \log p_F(i | u'_{i,k'}, \theta) \quad (4.12)$$

where $p_F(i | u'_{i,k'}, \theta)$ is computed based on cosine similarity and the choice of likelihood function for few-shot training.

The overall loss L is composed of two terms, the few-shot recommendation loss L_P (Equation 4.12) and the knowledge distillation loss L_G for group embedding learning (Equation 4.10), given by:

$$L = L_P + \lambda L_G \quad (4.13)$$

where λ is a tunable hyper-parameter that balances the two loss terms. Algorithm 4.1 summarizes the training procedure of our entire framework PROTOCF.

Model Inference

We infer the *gated prototype* \mathbf{e}_i for each item $i \in \mathcal{I}$ by sub-sampling K interactions from its historical interactions N_i as the support set (Equation 4.11). We generate personalized item recommendations for each user $u \in \mathcal{U}$ by:

$$\hat{y}_f(u, i) = \text{sim}_m(\mathbf{e}_u, \mathbf{e}_i) \quad i \in \mathcal{I} \quad \mathbf{e}_u = G_U(u, \mathbf{X} \mid \theta) \quad (4.14)$$

The few-shot recommender \mathbf{R}_F is designed for recommendations over tail items (sparse interactions), while the base recommender \mathbf{R}_B is effective for head items (abundant interactions). In PROTOCF, we compute rankings over the entire item set \mathcal{I} by ensembling predictions from \mathbf{R}_B and \mathbf{R}_F . One simple interpolation approach is given by:

$$\hat{y}(u, i) = (1 - \eta) \cdot \hat{y}_b(u, i) + \eta \cdot \hat{y}_f(u, i) \quad (4.15)$$

where $0 < \eta < 1$ balances \mathbf{R}_F and \mathbf{R}_B . We empirically show effective overall item recommendations with $\eta = 0.5$.

4.4.3 Model Details

We now discuss different choices of likelihood functions for training the few-shot recommender \mathbf{R}_F and architectural details of the base recommender \mathbf{R}_B .

Few-shot Likelihood

We examine two likelihood functions in neural CF models: *multinomial* and *logistic*.

- **Multinomial log-likelihood:** The scores $\text{sim}_m(\mathbf{e}_{u'}, \mathbf{e}_i)$ for each query user $u' \in \mathcal{Q}_i$, over the N gated item prototypes in meta-training task \mathcal{T} , are normalized to produce probabilities $p_F(i \mid u', \theta)$ over the N items $\mathcal{I}_{\mathcal{T}, N}$, defined by:

$$p_F(i \mid u', \theta) = \frac{\exp(\text{sim}_m(\mathbf{e}_{u'}, \mathbf{e}_i))}{\sum_{j \in \mathcal{I}_{\mathcal{T}, N}} \exp(\text{sim}_m(\mathbf{e}_{u'}, \mathbf{e}_j))} \quad u' \in \mathcal{Q}_i \quad (4.16)$$

The resulting loss L_G (with Equation 4.12) is also known as the *cross-entropy* loss, defined over the N items of meta-training task \mathcal{T} .

- **Logistic log-likelihood:** The relevance $\hat{y}_{u'i} = \text{sim}_m(\mathbf{e}_{u'}, \mathbf{e}_i)$ for item $i \in \mathcal{I}_{\mathcal{T}, N}$ to query user $u' \in \mathcal{Q}_i$, is transformed into a probability using the sigmoid function σ . We use a

Algorithm 4.1: PROTOCF: *Prototypical Collaborative Filtering*

Input: User-item interactions \mathbf{X} , meta-training task distribution $p(\mathbb{T})$.

Output: Function $f_T(i | u), i \in \mathcal{I}_T$ for few-shot item recommendations.

- 1: Pre-train the neural base recommender \mathbf{R}_B on the training interactions \mathbf{X} to learn $f_{H,\phi}(i | u)$ (eqn 4.2) and freeze parameters ϕ .
 - 2: Initialize user preference encoder $G_U(\cdot | \theta)$ of few-shot recommender \mathbf{R}_F with parameters $F_U(\cdot | \phi)$ from the base recommender \mathbf{R}_B .
 ▷ Meta-training: learn-to-recommend few-shot items.
 - 3: **while** *not converged* **do**
 - 4: Sample a meta-training task $\mathcal{T} = \{\mathcal{I}_{\mathcal{T},N}, \mathcal{S}, \mathcal{Q}\} \sim p(\mathbb{T})$.
 - 5: Calculate the initial \mathbf{p}_i and group-enhanced \mathbf{g}_i prototypes for items $i \in \mathcal{I}_{\mathcal{T},N}$ (eqn 4.5 and eqn 4.7).
 - 6: Compute the knowledge distillation loss L_G (eqn 4.10).
 - 7: Compute gated prototype \mathbf{e}_i for items $i \in \mathcal{I}_{\mathcal{T},N}$ (eqn 4.11).
 - 8: Estimate few-shot recommendation loss L_P (eqn 4.12) with *multinomial* (eqn 4.16) or *logistic* (eqn 4.17) log-likelihoods.
 - 9: Minimize overall loss L (eqn 4.13) using mini-batch gradient descent.
 ▷ Model Inference: few-shot and overall item recommendations.
 - 10: Generate item recommendations over \mathcal{I} for user $u \in \mathcal{U}$ (eqn 4.15)
-

confidence weight $\beta > 0$ to re-weight the likelihood of observed 1's which are far fewer than the unobserved 0's in implicit feedback. The logistic log-likelihood is given by:

$$\log p_F(i | u', \theta) = \beta \log \sigma(\hat{y}_{u'i}) + \sum_{j \in \mathcal{I}_{\mathcal{T},N}, u' \notin N_j} \log(1 - \sigma(\hat{y}_{u'j})) \quad (4.17)$$

The cosine similarity scores are also scaled to match the non-saturating regimes of the softmax and sigmoid functions in the multinomial and logistic log-likelihoods respectively.

Neural Base Recommender Architecture

We consider three neural collaborative filtering methods, matrix factorization (MF) [87], denoising (CDAE) [89] and variational (VAE-CF) [90] autoencoders, as base recommenders \mathbf{R}_B (Equation 4.1) for our framework PROTOCF. We briefly describe their architectures:

- **Matrix Factorization (MF)** [87]: The user F_U and item F_I encoders are learnable latent embeddings for each user and item respectively, which are trained using a logistic log-likelihood training objective.
- **Variational AutoEncoder (VAE-CF)** [90]: The user encoder F_U is a two-layer Multi-Layer Perceptron transforming the binary user preference vector $\mathbf{x}_u \in \mathbb{R}^I$ into a

D -dimensional user embedding $\mathbf{h}_u \in \mathbb{R}^D$, defined by:

$$\mathbf{h}_u = F_U(\mathbf{x}_u \mid \phi) = \sigma(\mathbf{W}_2^T(\sigma(\mathbf{W}_1^T \mathbf{x}_u + b_1) + b_2) \quad (4.18)$$

with learnable weight matrices $\mathbf{W}_1 \in \mathbb{R}^{I_H \times D}$ and $\mathbf{W}_2 \in \mathbb{R}^{D \times D}$, biases $b_1, b_2 \in \mathbb{R}^D$, and $\tanh(\cdot)$ activations for non-linearity σ . The item encoder F_I is a latent embedding. VAE-CF uses a multinomial likelihood with KL-divergence regularization.

- **Denoising AutoEncoders (CDAE)** [89]: The user encoder F_U operates on partially corrupted inputs \mathbf{x}_u and adds an auxiliary per-user embedding to the encoder output from Equation 4.18. We train CDAE using a multinomial log-likelihood, albeit without the KL-divergence prior regularization.

4.5 EXPERIMENTS

We present experiments on four real-world datasets to evaluate our framework PROTOCF. We introduce datasets, baselines and experimental setup in Sections 4.5.1, 4.5.2 and 4.5.3. We propose four research questions to guide our experiments:

- (RQ₁) Does PROTOCF beat state-of-the-art NCF and sparsity-aware methods on *overall* recommendation performance?
- (RQ₂) What is the impact of item *interaction sparsity* on the *few-shot* recommendation performance of PROTOCF?
- (RQ₃) How do the different *architectural* choices impact the few-shot and overall performance of PROTOCF?
- (RQ₄) How do the *hyper-parameters* (distillation loss balance factor λ and meta-training task size N) affect PROTOCF?

Finally, we discuss the limitations of our PROTOCF model and explore future directions in Section 4.5.8.

4.5.1 Datasets

We conducted experiments on four publicly available benchmark datasets, including two online product review platforms (*Epinions*, *Yelp*) and two check-in based social networks (*Weeplaces*, *Gowalla*).

Dataset	Epinions	Yelp	Weeplaces	Gowalla
# Items	9,035	10,451	11,679	27,920
# Users	9,729	13,926	6,167	17,848
# Interactions	260,263	465,386	427,236	907,351
# Interactions per item	28.81	44.53	36.58	32.50

Table 4.3: *Dataset statistics*

- **Epinions**¹: product ratings from an e-commerce platform; we retain interactions with ratings higher than two.
- **Yelp**²: user ratings on local businesses located in the state of Arizona, obtained from Yelp dataset challenge round 13.
- **Weeplaces**³: we extract business check-ins from Weeplaces of different categories, including Nightlife, Outdoors, Entertainment, Travel and Food, across all cities in the United States.
- **Gowalla** [211]: restaurant check-ins in Gowalla by users across different cities in the United States.

We pre-process the datasets to retain users and items with at least ten interactions using the 10-core setting [212] (Table 4.3).

4.5.2 Baselines

We present comparisons against prior work that broadly fall into two categories: standard *neural collaborative filtering* models, and *sparsity-aware long-tail item recommendation* models, including regularization and meta-learning techniques.

- **Neural Base Recommenders**: neural CF methods with matrix factorization (BPR) [87], and autoencoder models VAE-CF [90] and CDAE [89] (described in Section 4.4.3).
- **NCF** [88]: neural CF model with non-linear interactions (via neural layers) between the user and item embeddings.
- **NGCF** [91]: state-of-the-art graph-based NCF with embedding propagation layers on the user-item interaction graph.

¹<https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm>

²<https://www.yelp.com/dataset>

³<https://www.yongliu.org/datasets/>

- **Cofactor** [193]: regularized MF to capture inter-item associations by jointly decomposing the user-item interaction matrix and the item-item co-occurrence matrix, with shared latent item factors.
- **EFM** [194]: embedding factorization model that uses item-item co-occurrence with bayesian personalized ranking.
- **DropoutNet** [108]: randomly dropout latent CF embedding for regularization in content-based CF for cold-start (auxiliary content). We adapt it to long-tail items by replacing content embeddings with prototypes (over support set).
- **MetaRec-LWA** [116] meta-learning method to construct recommendation models for cold-start users with user-specific linear transforms; we adapt it to construct few-shot item embeddings.
- **MetaRec-NLBA** [116]: meta-learning recommendation model that learns user-specific biases parameterized by non-linear layers, to replace the user-specific weights in MetaRec-LWA.

Note that we omit experimental baseline comparisons with gradient-based meta-learning recommenders [110, 112], since they are designed for cold-start recommendation in the presence of auxiliary attributes. We test PROTOCF by adopting BPR [87] (matrix factorization), VAE-CF [90], and CDAE [89] as base recommenders of our framework.

4.5.3 Experimental Setup

In each dataset, we randomly split the user interactions of all items into train (70%) and test (30%) sets, which ensures consistent interaction distributions across train and test sets. We also use 10% of training interactions as validation for hyper-parameter tuning. We use NDCG@K and Recall@K as evaluation metrics, computed based on the rank of ground-truth test interactions in top- K ranked item lists.

We design a unified model to enhance few-shot recommendation on the tail without affecting the overall performance. We first evaluate *overall* recommendations over the entire item set \mathcal{I} followed by *few-shot* performance analysis over the long-tail.

All experiments are conducted on a Tesla K-80 GPU using PyTorch. PROTOCF is trained using episodic learning for a maximum of 300 episodes with a meta-training task size of 512 items using Adam optimizer. PROTOCF uses a multinomial log-likelihood for few-shot training and uses support sets of upto $K = 10$ and query sets of $K' = 5$ interactions

Dataset	Epinions		Yelp		Weeplaces		Gowalla	
Metric	N@50	R@50	N@50	R@50	N@50	R@50	N@50	R@50
STANDARD NEURAL COLLABORATIVE FILTERING METHODS								
BPR [87]	0.0860	0.1666	0.0749	0.1416	0.2537	0.3778	0.1661	0.2703
NCF [88]	0.0878	0.1694	0.0752	0.1429	0.2462	0.3694	0.1702	0.2745
NGCF [91]	0.0913	0.1725	0.0826	0.1579	0.2533	0.3764	0.1696	0.2758
VAE-CF [90]	0.0938	0.1778	0.0854	0.1602	0.2482	0.3730	0.1710	0.2769
CDAE [89]	0.0927	0.1774	0.0870	0.1611	0.2570	0.3760	0.1634	0.2644
SPARSITY-AWARE LONG-TAIL ITEM RECOMMENDATION METHODS								
DropoutNet [108]	0.0881	0.1697	0.0761	0.1435	0.2516	0.3751	0.1697	0.2768
Cofactor [193]	0.0845	0.1639	0.0734	0.1402	0.2342	0.3539	0.1596	0.2642
EFM [194]	0.0742	0.1534	0.0741	0.1403	0.2306	0.3429	0.1532	0.2584
MetaRec-NLBA [116]	0.0453	0.0937	0.0381	0.0875	0.1698	0.2889	0.0753	0.1384
MetaRec-LWA [116]	0.0467	0.0943	0.0392	0.1425	0.1702	0.2997	0.0722	0.1391
PROTOTYPICAL COLLABORATIVE FILTERING RECOMMENDERS (PROTOCF)								
ProtoCF + BPR	0.0964	0.1812	0.0815	0.1533	0.2576	0.3879	0.1737	0.2800
ProtoCF + VAE	0.0977	0.1830	0.0857	0.1605	0.2725	0.4035	0.1899	0.3004
ProtoCF + CDAE	0.0972	0.1824	0.0883	0.1623	0.2697	0.4011	0.1786	0.2875
Percentage Gains	4.16%	2.92%	1.50%	0.75%	6.03%	6.80%	11.05%	8.49%

Table 4.4: Overall item recommendation results on four datasets, $R@K$ and $N@K$ denote Recall@ K and NDCG@ K metrics at $K = 50$. Sparsity-aware models are generally outperformed by standard NCF methods on overall item recommendation; PROTOCF achieves overall NDCG@50 gains of 6% and Recall@50 gains of 4% (over the best baseline) across all datasets.

per item. We learn $M = 100$ group embeddings \mathbf{Z}_M , tune the learning rate and balance hyper-parameter λ (Equation 4.13) in the range $\{10^{-4}, 10^{-3}, 10^{-2}\}$, and use dropout regularization with a rate of 0.5. We tune baselines in hyper-parameter ranges centered at the author-provided values on each dataset and set the latent embedding dimension to 128 for consistency. Our implementation of PROTOCF and datasets are publicly available⁴.

4.5.4 Overall Recommendation Results (RQ₁)

Our experimental results comparing the *overall* recommendation performance of PROTOCF with competing baselines are shown in Table 4.4. We summarize our key empirical observations below:

- Neural CF models based on autoencoders (VAE-CF, CDAE) and graph neural networks (NGCF) outperform other latent-factor models (NCF, BPR) on overall model

⁴<https://github.com/aravindsankar28/ProtoCF>

performance (Table 4.4).

- Model regularization strategies that use item co-occurrence information (CoFactor, EFM) to improve long-tail item recommendations, are noticeably worse than BPR in overall performance.
- Sparsity-aware meta-learning models (MetaRec) perform poorly in overall item rankings. One potential reason is their inability to effectively exploit or transfer knowledge from head items.
- PROTOCF outperforms state-of-the-art baselines (NDCG@50 gains of 5% on average) on *overall* item ranking, with consistent gains for the variants of all neural base recommenders.

Next, we examine the few-shot performance of PROTOCF.

4.5.5 Few-Shot Recommendation Results (RQ₂)

To evaluate few-shot results, we qualitatively analyze performance for long-tail items with sparse interactions. Specifically, we compare recommendation results for long-tail items with varying number of training interactions K (5 to 30) by computing Recall@50 metrics only on their associated test interactions (Figure 4.7). We only include the base recommenders BPR and VAE-CF here for comparison since they consistently outperform other sparsity-aware variants.

We find that model performance generally increases with item interaction count; PROTOCF achieves significant performance gains for items with less than 20 interactions. Episodic training with knowledge transfer is one of the key factors responsible for our higher gains over items with sparse interactions (small values of K).

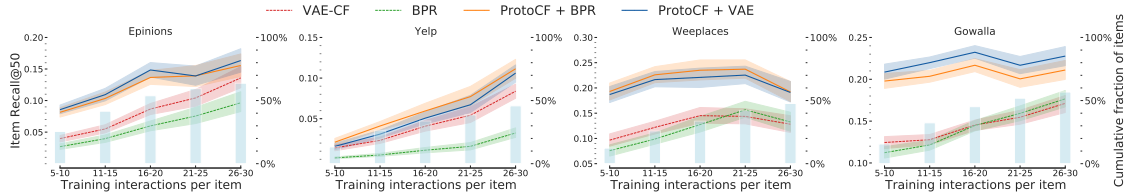


Figure 4.7: *Few-shot item recommendation results: Performance comparison for long-tail items with varying number of training interactions K (5 to 30); lines denote model performance (Recall@50) and background histograms indicate the cumulative fraction of the item inventory covered by tail items with $\leq K$ impressions. Overall performance generally increases with K for all models; PROTOCF achieves notably stronger gains (over baselines) for items with few training interactions (small K).*

To evaluate the impact of *interaction sparsity* across the entire item set, we compare overall recommendation performance (Recall@K) across item-groups with different sparsity levels. We divide the test set into ten equal-sized item-groups, sorted in increasing order by the average number of interactions per item. Figure 4.8 compares PROTOCF against two base recommenders BPR and VAE-CF.

From figure 4.8, model performance is lower on the *long-tail* for base recommenders BPR and VAE due to severe interaction sparsity. Notably, PROTOCF achieves much higher item recall scores on the *tail* items with significant gains over the corresponding base recommenders, while ensuring comparable performance on the *head* items. Knowledge transfer of *item-to-item relationships* via group embeddings and pre-trained user encoder enables PROTOCF to learn discriminative prototypes for tail items with sparse interactions.

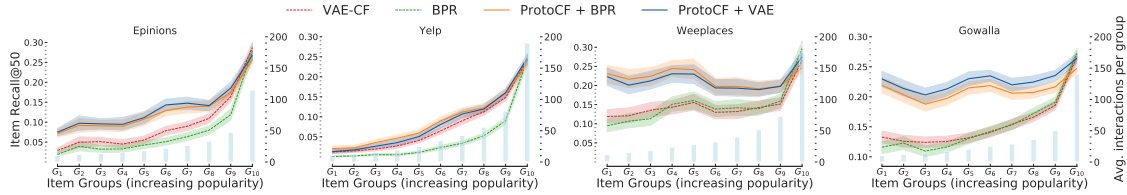


Figure 4.8: *Impact of item interaction sparsity: Performance comparison for item-groups sorted in increasing order by their average training interaction counts; lines denote model performance (Recall@50) and background histograms indicate the average number of interactions in each item-group. PROTOCF has significant performance gains (over baselines) on the tail items (item-groups G_1 to G_8) while maintaining comparable performance on the head items (item-groups G_9 to G_{10}).*

4.5.6 Model Ablation Study (RQ₃)

We examine the impact of different *architectural* design choices in PROTOCF on its *overall* and *few-shot* performance (Table 4.5). We choose VAE-CF as the base recommender to instantiate PROTOCF due to its consistent results. Here, we report few-shot performance by only considering the test interactions of *long-tail* items with less than 20 training interactions. Note that most ablation variants do not have a significant impact on overall results since all predictions are computed by ensembling \mathbf{R}_F and \mathbf{R}_B (Equation 4.15). Our key design choices and empirical insights are shown below:

- **Remove Prototype Gating:** We replace the gating layer (Equation 4.11) that adaptively fuses the initial and group-enhanced item prototypes, with a simpler additive operation. Adaptive gating contributes 5-11% few-shot performance gains.

Dataset	Epinions		Gowalla	
Metric	Overall R@50	Few-shot R@50	Overall R@50	Few-shot R@50
ProtoCF	0.1830	0.1070	0.3004	0.2195
w/o Prototype Gating	0.1823	0.0948	0.2992	0.2082
w/o Knowledge Distillation	0.1805	0.0869	0.2923	0.1983
ProtoCF-Avg	0.1801	0.0712	0.2898	0.1696
PROTOCF-logistic	0.1804	0.0896	0.2853	0.1843
VAE-CF [90]	0.1778	0.0549	0.2769	0.1316
MetaRec-LWA [116]	0.0943	0.0898	0.1391	0.1804

Table 4.5: *Model ablation study of PROTOCF; few-shot performance is reported for tail items (less than 20 training interactions). Knowledge transfer and prototype gating contribute 10-19% and 5-11% to few-shot gains respectively.*

- **Remove Knowledge Distillation:** We test the importance of item-to-item relationships transferred from the base recommender via distillation loss L_G ; here, we exclude knowledge transfer from group embeddings \mathbf{Z}_M and only train on the few-shot loss L_P . Removing distillation loss L_G reduces few-shot results by 10-19%.
- **Averaging-based Prototype (ProtoCF-AVG):** We directly use the averaging-based initial item prototype (Equation 4.5) for inference; here, we also exclude the pre-trained parameter initialization for the user encoder G_U (from F_U in the base recommender). PROTOCF-AVG (without any knowledge transfer) is worse than PROTOCF by a margin of 20-30%, yet outperforms the base recommender VAE-CF by nearly 30% on few-shot items.
- **Logistic log-likelihood (ProtoCF-logistic):** We train PROTOCF using *logistic* log-likelihood (Equation 4.17). PROTOCF with multinomial log-likelihood (Equation 4.16) outperforms PROTOCF-logistic by a considerable margin; this validates prior findings [90] on the efficacy of multinomial for top- N recommendation.

Note that the meta-learning baseline MetaRec-LWA improves few-shot results (compared to VAE-CF), but forgets knowledge of head items resulting in poor overall recommendations.

4.5.7 Parameter Sensitivity (RQ₄)

We analyze sensitivity to the hyper-parameter λ that weights the knowledge distillation loss L_G (Equation (4.13)). In Figure 4.9 (a), we show the effect of λ on few-shot results

with base recommenders BPR and VAE-CF on Gowalla. We empirically find the optimal value of λ to be 0.01 for both models, which also transfers across datasets.

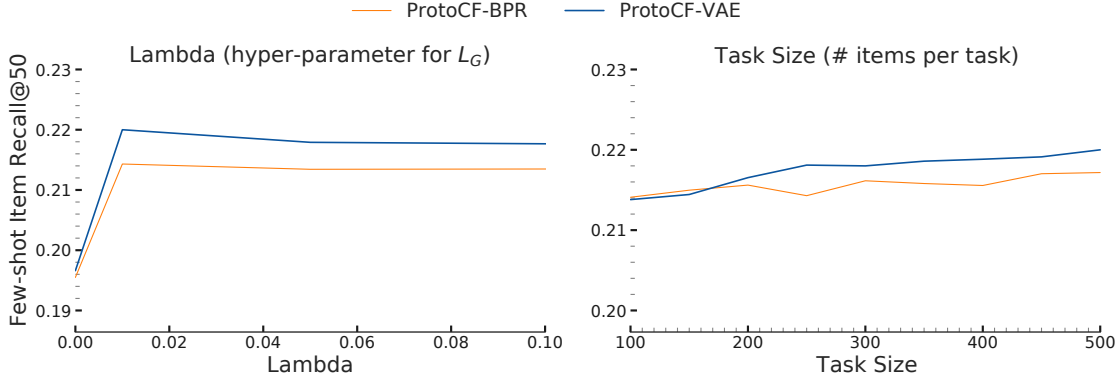


Figure 4.9: *Few-shot performance on Gowalla (for tail items with less than 20 training interactions) is higher for larger meta-training tasks; the empirically optimal value of balance factor $\lambda = 0.01$ also transfers across all datasets.*

We investigate the impact of *task size*, which is the number of items N sampled in each meta-training task \mathcal{T} . In Figure 4.9 (b), performance typically increases with task size (and stabilizes ~ 400), due to a larger set of sampled items available for ranking; this observation is consistent with prior few-shot learning studies [101].

4.5.8 Discussion

Our framework PROTOCF is orthogonal to architectural advances in neural recommendation models that enhance representational capacity to learn from massive interaction data. We adapt expressive neural recommenders to develop light-weight few-shot models tailored to the long-tail. Furthermore, our few-shot formulation requires no side information and can further be easily adapted to address the long-tail of users.

PROTOCF learns a metric space predicated on knowledge transfer over a shared user space. Few-shot transfer learning across domains with disjoint feature spaces is a potential future direction.

4.6 CONCLUSION

We formulate long-tail item recommendation as learning-to-embed items with sparse interactions. A few-shot learning framework PROTOCF is introduced to extract meta-knowledge

across a collection of training tasks designed to simulate tail item ranking. PROTOCF efficiently transfers knowledge from arbitrary base recommenders to construct discriminative prototypes for items with few interactions. Our experiments indicate 5% overall performance gains (Recall@50) for PROTOCF over the state-of-the-art, with notable 60-80% few-shot performance gains (Recall@50) on long-tail items with less than 20 training interactions.

In this chapter, we examined another common transductive learning setting of collaborative filtering over bipartite user-item interactions that are prevalent in several online interaction platforms. To overcome entity-level (user/item) interaction sparsity challenges that arise due to heavy-tailed distributions in user interests and interaction patterns, we introduced a meta-learning framework for few-shot recommendation. In the next part of this dissertation, we examine several inductive learning applications targeted at behavior modeling for new (or unseen) entities that are only observed during inference.

CHAPTER 5: INFVAE: SOCIAL REGULARIZATION TO PREDICT INFORMATION DIFFUSION

5.1 INTRODUCTION

In social media, information disseminates or *diffuses* to a large number of users through content posting or re-sharing behavior, resulting in a *cascade* of user activations, *e.g.*, a user voting a news story on Digg (a social news sharing website) triggers a series of votes from multiple users, who may be his friends or other users interested in the same story. Given a set of *activated* seed users who have posted a piece of user-generated content, diffusion models aim to predict the set of all influenced users who re-share it. Diffusion modeling has widespread social media applications, including viral marketing [213], limiting misinformation spread, personalized recommendations [180], and popularity prediction [214].

The diffusion prediction problem has received significant attention in the research community. The earliest methods assume that the mechanism of social influence propagation is known a priori, *e.g.*, the Independent Cascade (IC) [70, 73] and Linear Threshold (LT) models [71] that associate each link with a non-negative weight indicative of pairwise user-user influence. Unlike pre-defined propagation hypotheses [70], recent methods learn data-driven diffusion models from collections of user activation sequences (*diffusion cascades*). Prior diffusion models broadly fall into two categories.

Probabilistic generative cascade models use hand-crafted features including social roles [215], communities [75], topics [72], and structural patterns [216]. Such methods rely on feature engineering that requires manual effort and extensive domain knowledge, and are limited by the modeling capacity of carefully chosen probability distributions.

Representation learning methods avoid feature extraction by learning user representations characterizing their influencing ability and conformity [79, 80]. Prior work mainly consider the impact of temporal user-user influence learned from historical diffusion cascades; prior methods project cascades onto local social neighborhoods to generate Directed Acyclic Graphs (DAGs), and propose extensions of Recurrent Neural Networks (RNNs) to model *temporal influence*. State-of-the-art models include DAG-structured LSTMs [81] that explicitly operate on induced DAGs and attention-based RNNs [82, 83, 217] that implicitly consider cross-correlations across activated seed users for diffusion prediction.

Prior work only model the sequence or projected social structure (induced DAG) of previously influenced users while ignoring *social structures that do not manifest in cascades*; they only capture the temporal correlation in diffusion behaviors known as *temporal influence* or *contagion* [13]. Consider a Twitter user interested in politics who follows famous

political leaders and joins interest groups; this induces transitive connections to other users that however may not manifest in cascades unless she re-tweets or posts content. *Social homophily* [11] suggests *stronger ties* between users with shared traits or interests, which induces correlated diffusion behaviors without direct causal influence. Social graph connectivity information reveals insights into *homophily* and is critical to model diffusion behaviors for the vast majority of users who rarely post content and appear in cascades.

However, homophilous diffusion and contagion can result in differing dynamics, *e.g.*, contagions are self-reinforcing and viral while homophily hinges on users’ preferences or traits. Indeed, social homophily and temporal influence are fundamentally confounded in observational studies of diffusion processes [13], which makes it challenging to contextually model the impact of both factors. Thus, our key objective is to *develop a principled neural framework to contextually model the co-variance of temporal influence (recent posting activities) with social homophily (structural graph connectivity) for diffusion prediction*.

Present Work: Our *social regularization* framework INFVAE jointly models *homophily* through *social* embeddings preserving social network proximity and *influence* through *temporal* embeddings encoding the relative sequential order of user activations. Motivated by the capabilities of variational autoencoders (VAEs) [135] in alleviating sparsity via Gaussian priors [90] and the expressive power of Graph Neural Networks (GNNs) [28, 39], we adopt graph-VAEs to model social homophily. We learn structure-preserving social embeddings for each user through VAEs, and differentiate their social roles (influential versus susceptible) towards diffusion modeling. Given an initial set of seed user activations, INFVAE utilizes an expressive *co-attentive fusion network* that captures non-linear correlations between social and temporal embeddings, to contextually model the co-variance of homophily and influence on predicting the set of all influenced users. We make the following contributions:

Generalizable Variational Autoencoder Framework: Unlike existing diffusion prediction methods that only consider temporal influence via induced propagation structures, INFVAE models homophily and distinguishes social roles of users towards diffusion prediction through a novel generalizable VAE framework that can be instantiated with a wide variety of GNN architectures of arbitrary model complexity.

Efficient Homophily and Influence Integration: To the best of our knowledge, ours is the first neural framework to exploit the co-variance of social homophily and temporal influence for predicting diffusion behaviors. Given a sequence of seed user activations, INFVAE employs an expressive *co-attentive fusion network* to jointly attend over their social and temporal representations to predict the set of all influenced users. INFVAE is faster than state-of-the-art recurrent methods by an order of magnitude.

Robust Experimental Results: Our experiments on multiple real-world social networks, including Digg, Weibo, and Stack-Exchanges, demonstrate significant gains for INFVAE over state-of-the-art models. Modeling social homophily through VAEs enables massive gains for users with *sparse activities*, and users who *lack direct social connections in seed sets*. An ablation analysis of various modeling choices further justifies the benefits of modeling the co-variance of social homophily and temporal influence.

We organize the rest of the chapter as follows. In Section 5.3, we formally define the problem of diffusion prediction in social networking platforms. In Section 5.4, we describe our proposed Influence Variational AutoEncoder framework (INFVAE). We present experimental results in Section 5.5, and finally conclude in Section 5.7.

5.2 RELATED WORK

We discuss existing work on diffusion modeling followed by related work on network representation learning, variational auto-encoders and co-attentions.

Information diffusion overview. Historically, information diffusion has been studied through two seminal models: Independent Cascade (IC) [70] and Linear Threshold (LT) [71]. Three distinct applications emerged, namely: *network inference* [218], which infers the underlying social network that best explains the observed cascades; *cascade prediction* [69], which predicts macroscopic properties of cascades, including size, growth, and shape; and *diffusion prediction* [81], which learns a model from social links and cascade sequences, to predict the set of influenced users given a seed set of activated users. In this work, we focus on diffusion prediction.

Diffusion prediction. The earliest data-driven methods propose several extensions of IC and LT incorporating topics [72], continuous timestamps [73], user profiles [74], and community structure [75]. A few techniques explore probabilistic generative models via latent topics and communities [76, 219]. Most recent studies focus on learning representations to overcome feature engineering or pre-defined hypotheses in diffusion modeling [77, 78, 79, 80, 81, 82, 217, 220]. Emb-IC [79], Inf2vec [80] embed user influencing capability and susceptibility in diffusion. Topo-LSTM [81], CYAN-RNN [217], SNIDSA [82], and DeepDiffuse [83] project the diffusion cascades on local social neighborhoods and model the resulting DAG propagation structures with RNNs. These techniques outperform classical approaches by significant margins in diffusion prediction. Our key observation is that these projected DAGs could ignore social structures that do not appear in any observed cascade. In contrast, our model INFVAE can account for unobserved social connections in the user

activation process by modeling social homophily through VAEs.

A related problem is social influence prediction, which aims to classify social media users based on the activation status of their ego-network [220, 221]. Direct extensions to predict the set of all influenced users (diffusion prediction) entails reapplying their models on each candidate inactive user in the social network, resulting in prohibitive inference costs, hence preventing a comparison.

Network representation learning: This line of work captures varied notions of structural node proximity [18, 222] in networks via low-dimensional vectors. Notably, graph neural networks have achieved great success in node classification and link prediction [39, 40, 41, 46, 58, 223, 224]. Graph Autoencoders [26, 27] employ various encoding and decoding architectures to embed network structure and learn unsupervised node embeddings. [28] unify a large family of network embedding methods in an autoencoder framework. However, general-purpose embeddings modeling structural proximity are not directly suited to diffusion modeling.

Co-attentional models: Our work also leverages recent advances in neural attention mechanisms, especially in Natural Language Processing [130]. Specifically, co-attention has achieved great success in modeling relationships between pairs of sequences, *e.g.*, question-answer [225], etc. Co-attentional methods compute interaction weights between data modalities, learning fine-grained non-linear correlations. In our work, we develop a co-attentive fusion network to capture the contextual interplay of users’ social and temporal representations for diffusion prediction.

5.3 PROBLEM DEFINITION

We study diffusion prediction in social networks where the goal is to predict the set of all influenced users, given temporally ordered seed user activations for user-generated content.

Definition 5.1. Social Network: The social network is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_i\}_{i=1}^N$ is the set of N users and $\mathcal{E} = \{e_{ij}\}_{i,j=1}^N$ is the set of links. We denote the adjacency matrix of \mathcal{G} by $A \in \mathbb{R}^{N \times N}$ where $A_{i,j} = 1$ if $e_{i,j} \in \mathcal{E}$ otherwise 0.

Definition 5.2. Diffusion cascade: A diffusion cascade D_i is an ordered sequence of user activations in ascending time order denoted by: $D_i = \{(v_{i_k}, t_k) \mid v_{i_k} \in \mathcal{V}, t_k \in [0, \infty), k = 1 \dots K\}$, each v_{i_k} is a distinct user in \mathcal{V} (no repeats) and t_k is non-decreasing, *i.e.*, $t_k \leq t_{k+1}$. The k^{th} user activation has a tuple (v_{i_k}, t_k) , referring the activated user and activation time.

We represent cascades by delay-agnostic relative activation orders similar to [79, 80, 81], *i.e.*, a cascade is equivalently written as $D = \{(v_{i_k}, k) \mid v_{i_k} \in \mathcal{V}\}_{k=1}^K$. We do not assume

Symbol	Description
\mathbf{Z}	Social variables modeling network proximity, for all users \mathcal{V}
\mathbf{V}_S	Sender variables for all users \mathcal{V}
\mathbf{V}_R	Receiver variables for all users \mathcal{V}
\mathbf{V}_T	Temporal influence variables for all users \mathcal{V}
\mathbf{V}_P	User-specific popularity variables for all users \mathcal{V}
\mathbf{P}_K	Position-encoded temporal embeddings for all time steps K

Table 5.1: *Notations*

the availability of explicit re-share links between users in cascades; this corresponds to the simplest yet most general setting of diffusion [80, 81]. Though timestamps may be easily used as input features, we leave generation of continuous timestamps as future work.

Definition 5.3. Diffusion prediction: Given a social network G and a collection of cascade sequences $\mathbb{D} = \{D_i, 1 \leq i \leq |\mathbb{D}|\}$, learn diffusion model M to predict the future set of influenced users in a cascade with seed activation sequence $I = \{(v_{i_1}, 1), \dots, (v_{i_k}, k)\}$ of k seed users. Diffusion prediction estimates the probability of influencing each inactive user: $P_{\Theta}(v \mid I) \forall v \in \mathcal{V} - I$, inducing a ranking of activation likelihoods over the inactive user set.

We create a training set \mathbb{T} of diffusion *episodes* containing (seed activations, activated users) tuples from the cascade collection \mathbb{D} , by randomly splitting each cascade $D \in \mathbb{D}$ of length K at each time step $2 \leq k \leq K - 1$. Specifically, a split at time step $k \geq 2$, creates a training episode (I_k, C_k) where $I_k = \{(v_{i_j}, j); 1 \leq j \leq k\}$ is the seed set consisting of the cascade sliced at k and $C_k = \{v_{i_{k+1}}, \dots, v_{i_K}\}$ is the set of influenced users after time step k . Thus, we denote the training set by $\mathbb{T} = \{(I_i, C_i) \mid 1 \leq i \leq |\mathbb{T}|\}$.

5.4 INFVAE FRAMEWORK

In this section, we describe our proposed Influence Variational Autoencoder (INFVAE) framework for predicting information diffusion in social networks. We first introduce the different interacting latent variables in our deep generative model, then describe the process of generating diffusion cascades and finally present model training details.

5.4.1 Generative Model Description

INFVAE is a neural latent variable model that jointly capture social homophily and temporal influence for diffusion prediction. We describe the latent variables modeling homophily

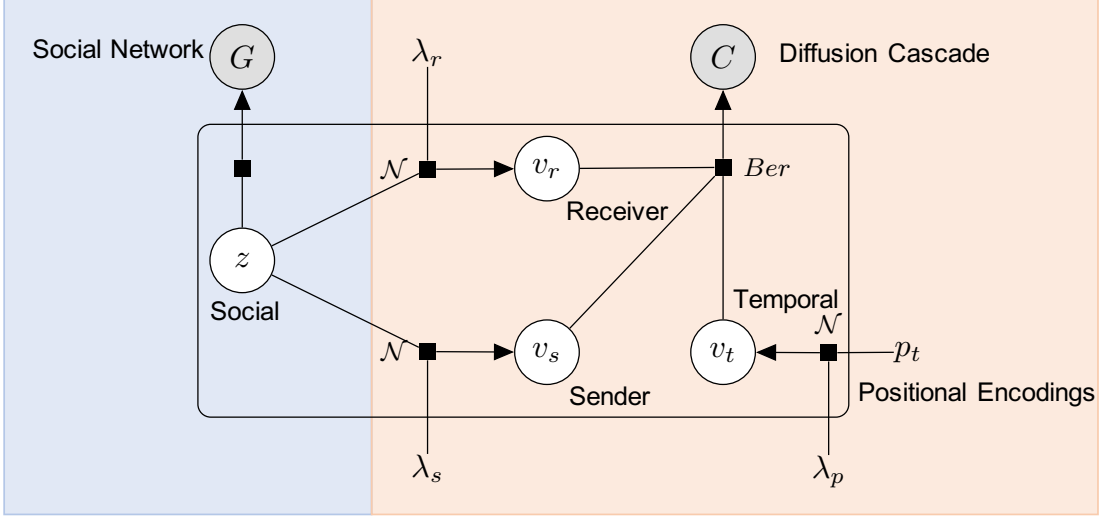


Figure 5.1: Plate Diagram of INFVAE depicting interactions between different latent variables: social, sender, receive and temporal variables.

and influence, followed by the structure of our generative network INFVAE.

Social Homophily.

Our objective is to define latent *social variables* for users that capture social homophily. The homophily principle stipulates that users with similar interests are more likely to be connected [11]. In the absence of explicit user attributes (*e.g.*, demographics or interests), we posit that highly interconnected users in social communities share homophilous relationships. We model social homophily through latent *social* variables designed to encourage users with shared social neighborhoods to have similar latent representations.

Specifically, we assign a latent *social* variable \mathbf{z}_i for user v_i , where the prior for \mathbf{z}_i is chosen to be a unit normal distribution, in line with standard assumptions in VAEs. Normal distributions are chosen in VAE frameworks due to their flexibility to support arbitrary functional parameterizations by isolating sampling stochasticity to facilitate back-propagation [135]. We assume the latent social variables \mathbf{Z} to collectively generate the social network \mathcal{G} , through a graph generation neural network $f_{\text{DEC}}(\mathbf{Z})$ parameterized by θ . The corresponding graph generative process is given by:

$$\mathbf{z}_i \sim \mathcal{N}(0, I_D) \quad \mathcal{G} \sim p_\theta(\mathcal{G} \mid \mathbf{Z}) = p_\theta(\mathcal{G} \mid f_{\text{DEC}}(\mathbf{Z})) \quad (5.1)$$

where $I_D \in \mathbb{R}^{D \times D}$ is an identity matrix of D dimensions. Here, the graph generation neural network $f_{\text{DEC}}(\mathbf{Z})$ can be instantiated to preserve an arbitrary notion of structural proximity

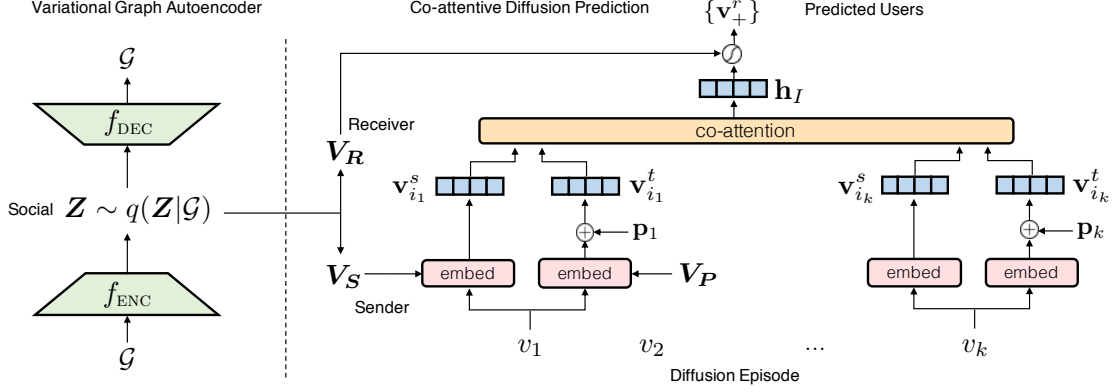


Figure 5.2: *Neural Architecture of INFVAE depicting latent variable interactions. The left side indicates the VAE framework to model social homophily; right side denotes the co-attentive fusion network to integrate the social and temporal variables.*

in the social network \mathcal{G} (Sec 5.4.3). In the above equation, we abuse the notation of \mathcal{G} to denote an appropriate representational form of the social network structure, which can take multiple forms, including the adjacency matrix, random walks sampled from \mathcal{G} , etc.

While homophily characterizes peer-to-peer interest similarity, its impact on diffusion behaviors of users is asymmetric since users who share interests may drastically differ in their posting rates; *e.g.*, certain users are naturally predisposed to be socially active and hence more *influential* in comparison to others. Thus, it is necessary to differentiate social user *roles* (influential versus susceptible) when modeling the impact of social homophily on diffusion behaviors. Similar concepts have been examined in social influence literature to characterize users by their influencing capability and conformity [78, 79, 80, 81, 226].

We associate each user $v_i \in \mathcal{V}$ with a *sender* $\mathbf{v}_i^s \in \mathbb{R}^D$ and *receiver* $\mathbf{v}_i^r \in \mathbb{R}^D$ latent variable. In contrast to prior VAEs over graphs [28] that preserve structural graph connectivity, our key innovation lies in conditioning the information sending and receiving capabilities of users on their homophilous traits through their social latent variables. We use normal distributions centered at \mathbf{z}_i to define the *sender* and *receiver* variables for user v_i as:

$$\mathbf{v}_i^s \sim \mathcal{N}(\mathbf{z}_i, \lambda_s^{-1} I_D) \quad \mathbf{v}_i^r \sim \mathcal{N}(\mathbf{z}_i, \lambda_r^{-1} I_D) \quad (5.2)$$

where λ_s and λ_r are tunable hyper-parameters controlling the degree of variation or uncertainty for \mathbf{v}_i^s and \mathbf{v}_i^r w.r.t. \mathbf{z}_i for user v_i . Let \mathbf{V}_S and \mathbf{V}_R denote the set of all sender and receiver variables respectively for all users in the social network, which are independently conditioned on the corresponding set of latent social variables \mathbf{Z} .

Temporal Influence.

Now, we define latent *temporal influence* variables to describe the varying influence effects of seed users depending on the relative sequential order of activations. There are two interesting factors at play: activation orders and popularity effects. A majority of social media users adopt more recent information while often ignoring old and obsolete content [227]. On the other hand, social status impacts the influencing power of seed users independent of their activation order and social neighbors, *e.g.*, famous media figures naturally exert significant influence. Thus, we consider both the relative sequential order of user activations and popularity effects of seed users to model temporal influence.

To quantify the temporal influence exerted by a seed user activation (v_{i_k}, k) of user v_{i_k} at time step k ($1 \leq k \leq K$), we first encode the relative position k through positional-encodings [228] to obtain temporal embeddings \mathbf{p}_k . Since we expect the variation in popularity effects to be quite small, we draw user-specific popularity variables from a zero-mean normal distribution to serve as offsets to the temporal embeddings. Specifically, the *temporal influence* variable for activation (v_{i_k}, k) denoted by $\mathbf{v}_{i_k}^t$, is given by:

$$\begin{aligned} \mathbf{v}_{i_k}^p &\sim \mathcal{N}(0, \lambda_p^{-1} I_D) \quad \mathbf{p}_k = PE(k) \quad \mathbf{v}_{i_k}^t = \mathbf{v}_{i_k}^p + \mathbf{p}_k \\ PE(k)_{2d} &= \sin(k/10000^{2d/D}) \quad PE(k)_{2d+1} = \cos(k/10000^{2d/D}) \end{aligned} \quad (5.3)$$

where λ_p is a hyper-parameter to control the popularity effects, and $1 \leq d \leq D/2$ denotes the dimension in the temporal embedding \mathbf{p}_k . Note that the popularity variable $\mathbf{v}_{i_k}^p$ is user-specific, while temporal embedding \mathbf{p}_k only depends on the activation step k . The set of all latent user popularity variables are denoted by \mathbf{V}_P , while \mathbf{P}_K represents the set of position-encoded *temporal* embeddings.

Co-attentive Diffusion Episode Generation.

Let us consider a single diffusion episode $(I, C) \in \mathbb{T}$, with initial seed user activations $I = \{(v_{i_1}, 1), \dots, (v_{i_k}, k)\}$ and influenced users $C = \{v_{i_{k+1}}, \dots, v_{i_K}\}$. A diffusion model aims to predict the set of influenced users C given seed activations I . Since information diffusion is always conditioned on the seed user I , we conditionally sample C given I .

Let us denote the set of seed users by $I_U = \{v_{i_1}, \dots, v_{i_k}\}$. Our objective is to model the covariance of social homophily and temporal influence exerted by seed users I_U , which can be summarized by: *sender variable sequence* $(\mathbf{v}_{i_1}^s, \mathbf{v}_{i_2}^s, \dots, \mathbf{v}_{i_K}^s)$; and *temporal influence variable*

sequence $(\mathbf{v}_{i_1}^t, \mathbf{v}_{i_2}^t, \dots, \mathbf{v}_{i_K}^t)$. To model complex correlations between the sender and temporal influence variable sequences, we propose an expressive *co-attentive* fusion strategy to learn attention scores for each seed user by modeling interactions between the two sequences. We describe the conditional generative process in two steps:

- The social homophily and temporal influence aspects of seed users, are integrated into an aggregate seed set representation \mathbf{h}_I . The co-attentive fusion network $G_{\text{DIFF}}(\cdot)$ performs homophily-guided temporal attention, *i.e.*, attends over the temporal influence variables by computing co-attentional weights that jointly depend on both homophily and temporal influence characteristics. As illustrated in Figure 5.2, the sender and temporal influence variables of seed users feed into a fusion network $G_{\text{DIFF}}(\mathbf{v}_{i_k}^s, \mathbf{v}_{i_k}^t)$. The aggregate seed representation \mathbf{h}_I is computed as:

$$\alpha_k = \frac{\exp(G_{\text{DIFF}}(\mathbf{v}_{i_k}^s, \mathbf{v}_{i_k}^t(k)))}{\sum_{j=1}^K \exp(G_{\text{DIFF}}(\mathbf{v}_{i_j}^s, \mathbf{v}_{i_j}^t(j)))} \quad \mathbf{h}_I = \sum_{j=1}^K \alpha_j \mathbf{v}_{i_j}^t(j) \quad (5.4)$$

Each α_j is the normalized co-attentional coefficient for seed user v_{i_k} denoting its contribution in computing the aggregate representation \mathbf{h}_I . To model the co-dependence between $\mathbf{v}_i^s, \mathbf{v}_i^t$, we define the co-attentive function $G_{\text{DIFF}}(\mathbf{v}_{i_k}^s, \mathbf{v}_{i_k}^t) = \tanh(\mathbf{v}_{i_k}^{sT} \mathbf{W} \mathbf{v}_{i_k}^t)$ as a bi-linear product parameterized by $\mathbf{W} \in \mathbb{R}^{D \times D}$.

- The probability of influencing an inactive user v_j depends on the sending capacity of seed users (embedded in \mathbf{h}_I) and her receiving capability (encoded by *receiver* variable \mathbf{v}_j^r). We quantify the likelihood of influencing v_j by the inner product $\mathbf{h}_I^T \mathbf{v}_j^r$. For each inactive user $v_j \in \mathcal{V} - I_U$, we draw a binary variable $C_j \in \{0, 1\}$ indicating whether user v_j is influenced by the seed users I_U or not, given by:

$$C_j \sim \text{Ber}(\sigma(\mathbf{h}_I^T \mathbf{v}_j^r)) \quad \forall v_j \in \mathcal{V} - \{v_{i_1}, \dots, v_{i_K}\} \quad (5.5)$$

where $\sigma(\cdot)$ is the sigmoid function and $\text{Ber}(\cdot)$ is the Bernoulli distribution. The corresponding logistic log-likelihood of generating a diffusion episode (I, C) is given by:

$$\begin{aligned} \mathcal{L}_{I,C}^{\text{DIFF}} &= \log p_{\theta}(C \mid I, V_S, V_R, V_P) \\ &= \sum_{v \in C} \eta \log(\sigma(\mathbf{h}_I^T \mathbf{v}_i^r)) + \sum_{v_n \in \mathcal{V} - C - I_U} \log(1 - \sigma(\mathbf{h}_I^T \mathbf{v}_n^r)) \end{aligned} \quad (5.6)$$

Here, the hyper-parameter η re-weights the observed positive examples since the actual number of influenced users is much smaller than the total number of users.

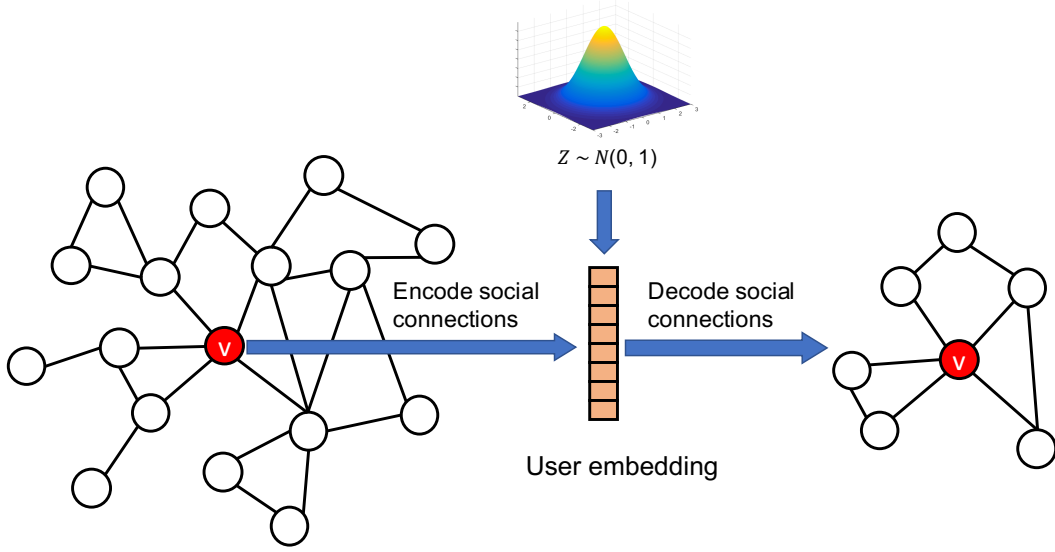


Figure 5.3: Variational Graph Encoder-Decoder framework showing the social latent variables in INFVAE that preserve structural proximity in the social network.

5.4.2 Model Likelihood

An analytical computation of the latent posterior distribution $p(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z} | \mathcal{G}, \mathbb{T})$ is intractable due to the learnable neural layers in the generative process. Thus, we use variational inference to factorize the latent posterior with a mean-field approximation as:

$$q(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z} | \mathcal{G}) = q(\mathbf{V}_S)q(\mathbf{V}_R)q(\mathbf{V}_P)q(\mathbf{Z} | \mathcal{G}) \quad (5.7)$$

The variational distributions of variables $\mathbf{V}_S, \mathbf{V}_R$, and \mathbf{V}_P follow normal distributions while the social variables \mathbf{Z} are conditioned on \mathcal{G} through a structure-encoding inference network [135]. Specifically, the variational distribution of \mathbf{Z} denoted by $q_\phi(\mathbf{Z} | \mathcal{G})$, is a diagonal normal distribution parameterized by $f_{\text{ENC}}(\mathcal{G})$ defined as:

$$f_{\text{ENC}}(\mathcal{G}) \equiv [\mu_\phi(\mathcal{G}), \log \sigma_\phi^2(\mathcal{G})] \quad q_\phi(\mathbf{Z} | \mathcal{G}) = \mathcal{N}(\mu_\phi(\mathcal{G}), \text{diag}(\sigma_\phi^2(\mathcal{G}))) \quad (5.8)$$

The inference network outputs the parameters, $\mu_\phi(\mathcal{G}), \sigma_\phi(\mathcal{G})$ of the variational distribution $q_\phi(\mathbf{Z} | \mathcal{G})$, which is designed to approximate the corresponding posterior $p(\mathbf{Z} | \mathcal{G})$. The inference network $f_{\text{ENC}}(\mathcal{G})$ endows the model with added flexibility to incorporate arbitrary neighborhood aggregation functions such as graph convolutions [39], attentions [40], etc. The variational structure distribution $q_\phi(\mathbf{Z} | \mathcal{G})$ and the structure generative model $p_\theta(\mathcal{G} | \mathbf{Z})$ (Eqn. 5.1) together constitutes a *variational graph autoencoder* [27].

5.4.3 Neural Graph Autoencoder Details

In this section, we describe functions $f_{\text{ENC}}(\mathcal{G})$ and $f_{\text{DEC}}(\mathbf{Z})$ which describe the graph structure inference and generative networks of INFVAE. The *encoder* summarizes local social neighborhoods into latent vectors, which are subsequently transformed by the *decoder* into high-dimensional structural information (*e.g.*, adjacency matrix). [28] present an encoder-decoder framework to conceptually unify a large family of graph embedding methods. Encoder architectures fall into three major categories: embedding lookups [17, 18], neighborhood vector encoding [26], and neighborhood aggregation [41], while decoders comprise unary and pairwise variants. In INFVAE, we explore two representative choices:

MLP + MLP: We use a Multi-Layer Perceptron (MLP) to both encode and decode the laplacian matrix of \mathcal{G} , given by $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. The neighborhood vector for user v_i , denoted by \mathbf{a}_i , is the i^{th} row of $\mathbf{L} = [\mathbf{a}_1, \dots, \mathbf{a}_N]^T$. The encoder is an MLP network $f_{\text{ENC}}(\mathbf{a}_i)$ which encodes \mathbf{a}_i into \mathbf{z}_i , while the decoder $f_{\text{DEC}}(\mathbf{z}_i)$ strives to reconstruct \mathbf{a}_i from \mathbf{z}_i . We introduce a re-weighting vector $\mathbf{b}_i = \{b_{ij}\}_{j=1}^N$ where $b_{ij} = 1$ if $L_{ij} = 0$ and $b_{ij} = \beta > 1$ when $L_{ij} > 0$. β is a confidence parameter that re-weights the positive terms ($L_{ij} > 0$) to balance the unobserved 0's which far outnumber the observed links in real-world networks. The generative process to obtain \mathbf{a}_i from \mathbf{z}_i is given by:

$$\mathbf{a}_i \sim p_{\theta}(\mathbf{a}_i|\mathbf{z}_i) = \mathcal{N}(f_{\text{DEC}}(\mathbf{z}_i), \text{diag}(\mathbf{b}_i)) \quad (5.9)$$

where $\text{diag}(\mathbf{b}_i)$ is a diagonal matrix with non-zero entries from vector \mathbf{b}_i . The corresponding Gaussian log-likelihood is given by:

$$\log p_{\theta}(\mathbf{A}|\mathbf{Z}) = \sum_{i=1}^N \log p_{\theta}(\mathbf{a}_i|\mathbf{z}_i) = \sum_{i=1}^N \|\mathbf{b}_i \odot (\mathbf{a}_i - f_{\text{DEC}}(\mathbf{z}_i))\|^2 \quad (5.10)$$

GCN + Inner Product: We use a Graph Convolutional Network (GCN) as the encoder and an inner product decoder that maps pairs of user embeddings to a binary indicator of link existence in \mathcal{G} . The GCN network comprises multiple stacked graph convolutional layers to extract features from higher-order structural neighborhoods. The input to a layer is a user feature (or embedding) matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ and a normalized adjacency matrix $\hat{\mathbf{A}}$, where each GCN layer computes the function:

$$f_{\text{ENC}}(\mathbf{A}) = \sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}) \quad \hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} + \mathbf{I}_N \quad (5.11)$$

Algorithm 5.1: INFVAE *training with block coordinate ascent.*

Input: Social Network (\mathcal{G}), Training episodes (\mathbb{T})

Output: MAP estimates of $\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P$ and parameters θ, ϕ .

- 1: Initialize latent variables from a standard normal distribution.
- 2: **Pre-training:** Train $f_{\text{DEC}}(\mathcal{G}|\mathbf{Z})$ and $f_{\text{ENC}}(\mathbf{Z}|\mathcal{G})$ on \mathcal{G} using a VAE with log-likelihood:

$$L^{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \log p_\theta(\mathcal{G}|\mathbf{Z}) - D_{\text{KL}}(q_\phi(\mathbf{Z}|\mathcal{G}), p(\mathbf{Z}))$$

- 3: **while not converged do**
 - ▷ **Optimize over social network \mathcal{G}**
 - 4: **for** each batch of users $\mathcal{U} \subseteq \mathcal{V}$ **do**
 - 5: Fix $\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, G_{\text{DIFF}}(\cdot)$ and update weights of $f_{\text{ENC}}(\mathcal{G})$ and $f_{\text{DEC}}(\mathbf{Z})$ using mini-batch gradient ascent (Equation 5.15)
 - ▷ **Optimize over diffusion episodes \mathbb{T}**
 - 6: **for** each batch of diffusion episodes $B \subseteq \mathbb{T}$ **do**
 - 7: Fix \mathbf{Z} , $f_{\text{ENC}}(\mathcal{G})$, $f_{\text{DEC}}(\mathbf{Z})$ and update $\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P$, and $G_{\text{DIFF}}(\cdot)$ using mini-batch gradient ascent. (Equation 5.15)
-

where \mathbf{X} is an identity matrix encoding user identities. Each entry A_{ij} of adjacency matrix \mathbf{A} is generated according to:

$$A_{ij} \sim p_\theta(A_{ij}|\mathbf{z}_i, \mathbf{z}_j) = \text{Ber}(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) \quad (5.12)$$

Similar to above, we re-weight the positive entries of \mathbf{A} with a confidence parameter β . The logistic log-likelihood is given by:

$$\log p_\theta(\mathbf{A}|\mathbf{Z}) = \sum_{(i,j) \in \mathcal{E}} \beta \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) + \sum_{(i,j) \notin \mathcal{E}} \log(1 - \sigma(\mathbf{z}_i^T \mathbf{z}_j)) \quad (5.13)$$

As an alternative to re-weighting positive entries, negative sampling [17] can scale this training objective to large-scale social networks.

5.4.4 Model Inference

The overall model training objective maximizes a lower bound on the marginal log likelihood, also named evidence lower bound (ELBO) [229], which is given by:

$$L_q = \mathbb{E}_q[\log p(\mathcal{G}, \mathbb{T}, \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}) - \log q(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}|\mathcal{G})] \quad (5.14)$$

Note that L_q is a function of both generative (θ) and variational (ϕ) parameters. However, an analytical computation of the expectation with respect to $q_\phi(\mathbf{Z}|\mathcal{G})$ is intractable, while Monte Carlo sampling prevents gradient back-propagation to the neural parameters of $f_{\text{ENC}}(\mathcal{G})$. With the reparametrization trick [135], we instead sample $\epsilon \sim \mathcal{N}(0, I_{N \times D})$ and form samples of $\mathbf{Z} = \mu_\phi(\mathcal{G}) + \epsilon \odot \sigma_\phi(\mathcal{G})$. This isolates the stochasticity during sampling and the gradient with respect to ϕ can be back-propagated through the sampled \mathbf{Z} .

Optimization

Since bayesian learning methods to infer latent posterior distributions incur high computational costs, and considering our goal of making good diffusion predictions rather than explanations, we resort to MAP (Maximum A Posteriori) estimation. Thus, we sample \mathbf{Z} from $q_\phi(\mathbf{Z}|\mathcal{G})$ using point estimates for $\mathbf{V}_S, \mathbf{V}_R$ and \mathbf{V}_P . We maximize the joint log-likelihood with MAP estimates of latent variables $\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P$, inference and generative network parameters θ, ϕ , and observations \mathbb{T} and \mathcal{G} , given hyper-parameters $\lambda_s, \lambda_r, \lambda_p$, as defined by:

$$\begin{aligned} \mathcal{L}^{\text{MAP}} = & \mathbb{E}_{q_\phi}[\log p_\theta(\mathcal{G}|\mathbf{Z})] - D_{\text{KL}}(q_\phi, p(\mathbf{Z})) + \sum_{(I,C) \in \mathbb{T}} \mathcal{L}_{I,C}^{\text{DIFF}} \\ & - \sum_{i=1}^N \left(\frac{\lambda_s}{2} \mathbb{E}_{q_\phi} \|\mathbf{v}_i^s - \mathbf{z}_i\|^2 + \frac{\lambda_r}{2} \mathbb{E}_{q_\phi} \|\mathbf{v}_i^r - \mathbf{z}_i\|^2 + \frac{\lambda_p}{2} \|\mathbf{v}_i^p\|^2 \right) \end{aligned} \quad (5.15)$$

where q_ϕ is a shorthand for the variational distribution of network structure generation $q_\phi(\mathbf{Z}|\mathcal{G})$, and $\mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})}[\mathbf{Z}]$ is equal to $\mu_\phi(\mathcal{G})$ output by the inference network. To optimize this objective, we employ block coordinate ascent with two sets of variables, $\{f_{\text{ENC}}(\mathcal{G}), f_{\text{DEC}}(\mathbf{Z})\}$ and $\{\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{G}_{\text{DIFF}}\}$. As illustrated in Alg 5.1, each iteration of the algorithm proceeds in two steps, by alternating optimization over the social network and diffusion cascades.

Diffusion Prediction

Our goal is to design a deep generative model towards diffusion prediction in a social network. Thus, after learning the (locally) optimal model parameters and MAP estimates of latent variables in our model INFVAE, we compute the likelihood of influencing a candidate user v_j given a sequence of seed activations I as:

$$p(v_j|I) = \sigma(h_I^T \mathbf{v}_j^r) \quad (5.16)$$

Complexity

The cost per iteration comprises two parts: (a) optimizing over social network \mathcal{G} gives $O(|\mathcal{E}| \cdot F^2 + |\mathcal{E}| \cdot D)$ assuming GCN + Inner Product (b) optimizing over diffusion episodes is $O(|\mathbb{T}| \cdot D \cdot N)$ where F is the maximum layer dimension in f_{ENC} . The overall complexity per iteration is $O(|\mathcal{E}| \cdot F^2 + |\mathcal{E}| \cdot D + |\mathbb{T}| \cdot D \cdot N)$.

5.4.5 Model Objective Derivation

In this section, we start from the model likelihood and derive the final objective function (Equation 9) by following the steps of variational inference and MAP estimation. The joint probability distribution of both the observed data and latent variables in our deep generative framework INFVAE is given by:

$$p(\mathcal{G}, \mathbb{T}, \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}) = p(\mathcal{G}, \mathbb{T} | \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}) p(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z} | \mathcal{G}, \mathbb{T}) \quad (5.17)$$

The conditional likelihood of generating the observed data $\{\mathcal{G}, \mathbb{T}\}$ given latent variables $\{\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}\}$ is given by:

$$p(\mathcal{G}, \mathbb{T} | \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}) = p(\mathcal{G} | \mathbf{Z}) p(\mathbb{T} | \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P) \quad (5.18)$$

$$= p_\theta(\mathcal{G} | \mathbf{Z}) \times \prod_{(I, C) \in \mathbb{T}} p_\theta(C | \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, I) \quad (5.19)$$

The latent posterior distribution, given by $p(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z} | \mathcal{G}, \mathbb{T})$, can be factorized as:

$$p(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z} | \mathcal{G}, \mathbb{T}) = \prod_{i=1}^N p(\mathbf{v}_i^S | \mathbf{z}_i) p(\mathbf{v}_i^R | \mathbf{z}_i) p(\mathbf{v}_i^P) p(\mathbf{Z} | \mathcal{G}) \quad (5.20)$$

Due to the intractability of computing an analytical form for the posterior, we use variational inference to factorize the posterior with a mean-field approximation:

$$\begin{aligned} q(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z} | \mathcal{G}) &= q(\mathbf{V}_S) q(\mathbf{V}_R) q(\mathbf{V}_P) q(\mathbf{Z} | \mathcal{G}) \\ &= \prod_{i=1}^N (q(\mathbf{v}_i^S | \phi_i^S) q(\mathbf{v}_i^R | \phi_i^R) q(\mathbf{v}_i^P | \phi_i^P)) q_\phi(\mathbf{Z} | \mathcal{G}) \end{aligned} \quad (5.21)$$

where the variational distributions of latent variables $\mathbf{V}_S, \mathbf{V}_R$, and \mathbf{V}_P follow normal distributions, parameterized by θ_i^S, θ_i^R , and $\theta_i^P \forall 1 \leq i \leq N$ respectively. The variational distribu-

tion of \mathbf{Z} , while being conditioned on \mathcal{G} is modeled through a normal distribution whose parameters are given by an inference network $f_{\text{ENC}}(\mathcal{G})$. Following the standard conventions of variational inference, the objective function to optimize, is given by the variation lower bound or evidence lower bound (ELBO):

$$L_q = \mathbb{E}_q[\log p(\mathcal{G}, \mathbb{T}, \mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}) - \log q(\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P, \mathbf{Z}|\mathcal{G})] \quad (5.22)$$

$$= \mathbb{E}_q[\log p_\theta(\mathcal{G}|\mathbf{Z}) + \log p_\theta(\mathbb{T}|\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P) + \log p(\mathbf{V}_S|\mathbf{Z}) + \log p(\mathbf{V}_R|\mathbf{Z}) + \log p(\mathbf{V}_P) + \log p(\mathbf{Z})] \quad (5.23)$$

$$- \mathbb{E}_q[\log q(\mathbf{V}_S) + \log q(\mathbf{V}_R) + \log q(\mathbf{V}_P) + \log q(\mathbf{Z})] \quad (5.24)$$

By factorizing the joint variational distribution into separate components, we obtain:

$$L_q = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})}[\log p_\theta(\mathcal{G}|\mathbf{Z})] + \mathbb{E}_{q(\mathbf{V}_S)q(\mathbf{V}_R)q(\mathbf{V}_P)}[\log p_\theta(\mathbb{T}|\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P)] \quad (5.25)$$

$$+ \mathbb{E}_{q(\mathbf{V}_S)q_\phi(\mathbf{Z}|\mathcal{G})} \log p(\mathbf{V}_S|\mathbf{Z}) + \mathbb{E}_{q(\mathbf{V}_R)q_\phi(\mathbf{Z}|\mathcal{G})} \log p(\mathbf{V}_R|\mathbf{Z}) \quad (5.26)$$

$$+ \mathbb{E}_{q(\mathbf{V}_P)} \log p(\mathbf{V}_P) + \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \log p(\mathbf{Z}) - \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \log q(\mathbf{Z}) \quad (5.27)$$

$$- [\mathbb{E}_{q(\mathbf{V}_S)} \log q(\mathbf{V}_S) + \mathbb{E}_{q(\mathbf{V}_R)} \log q(\mathbf{V}_R) + \mathbb{E}_{q(\mathbf{V}_P)} \log q(\mathbf{V}_P)] \quad (5.28)$$

Instead of conducting full variational inference to obtain the variational distributions of each latent variable, we instead use MAP estimates for the variational distributions $q(\mathbf{V}_S)$, $q(\mathbf{V}_R)$, and $q(\mathbf{V}_P)$ of latent variables \mathbf{V}_S , \mathbf{V}_R , and \mathbf{V}_P . Specifically, we denote the point estimates by $\{\mathbf{v}_i^s, \mathbf{v}_i^r, \mathbf{v}_i^p \mid \forall 1 \leq i \leq N\}$, which are the means of the normal distributions parameterized by $\{\theta_i^s, \theta_i^r, \theta_i^p \mid \forall 1 \leq i \leq N\}$ respectively. However, we retain the variational distribution $q(\mathbf{Z})$ for \mathbf{Z} since it is sampled from $\mathcal{N}(0, I_{N \times D})$ and $f_{\text{ENC}}(\mathcal{G})$ through the reparameterization trick. Using these assumptions, we can simplify each of the terms in the above equation as follows:

$$\mathcal{L}^{MAP} = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})}[\log p_\theta(\mathcal{G}|\mathbf{Z})] + \sum_{(I,C) \in \mathbb{T}} \mathcal{L}_{I,C}^{\text{DIFF}} \quad (5.29)$$

$$+ \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \log p(\mathbf{v}_i^s|\mathbf{z}_i) + \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \log p(\mathbf{v}_i^r|\mathbf{z}_i) \quad (5.30)$$

$$+ \sum_{i=1}^N \log p(\mathbf{v}_i^p) + \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \log \frac{p(\mathbf{Z})}{q(\mathbf{Z})} \quad (5.31)$$

Here, it can be easily seen that $\mathbb{E}_{q(\mathbf{V}_S)q(\mathbf{V}_R)q(\mathbf{V}_P)}[\log p_\theta(\mathbb{T}|\mathbf{V}_S, \mathbf{V}_R, \mathbf{V}_P)]$ reduces to $\mathcal{L}_{I,C}^{\text{DIFF}}$ under point-estimates for \mathbf{V}_S , \mathbf{V}_R and \mathbf{V}_P . A key point to note is the absence of three terms:

$\mathbb{E}_{q(\mathbf{V}_S)} \log q(\mathbf{V}_S)$, $\mathbb{E}_{q(\mathbf{V}_R)} \log q(\mathbf{V}_R)$ and $\mathbb{E}_{q(\mathbf{V}_P)} \log q(\mathbf{V}_P)$. These expectations integrate to 1, since the respective variational distributions reduce to point-estimates during MAP estimation. On substituting the probability densities of remaining probability distributions, we get our final objective as:

$$\mathcal{L}^{MAP} = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} [\log p_\theta(\mathcal{G}|\mathbf{Z})] + \sum_{(I,C) \in \mathbb{T}} \mathcal{L}_{I,C}^{\text{DIFF}} \quad (5.32)$$

$$- \frac{\lambda_s}{2} \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \|\mathbf{v}_i^s - \mathbf{z}_i\|^2 - \frac{\lambda_r}{2} \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \|\mathbf{v}_i^r - \mathbf{z}_i\|^2 \quad (5.33)$$

$$- \frac{\lambda_p}{2} \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{Z}|\mathcal{G})} \|\mathbf{v}_i^p\|^2 - D_{\text{KL}}(q_\phi(\mathbf{Z}|\mathcal{G}), p(\mathbf{Z})) \quad (5.34)$$

where $D_{\text{KL}}(x, y)$ refers to the KL-divergence between probability distributions x and y .

5.5 EXPERIMENTS

In this section, we present our experimental results on multiple datasets from real-world social networks and public Stack-Exchanges¹. We examine two popular social networking platforms, *Digg* and *Weibo* and three stack-exchange networks, *Android*, *Christianity* and *Travel*, to demonstrate the effectiveness of INFVAE for diffusion prediction.

- **Digg** [230]: A social news aggregation platform where users vote on news stories. The sequence of votes on each news story constitutes a diffusion cascade, while the social network comprises friendship links among voters. We retain only users who have voted on at least 40 stories.
- **Weibo** [221]: A Chinese micro-blogging platform, where the social network consists of follower links, and cascades reflect re-tweeting behavior of users. We use the posts and social connections of the 5000 most popular users in our experiments.

Stack-Exchanges: Community Q&A websites where users post questions and answers on a wide range of topics. The inter-user knowledge-exchanges on various interaction channels (*e.g.*, question, answer, comment, upvote, etc.), constitute the social network. Cascades correspond to chronologically ordered series of posts associated with the same tag, *e.g.*, “google-pixel-2” on Android. We choose three Stack-Exchanges, Android, Christianity and Travel, spanning diverse themes. Dataset statistics are provided in Table 5.2.

¹<https://archive.org/details/stackexchange>

Dataset	Social Networks		Stack-Exchange Networks		
	Digg	Weibo	Android	Christianity	Travel
# Users	8,602	5,000	9,958	2,897	8,726
# Links	173,489	123,691	48,573	35,624	76,555
# Cascades	968	23,475	679	589	711
Avg. cascade len	100.0	23.6	33.3	22.9	26.8

Table 5.2: *Statistics of datasets used in our experiments*

5.5.1 Baselines

We compare INFVAE against state-of-the-art representation learning methods for diffusion prediction since they have been shown to significantly outperform classical approaches (*e.g.*, IC and LT) and probabilistic generative models [80, 81].

- **CDK** [77]: an embedding method that models information spread as a heat diffusion process in the representation space of users.
- **Emb-IC** [79]: an embedded cascade model that generalizes IC to learn user representations from partial orders of user activations.
- **Inf2vec** [80]: an influence embedding method that combines local propagation structure and user co-occurrence in cascades.
- **DeepDiffuse** [83]: an attention-based RNN that operates on just the sequence of previously influenced users, to predict diffusion.
- **CYAN-RNN** [217]: a sequence-based RNN that uses an attention mechanism to capture cross-dependence among seed users.
- **SNIDSA** [82]: an RNN-based model to compute structure attention over the local propagation structure of a cascade.
- **Topo-LSTM** [81]: a recurrent model that exploits the local propagation structure of a cascade through a dynamic DAG-LSTM.

5.5.2 Experimental Setup

We denote our two model variants with GCN and MLP architectures, by INFVAE+GCN and INFVAE+MLP respectively. We randomly sample 70% of the cascades for training,

10% for validation and remaining 20% for testing. We consider the task of predicting the set of all influenced users as a retrieval problem [79, 80, 81, 217]. The fraction of users sampled from each test cascade to serve as the seed set is defined as **seed set percentage**, which is varied from 10% to 50% to create a large evaluation test-bed spanning diverse cascade lengths. The likelihood of influencing an inactive user determines its rank (Equation 5.16). We use MAP@ K (Mean Average Precision) and Recall@ K as evaluation metrics. Note that MAP@ K considers both the *existence* and *position* of ground-truth target users in the rank list, while Recall@ K only reports occurrence within top- K ranks.

Hyper-parameters are tuned by evaluating MAP@10 on the validation set. Since Emb-IC generalizes IC, we use 1000 Monte Carlo simulations to estimate influence probabilities. Since the recurrent neural models (*e.g.*, Topo-LSTM) are trained for next user prediction, we use the ranking induced by user activation probabilities for diffusion prediction, which we found to significantly outperform a similar simulation approach. For Inf2vec, we examine several seed influence aggregation functions (Ave, Sum, Max, and Latest) to report the best results. Our reported results are averaged over 10 independent runs with different random weight initializations. Our implementation of INFVAE is publicly available².

5.5.3 Evaluation Metric Definitions

We consider the task of predicting the set of all influenced users as a retrieval problem due to the large number of potential targets [79, 80, 81, 217]. The likelihood of influencing a user is used as a ranking score for evaluation (Equation 5.16). We use MAP@ K (Mean Average Precision) as our primary metric to evaluate diffusion prediction performance.

Mean Average Precision (MAP): Average precision (AP) is a ranked precision metric that gives larger credit to correctly predicted users in top ranks. Given a ranked list with K user predictions, $AP@K$ is defined as:

$$AP@K = \frac{\sum_{k=1}^K P(k) \times rel(k)}{\# \text{ Influenced Users}} \quad (5.35)$$

where $P(k)$ is the precision at cut-off k in the top- K list, and $rel(k)$ is an indicator function equaling 1 if the user at rank k has been influenced, otherwise 0. Finally, $MAP@K$ is defined as the mean of the AP scores over all diffusion episodes.

Recall: Recall measures the fraction of correctly predicted influenced users within the

²<https://github.com/aravindsankar28/Inf-VAE>

Method	Digg			Weibo		
MAP	@10	@50	@100	@10	@50	@100
CDK	0.0437	0.0222	0.0228	0.0130	0.0106	0.0123
Emb-IC	0.0862	0.0431	0.0431	0.0140	0.0116	0.0131
Inf2vec	0.1189	0.0554	0.0546	0.0156	0.0103	0.0121
DeepDiffuse	0.0919	0.0460	0.0471	0.0291	0.0186	0.0213
CYAN-RNN	0.1188	0.0479	0.0427	0.0296	0.0207	0.0234
SNIDSA	0.0941	0.0363	0.0348	0.0224	0.0146	0.0169
Topo-LSTM	0.1193	0.0577	0.0587	0.0325	0.0226	0.0247
Inf-VAE+MLP	0.1587	0.0774	0.0719	0.0322	0.0211	0.0234
Inf-VAE+GCN	0.1642	0.0779	0.0724	0.0373	0.0230	0.0257

Table 5.3: *Experimental results for diffusion prediction on two social network datasets (MAP@K scores for K = 10, 50 and 100), the seed set percentage varies in the range to 10 to 50% users in each test cascade. INFVAE achieves 26% relative gains in MAP@10 (on average) over the best baseline*

top- K ranks.

$$\text{Recall@K} = \frac{\sum_{k=1}^K \text{rel}(k)}{\# \text{Influenced Users}} \quad (5.36)$$

5.5.4 Experimental Results

We note the following key observations from our experimental results comparing INFVAE against competing baselines (Tables 5.3 and 5.4).

Methods that do not explicitly model sequential activation orders (*e.g.*, CDK and Emb-IC), perform markedly worse than their counterparts. Modeling local projected cascade structures with neural recurrent models results in improvements (*e.g.*, Topo-LSTM and others). Jointly modeling social homophily derived from global network structure and temporal influence by our model INFVAE yields significant relative gains of 22% (**MAP@10**) on average across all datasets. INFVAE+GCN consistently beats the MLP variant, validating the power of graph convolutional networks in effectively propagating higher-order local neighborhood features.

Figure 5.4 depicts the variation in recall with size of rank list K . As expected, recall increases with K , however, the relative differences across methods is much smaller. INFVAE consistently outperforms baselines across a wide range of K values. For instance, the

Method	Android			Christianity			Travel		
MAP	@10	@50	@100	@10	@50	@100	@10	@50	@100
CDK	0.0319	0.0121	0.0125	0.0876	0.0531	0.0578	0.0650	0.0333	0.0341
Emb-IC	0.0505	0.0248	0.0267	0.1340	0.0905	0.0962	0.0924	0.0584	0.0609
Inf2vec	0.0412	0.0141	0.0150	0.1824	0.0790	0.0852	0.1245	0.0495	0.0529
DeepDiffuse	0.0437	0.0228	0.0250	0.1632	0.0828	0.0831	0.1220	0.0675	0.0693
CYAN-RNN	0.0520	0.0276	0.0296	0.1971	0.1229	0.1304	0.1551	0.0791	0.0799
SNIDSA	0.0397	0.0207	0.0222	0.1233	0.0699	0.0781	0.0857	0.0562	0.0585
Topo-LSTM	0.0595	0.0283	0.0289	0.1811	0.0989	0.0991	0.1393	0.0773	0.0783
Inf-VAE+MLP	0.0584	0.0272	0.0285	0.2549	0.1355	0.1402	0.1865	0.0897	0.0913
Inf-VAE+GCN	0.0601	0.0290	0.0304	0.2594	0.1413	0.1461	0.1924	0.0906	0.0910

Table 5.4: Experimental results for diffusion prediction on three stock-exchange datasets (MAP@K scores for $K = 10, 50$ and 100), the seed set percentage varies in the range to 10 to 50% users in each test cascade. INFVAE achieves 16% relative gains in MAP@10 (on average). over the best baseline.

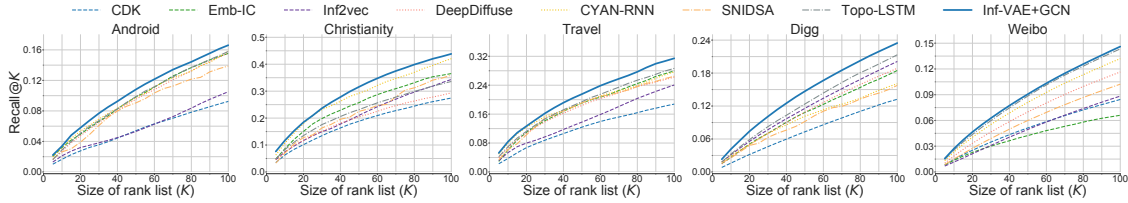


Figure 5.4: Experimental results for diffusion prediction on 5 datasets, Recall@K scores on varying size of the rank list K

Christianity dataset has seed sets with 2-10 users, and corresponding target sets with 10-15 users out of a possible 3000. Here, a recall@100 of 0.45 for INFVAE is quite impressive, especially considering the absence of explicit re-share links and the noise associated with real-world diffusion processes. We restrict our remaining analyses to INFVAE+GCN since it consistently beats the MLP variant.

5.5.5 Impact of Social and Interaction Sparsity

In this section, we analyze the benefits of explicitly modeling social homophily through VAEs, compared to the best baseline (Topo-LSTM) that only considers local propagation structures.

Users with sparse diffusion activities. We divide users into quartiles by their *activity levels*, which is the number of participating cascades per user. We evaluate *target recall*@100 for each user u , defined as the fraction of times u was predicted correctly

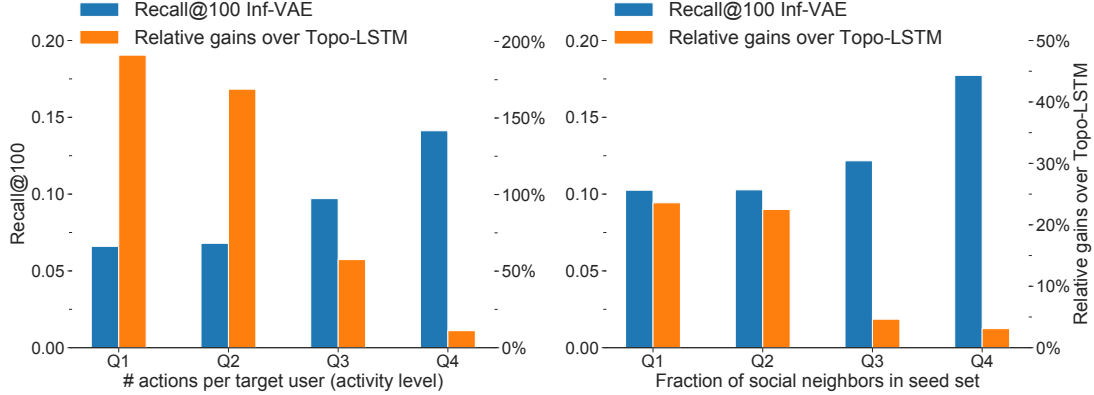


Figure 5.5: Performance across user quartiles on diffusion activity level, and seed neighbor fraction (*Q1*: lowest, *Q4*: highest). INFVAE has higher gains for users with sparse activities and lacking direct neighbors in seed sets (quartiles *Q1*-*Q3*).

within top-100 ranks. In Figure 5.5(a), we depict both recall scores and relative performance gains of INFVAE (over Topo-LSTM) across diffusion activity quartiles.

While target recall increases with activity levels, INFVAE significantly improves performance for inactive users (quartiles *Q1*-*Q3*). Thus, modeling social homophily through VAEs contributes to massive gains for users with *sparse diffusion activities*. Interestingly, Topo-LSTM performs comparably on the most active users (quartile *Q4*), which indicates the potential of pure sequential modeling techniques for highly active users.

Users that lack direct social connections in seed sets. We separate users into quartiles by *seed neighbor fraction*, which is computed as the fraction of seed users that are direct social neighbors, averaged over the training examples. We similarly report target recall@*K* and relative gains across quartiles (Figure 5.5(b)).

As expected, performance increases with seed neighbor fraction. Note higher relative gains over Topo-LSTM for users that lack direct neighbors in the seed set (quartiles *Q1*-*Q3*). This demonstrates the ability of INFVAE to implicitly regularize seed user representations based on higher-order social neighborhoods captured by GCN-based autoencoders. Again, we find that local sequential models suffice for users with large seed neighbor fractions, as evidenced by the results of Topo-LSTM in quartile *Q4*.

5.5.6 Model Analysis

In this section, we first present an *ablation study* on the architectural design choices in INFVAE, followed by a sensitivity analysis on *seed set percentage* and *hyper-parameters*.

Ablation Study

We analyze model design choices including modeling social homophily via VAEs and co-attention to capture co-variance of homophily with influence, on Android and Weibo datasets.

Social Homophily: We examine ways to condition the sender \mathbf{V}_S and receiver \mathbf{V}_R variables on \mathbf{Z} :

1. \mathbf{V}_S and \mathbf{V}_R are identical and are conditioned on \mathbf{Z} through hyper-parameter $\lambda_s(= \lambda_r)$, *i.e.*, $\mathbf{V}_S = \mathbf{V}_R \not\perp \mathbf{Z}$ (note that this is different from setting $\lambda_s = \lambda_r$ without enforcing $\mathbf{V}_S = \mathbf{V}_R$).
2. \mathbf{V}_S is a free variable conditionally independent of \mathbf{Z} , *i.e.*, $\mathbf{V}_S \perp \mathbf{Z}$, which is equivalent to setting $\lambda_s = 0$.
3. \mathbf{V}_R is a free variable, *i.e.*, $\mathbf{V}_R \perp \mathbf{Z}$, which is the inverse of (3).
4. \mathbf{V}_S and \mathbf{V}_R are both free variables conditionally independent of \mathbf{Z} ($\lambda_s = \lambda_r = 0$), *i.e.*, $\mathbf{V}_S \perp \mathbf{Z}, \mathbf{V}_R \perp \mathbf{Z}$.

Independent conditioning of \mathbf{V}_S and \mathbf{V}_R on \mathbf{Z} (default) achieves best results. Enforcing $\mathbf{V}_S = \mathbf{V}_R$ (row 1) is clearly inferior, which validates the choice of differentiating user roles. Notably, allowing \mathbf{V}_S to be a free variable results in minor performance degradation (row 2), while the drop is significant when \mathbf{V}_R is independent of \mathbf{Z} (row 3). As expected, setting both \mathbf{V}_S and \mathbf{V}_R as free variables (row 4), performs the worst due to lack of social homophily signals.

Metric	Weibo			Android		
	@10	@50	@100	@10	@50	@100
(0) Default	0.0373	0.0230	0.0257	0.0601	0.0290	0.0304
(1) $\mathbf{V}_S = \mathbf{V}_R \not\perp \mathbf{Z}$	0.0353	0.0220	0.0248	0.0558	0.0275	0.0287
(2) $\mathbf{V}_S \perp \mathbf{Z}$	0.0351	0.0213	0.0240	0.0595	0.0285	0.0301
(3) $\mathbf{V}_R \perp \mathbf{Z}$	0.0326	0.0217	0.0241	0.0567	0.0276	0.0291
(4) $\mathbf{V}_S \perp \mathbf{Z}, \mathbf{V}_R \perp \mathbf{Z}$	0.0313	0.0205	0.0235	0.0542	0.0274	0.0289
(5) Remove Coattention	0.0307	0.0207	0.0233	0.0553	0.0270	0.0284
(6) Separate Attentions	0.0293	0.0217	0.0192	0.0570	0.0277	0.0291
(7) Static-Pretrain	0.0342	0.0203	0.0226	0.0606	0.0281	0.0292

Table 5.5: Ablation study on architecture design ($MAP@K$ scores for $K = 10, 50, 100$), \perp denotes variable independence

Co-attention: We conduct two ablation studies defined by:

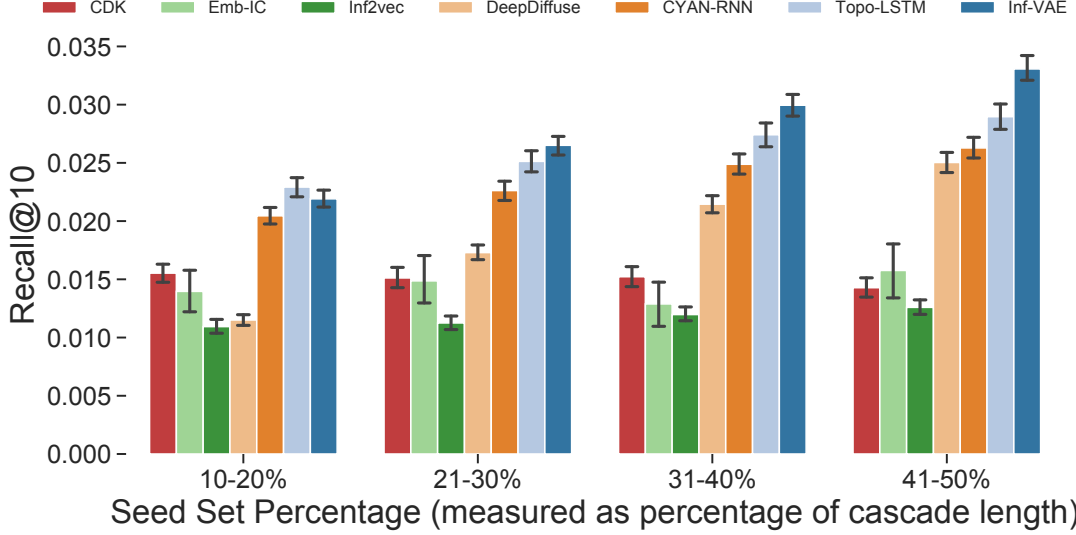


Figure 5.6: *Impact of seed set percentage in Weibo. INFVAE achieves higher gains for larger seed set fractions.*

5. Replace co-attention with meanpool over concatenated sender and temporal influence vectors, followed by a dense layer.
6. Replace co-attention with two separate attentions on the sender and temporal influence sequences, followed by concatenation.

Learning co-attentional weights (default) consistently outperforms mean pooling (5), illustrating the benefits of assigning variable contributions to seed users. Using separate attentions (6) significantly deteriorates results, which indicates the existence of complex non-linear correlations between the social and temporal latent factors.

Joint Training: In (2), we replace joint block-coordinate optimization (Alg 3.1) with a single step over cascades with pre-trained user embeddings (line 2), *i.e.*, \mathbf{Z} is not updated based on cascades.

Joint training is beneficial when social interactions are noisy (*e.g.*, Weibo) in comparison to focused stack-exchanges such as Android.

Impact of Seed Set Percentage.

We divide the test set into quartiles based on *seed set percentage*, and report performance per quartile. Since we require a sizable number of test examples per quartile to obtain unbiased estimates, we use the Weibo dataset.

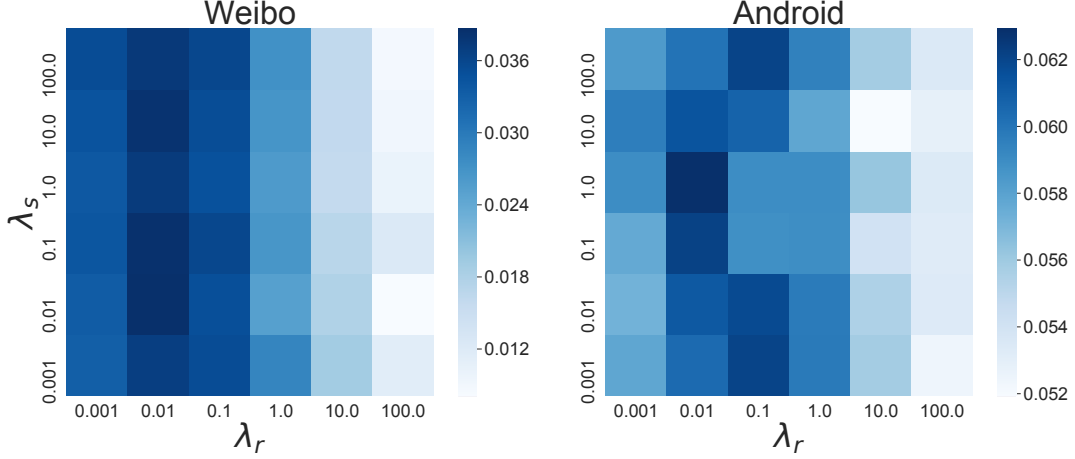


Figure 5.7: *MAP@10 on varying λ_s , λ_r over Android and Weibo. Performance is more sensitive to variations in λ_r than λ_s .*

Figure 5.6 depicts Recall@10 scores in different ranges. First, recall scores increase with seed set percentage since larger seed sets enable better model predictions; and target set size reduces with increase in seed set percentage. Second, relative gains of INFVAE over baselines increase with seed set percentage. This highlights the capability of co-attention in focusing on relevant users based on both social homophily and temporal influence factors.

Impact of λ_s and λ_r

Hyper-parameters λ_s and λ_r control the degree of dependence of the sender and receiver variables V_S, V_R on the social variables Z . Figure 5.7 depicts performance ($MAP@10$) on Android and Weibo datasets. The performance is sensitive to variations in λ_r with best values around 0.01 and 0.1, while λ_s results in minimal variations. Furthermore, the best values of λ_s, λ_r are stable in a broad range of values that transfer across datasets, indicating that INFVAE requires minimal tuning in practice. Since λ_p has minimal performance impact, we exclude it from our analysis.

Runtime Analysis.

In our empirical runtime analysis experiments, all diffusion prediction methods converge within 50 epochs with similar convergence rates. Thus, for the sake of brevity, we only compare model runtime per training epoch, which includes one optimization step over the social network and diffusion cascades for our model INFVAE.

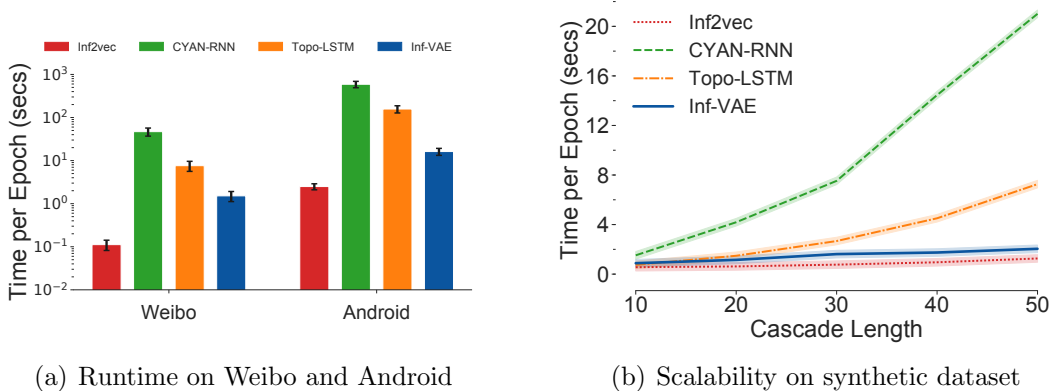


Figure 5.8: *Running time and scalability comparison of INFVAE with several baselines. INFVAE is faster than recurrent models (Topo-LSTM, CYANRNN) by an order of magnitude.*

From figure 5.8(a), Inf2vec is the fastest while our model INFVAE comes second. Thus, INFVAE achieves a good trade-off between expensive recurrent neural network models (*e.g.*, Topo-LSTM) and simpler embedding methods (*e.g.*, Inf2vec), while achieving consistently superior performance over all of them.

Scalability Analysis.

We analyze scalability on cascade sequences of varying lengths. Since real-world datasets possess heavily biased length distributions, we instead synthetically generate a Barabasi-Albert [1] network of 2000 users and simulate diffusion cascades using an IC model. We compare training times per epoch for each cascade length (l) in the range of 10 to 50.

Figure 5.8(b) depicts linear scaling for INFVAE and Inf2vec *wrt* cascade length. Recurrent methods scale poorly due to the sequential nature of back-propagation through time (BPTT), resulting in prohibitive costs for long cascade sequences. On the other hand, INFVAE avoids BPTT through efficient parallelizable co-attentions.

5.6 ACKNOWLEDGEMENTS

Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreements No. W911NF-17-C-0099 and FA8750-19-2-1004, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, and DTRA HDTRA11810026.

5.7 CONCLUSION

In this chapter, we present a novel variational autoencoder framework (INFVAE) to jointly embed homophily and influence in diffusion prediction. Given a sequence of seed user activations, INFVAE employs an expressive co-attentive fusion mechanism to jointly attend over their social and temporal variables, capturing complex correlations. Our experimental results on two social networks and three stock-exchanges indicate significant gains over state-of-the-art methods.

In future, INFVAE can be extended to include multi-faceted user attributes owing to the generalizable nature of our VAE framework. While the current implementation employs GCN networks, we foresee direct extensions with neighborhood sampling [41] to enable scalability to social networks with millions of users. We also plan to explore neural point processes to predict user activation times. Finally, similar frameworks may be examined for joint temporal co-evolution of social network and diffusion cascades.

In this chapter, we have examined our first inductive learning application that models the spread (or diffusion) of user-generated content in social networking platforms. Here, our proposed framework effectively overcomes interaction sparsity challenges for the vast population of users who seldom post content (sparse diffusion actions). In the next chapter, we explore a new social interaction setting in modern social platforms, involving group interactions; we focus on personalized item recommendations for ephemeral groups with limited or no historical interactions together.

CHAPTER 6: GROUPIM: SELF-SUPERVISED EPHEMERAL GROUP INTERACTION MODELING

6.1 INTRODUCTION

We address the problem of recommending items to *ephemeral* groups, which comprise users who purchase *very few (or no)* items together [231]. The problem is ubiquitous, and appears in a variety of familiar contexts, *e.g.*, dining with strangers, watching movies with new friends, and attending social events. We illustrate key challenges with an example: Alice (who loves Mexican food) is taking a visitor Bob (who loves Italian food) to lunch along with her colleagues, where will they go to lunch? There are three things to note here: first, the group is *ephemeral*, since there is *no* historical interaction observed for this group. Second, *individual preferences may depend on other group members*. In this case, the group may go to a fine-dining Italian restaurant. However, when Alice is with other friends, they may go to Mexican restaurants. Third, groups comprise users with *diverse individual preferences*, and thus the group recommender needs to be cognizant of individual preferences.

Prior work primarily target *persistent* groups which refer to fixed, stable groups where members have interacted with numerous items together *as a group* (*e.g.*, families watching movies). They mainly fall into two categories: heuristic pre-defined aggregation (*e.g.*, least misery [122]) that disregards group interactions; data-driven strategies such as probabilistic models [123, 129] and neural preference aggregators [131, 132]. A key weakness is that these methods either ignore individual user activities [132, 133] or assume that users have the same likelihood to follow individual and collective preferences, across different groups [123, 129, 131]. Lack of expressive power to distinguish the role of individual preferences across groups results in degenerate solutions for sparse ephemeral groups. A few related methods exploit external side information in the form of a social network [232, 233, 234], user personality traits [235] and demographics [236], for group decision making. However, such auxiliary side information may often be unavailable.

We train robust ephemeral group recommenders without resorting to any extra side information. Two observations help: first, while groups are ephemeral, group members may have rich individual interaction histories; this can alleviate group interaction sparsity. Second, since groups are ephemeral with sparse training interactions, base group recommenders need reliable guidance to learn informative (non-degenerate) group representations, but the guidance needs to be data-driven and learnable, rather than a heuristic.

Present Work: To overcome group interaction sparsity, we propose two key insights: First, we propose a self-supervised learning approach to utilize the intrinsic structure in

observed group interactions for model training; we exploit the *preference covariance* amongst individuals who are in the same group, to design an auxiliary training objective that enhances the informativeness of user and group representations. Second, we learn robust estimates of the *contextual relevance* of users’ individual preferences in each group and present a novel contextual regularization objective to incorporate knowledge of individual preferences.

We realize the above insights through two novel data-driven model training strategies for group recommendation. *First*, we introduce a *self-supervised learning* objective to contrastively regularize the user-group latent space to capture social user associations and distinctions across groups. We achieve this by maximizing *mutual information* (MI) between representations of groups and group members, which encourages group representations to encode shared group member preferences while regularizing user representations to capture their social associations. *Second*, we contextually identify *informative* group members and regularize the corresponding group representation to reflect their personal preferences. We introduce a novel regularization objective that *contextually* weights users’ personal preferences in each group, in proportion to their user-group MI. *Group-adaptive* preference weighting precludes degenerate solutions that arise during static regularization over ephemeral groups with sparse activities. We summarize our key contributions below:

Architecture-agnostic Group Recommendation Framework: To the best of our knowledge, **Group Information Maximization** (GROUPIM) is the first recommender architecture-agnostic framework for group recommendation. Unlike prior work [131, 132] that design customized group preference aggregators, GROUPIM can integrate arbitrary neural preference encoders and aggregators. We show state-of-the-art results with simple efficient aggregators (such as meanpool) that are regularized within our framework. The effectiveness of meanpool signifies substantially *reduced inference costs* without loss in model expressive power. Thus, GROUPIM facilitates straightforward enhancements to base neural group recommenders.

Group-adaptive Preference Learning and Prioritization: We learn robust estimates of group-specific member relevance. In contrast, prior work incorporate personal preferences through static regularization [123, 129, 131]. We use *Mutual Information* for self-supervised learning over user and group representations to capture preference covariance across individuals in the same group; and prioritize the preferences of highly relevant members through group-adaptive preference weighting; thus effectively overcoming group interaction sparsity in ephemeral groups. An ablation study confirms the superiority of our MI based learning strategies over static alternatives.

Robust Experimental Results: Our experimental results indicate significant performance gains for GROUPIM over prior state-of-the-art group recommendation models on four publicly available datasets (relative gains of 31-62% NDCG@20 and 3-28% Recall@20). Significantly, GROUPIM achieves stronger gains for groups of larger sizes, and groups with diverse member preferences.

We organize the rest of the chapter as follows. In Section 6.3, we formally define the problem, introduce a base group recommender unifying existing neural methods, and discuss its limitations. We describe our proposed framework GROUPIM in Section 6.4, present experimental results in Section 6.5, and finally conclude in Section 6.6.

6.2 RELATED WORK

Group Recommendation: This line of work can be divided into two broad categories based on group types: *persistent* and *ephemeral*. Persistent groups have stable members with rich activity history together, while ephemeral groups comprise users who interact with very few items together [231]. A common approach is to consider persistent groups as virtual users [121]; thus, personalized user-level recommenders can be directly applied. However, such methods cannot handle ephemeral groups with sparse item interactions. We focus on the more challenging scenario—*recommendations to ephemeral groups*.

Prior work either aggregate recommendation results (or item scores) for each member, or aggregate individual member preferences, towards group predictions. They fall into two classes: *score (or late)* aggregation [122] and *preference (or early)* aggregation [123].

Popular *score aggregation* strategies include least misery [122], average [124], maximum satisfaction [125], and relevance and disagreement [126]. However, these are hand-crafted heuristics that overlook real-world group interactions. An empirical comparison of different heuristic strategies [122] has demonstrated the absence of a clear winner, especially with the variance in group sizes and coherence levels.

Early *preference aggregation* strategies [128] generate recommendations by constructing a group profile that combines the profiles (raw item histories) of group members. Recent methods adopt a *model-based* perspective to learn data-driven models. *Probabilistic* methods [123, 129, 237] model the group generative process by considering both the personal preferences and relative influence of members, to differentiate their contributions towards group decisions. However, a key weakness is their assumption that users have the same likelihood to follow individual and collective preferences, across different groups. *Neural* methods explore attention mechanisms [130] to learn data-driven preference aggregators [131, 132, 133].

MoSAN [132] models group interactions via sub-attention networks; however, MoSAN operates on persistent groups while ignoring users’ personal activities. AGREE [131] employs attentional networks for joint training over individual and group interactions; yet, the extent of regularization applied on each user (based on personal activities) is the same across groups, which results in degenerate solutions when applied to ephemeral groups.

An alternative approach to tackle interaction sparsity is to exploit external side information, *e.g.*, social network of users [106, 179, 233, 234], personality traits [235], demographics [236], and interpersonal relationships [232, 238]. In contrast, our setting is conservative and does not include extra side information: we know only user and item ids, and item implicit feedback. We address interaction sparsity through novel data-driven regularization and training strategies [9]. Our goal is to enable a wide spectrum of neural group recommenders to seamlessly integrate suitable preference encoders and aggregators.

Mutual Information: Recent neural MI estimation methods [164] leverage the InfoMax [239] principle for self-supervised representation learning. They exploit the intrinsic *structure* of the input data (*e.g.*, spatial locality in images, community structure in graphs) via MI maximization objectives to improve representational quality. Recent advances employ auto-regressive models [145] and aggregation functions [44, 144, 240] with noise-contrastive loss functions to preserve MI between structurally related inputs.

We leverage the InfoMax principle to exploit the preference covariance *structure* shared amongst group members. A key novelty of our approach is MI-guided weighting to regularize group embeddings with the personal preferences of highly relevant members.

6.3 PRELIMINARIES

In this section, we first formally define the *ephemeral group recommendation* problem. Then, we present a base neural group recommender \mathbf{R} that unifies existing neural methods into a general framework. Finally, we analyze the key shortcomings of \mathbf{R} to discuss motivations for propose new model learning strategies for ephemeral group recommendation.

6.3.1 Problem Definition

We consider the implicit feedback setting (only visits or clicks, no explicit ratings) with a user set \mathcal{U} , an item set \mathcal{I} , a group set \mathcal{G} , a binary $|\mathcal{U}| \times |\mathcal{I}|$ user-item interaction matrix \mathbf{X}_U , and a binary $|\mathcal{G}| \times |\mathcal{I}|$ group-item interaction matrix \mathbf{X}_G . We denote \mathbf{x}_u , \mathbf{x}_g as the corresponding rows for user u and group g in user-item \mathbf{X}_U and group-item \mathbf{X}_G interaction matrices, with $|\mathbf{x}_u|$, $|\mathbf{x}_g|$ indicating their respective number of interacted items.

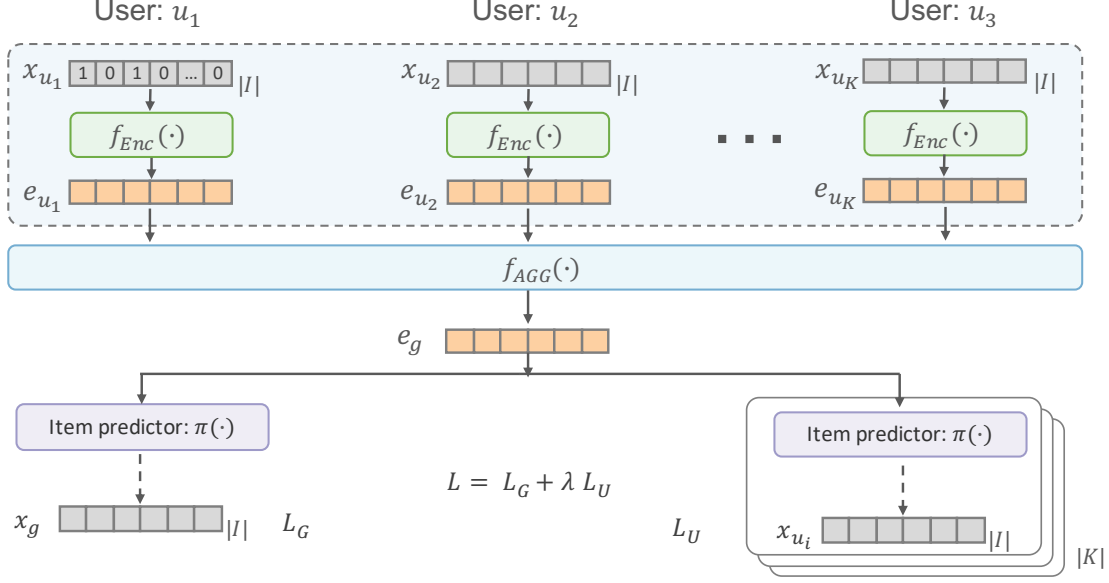


Figure 6.1: Neural architecture diagram of a base neural group recommender with user preference encoder, group preference aggregator and joint user and group losses.

Definition 6.1 (Ephemeral Group). An *ephemeral group* $g \in \mathcal{G}$ comprises a set of $|g|$ users $u^g = \{u_1^g, \dots, u_{|g|}^g\} \subset \mathcal{U}$ with sparse historical interactions \mathbf{x}_g .

Definition 6.2 (Ephemeral Group Recommendation). We evaluate group recommendation on *strict ephemeral groups*, which have never interacted together during training. Given a strict ephemeral group g during testing, our goal is to generate a ranked list over the item set \mathcal{I} relevant to users in u^g , *i.e.*, learn a function $f_G : P(\mathcal{U}) \times \mathcal{I} \mapsto \mathbb{R}$ that maps an ephemeral group and an item to a relevance score, where $P(\mathcal{U})$ is the power set of \mathcal{U} .

6.3.2 Base Neural Group Recommender

Several neural group recommendation models have achieved impressive results [131, 132]. Despite their diversity in modeling group interactions, we remark that state-of-the-art neural methods share a clear model structure: we present a base group recommender \mathbf{R} that includes three modules: a preference encoder; a preference aggregator; and a joint user and group interaction loss. Unifying these neural group recommenders within a single framework facilitates deeper analysis into their shortcomings in addressing ephemeral groups.

The base neural group recommender \mathbf{R} (Figure 6.1) first computes user representations $\mathbf{E} \in \mathbb{R}^{|\mathcal{U}| \times D}$ from individual user-item interactions \mathbf{X}_U using a preference encoder $f_{\text{ENC}}(\cdot)$, followed by applying a neural preference aggregator $f_{\text{AGG}}(\cdot)$ to compute the group representation

$\mathbf{e}_g \in \mathbb{R}^D$ for group g . Finally, the latent group representation \mathbf{e}_g is jointly trained over the group-item \mathbf{X}_G and user-item \mathbf{X}_U interactions, to make item recommendations for group g .

User Preference Representations.

User embeddings \mathbf{E} constitute a latent representation of their personal preferences, indicated in the interaction matrix \mathbf{X}_U . Since latent-factor collaborative filtering methods adopt a variety of strategies (such as matrix factorization, autoencoders, etc.) to learn user embeddings \mathbf{E} , we define the preference encoder $f_{\text{ENC}} : |\mathcal{U}| \times \mathbb{Z}_2^{|I|} \mapsto \mathbb{R}^D$ with two inputs: user u and associated binary personal preference vector \mathbf{x}_u , defined by:

$$\mathbf{e}_u = f_{\text{ENC}}(u, \mathbf{x}_u) \quad \forall u \in \mathcal{U} \quad (6.1)$$

We can augment the user representation \mathbf{e}_u with additional inputs, including user/item contextual attributes, item relationships, etc. via customized encoders [241].

Group Preference Aggregation.

A preference aggregator models the interactions among group members to compute an aggregate representation $\mathbf{e}_g \in \mathbb{R}^D$ for ephemeral group $g \in \mathcal{G}$. Since groups are sets of users with no inherent ordering, we consider the class of permutation-invariant functions (such as summation or pooling operations) on sets [242]. Specifically, $f_{\text{AGG}}(\cdot)$ is permutation-invariant to the order of group member embeddings $\{\mathbf{e}_{u_1}, \dots, \mathbf{e}_{u_{|g|}}\}$. We compute the group representation \mathbf{e}_g for group g using an arbitrary preference aggregator $f_{\text{AGG}}(\cdot)$ as:

$$\mathbf{e}_g = f_{\text{AGG}}(\{\mathbf{e}_u : u \in u^g\}) \quad \forall g \in \mathcal{G} \quad (6.2)$$

Joint User and Group Loss.

The group representation \mathbf{e}_g is trained over the group-item interactions \mathbf{X}_G with group-loss L_G . The framework supports different recommendation objectives, including pairwise [87] and pointwise [88] ranking losses. Here, we use a multinomial likelihood formulation owing to its impressive results in user-based neural collaborative filtering [90]. The group representation \mathbf{e}_g is transformed by a fully connected layer and normalized by a softmax function to produce a probability vector $\boldsymbol{\pi}(\mathbf{e}_g)$ over \mathcal{I} . The group-interaction loss measures the KL-divergence between the normalized purchase history $\mathbf{x}_g/|\mathbf{x}_g|$ (\mathbf{x}_g indicates items interacted by group g) and predicted item probabilities $\boldsymbol{\pi}(\mathbf{e}_g)$, given by:

$$L_G = - \sum_{g \in \mathcal{G}} \frac{1}{|\mathbf{x}_g|} \sum_{i \in \mathcal{I}} x_{gi} \log \pi_i(\mathbf{e}_g); \quad \pi(\mathbf{e}_g) = \text{softmax}(\mathbf{W}_I \mathbf{e}_g) \quad (6.3)$$

Next, we define the user-loss L_U that regularizes the user representations \mathbf{E} with user-item interactions \mathbf{X}_U , thus facilitating joint training with shared encoder $f_{\text{ENC}}(\cdot)$ and predictor (\mathbf{W}_I) layers [131]. We use a similar multinomial likelihood-based formation, given by:

$$L_U = - \sum_{u \in \mathcal{U}} \frac{1}{|\mathbf{x}_u|} \sum_{i \in \mathcal{I}} x_{ui} \log \pi_i(\mathbf{e}_u); \quad L_R = L_G + \lambda L_U \quad (6.4)$$

where L_R denotes the overall loss of the base recommender \mathbf{R} with balancing hyper-parameter λ . Prior work AGREE [131] trains an attentional group preference aggregator with a pairwise regression loss over both \mathbf{X}_U and \mathbf{X}_G , while MoSAN [132] trains a collection of sub-attentional aggregators with bayesian personalized ranking [87] loss on just \mathbf{X}_G . Thus, state-of-the-art neural methods AGREE [131] and MoSAN [132] are specific instances of the framework described by the base neural group recommender \mathbf{R} .

6.3.3 Motivation

To address ephemeral groups, we focus on *learning* strategies that are *independent* of the architectural choices in the base recommender \mathbf{R} . With the rapid advances in neural methods, we envision future enhancements in neural architectures for user representations and group preference aggregation. Since ephemeral groups by definition purchase very few items together, base group recommenders suffer from inadequate training data in group interactions. Here, the group embedding \mathbf{e}_g receives back-propagation signals from sparse interacted items in \mathbf{x}_g , thus lacking evidence to reliably estimate the role of each member. To address group interaction sparsity towards robust ephemeral group recommendation, we propose two data-driven model learning strategies that are independent of the base recommendation mechanisms to generate individual and group representations.

Self-supervised Representation Learning

We note that users' preferences are *group-dependent*; and users occurring together in groups typically exhibit *covarying* preferences (*e.g.*, shared cuisine tastes). Thus, group interactions reveal *distinctions* across groups (*e.g.*, close friends versus colleagues) and latent *user associations* (*e.g.*, co-occurrence of users in similar groups), that are not directly evident when the base group recommender \mathbf{R} only predicts sparse group interactions.

Symbol	Description
\mathbf{X}_U	Binary $ \mathcal{U} \times \mathcal{I} $ user-item interaction matrix
\mathbf{X}_G	Binary $ \mathcal{G} \times \mathcal{I} $ group-item interaction matrix
\mathbf{x}_u	$ \mathcal{I} $ -dimensional row for user u in matrix \mathbf{X}_U
\mathbf{x}_g	$ \mathcal{I} $ -dimensional row for group g in matrix \mathbf{X}_G
\mathbf{R}	Base neural group recommender
f_{ENC}	Individual user preference encoder
f_{AGG}	Group preference aggregator
\mathbf{e}_u	Latent user preference representation for user $u \in \mathcal{U}$
\mathbf{e}_g	Latent group preference representation for group $g \in \mathcal{G}$

Table 6.1: *Notation*

We *contrast* the preference representations of group members against those of non-member users with similar item histories, to effectively regularize the latent space of user and group representations. This promotes the representations to encode latent discriminative characteristics shared by group members, that are not discernible from their sparse interactions.

Group-adaptive Preference Prioritization.

To overcome group interaction sparsity, we critically remark that while groups are ephemeral with sparse interactions, the group members have comparatively richer individual interaction histories. Thus, we propose to *selectively* exploit the personal preferences of group members to enhance the quality of group representations via contextual regularization.

The user-loss L_U (equation 6.4) in base recommender \mathbf{R} attempts to regularize user embeddings \mathbf{E} based on their individual activities \mathbf{X}_U . A key weakness is that L_U forces \mathbf{e}_u to *uniformly* predict preferences \mathbf{x}_u across all groups containing user u . Since groups interact with items differently than individual members, inaccurately utilizing \mathbf{X}_U can become counter-productive. Fixed regularization results in degenerate models that either over-fit or are over-regularized, due to lack of flexibility in adapting preferences per group.

To overcome group interaction sparsity, we *contextually* identify members that are highly *relevant* to the group and regularize the group representation to reflect their personal preferences. To measure contextual relevance, we introduce group-specific relevance weights $w(u, g)$ for each user u where $w(\cdot)$ is a learned weighting function of both the user and group representations. This enhances the expressive power of the group recommender, thus effectively alleviating the challenges imposed by group interaction sparsity.

In this section, we defined ephemeral group recommendation, and presented a base group

recommender architecture with three modules: user representations, group preference aggregation, and joint loss functions. Finally, we motivated the need to: contrastively regularize the user-group space to capture member associations and group distinctions; and learn group-specific weights $w(u, g)$ to regularize group representations with individual preferences.

6.4 GROUPIM FRAMEWORK

We first motivate mutual information towards achieving our two proposed model learning strategies, followed by a detailed description of our proposed framework GROUPIM.

6.4.1 Mutual Information Maximization.

We introduce our user-group mutual information maximization approach through a stylized example. We extend the introductory example to illustrate how to regularize Alice’s latent representation based on her interactions in two different groups. Consider Alice who first goes out for lunch to an Italian restaurant with a visitor Bob, and later dines at a Mexican restaurant with her friend Charlie.

First, Alice plays different roles across the two groups (*i.e.*, stronger influence among friends than with Bob) due to the differences in group context (visitors versus friends). Thus, we require a measure to quantify the contextual *informativeness* of user u in group g .

Second, we require the latent representation of Alice to capture associations with both visitor Bob and friend Charlie, yet express variations in her group activities. Thus, it is necessary to not only *differentiate* the role of Alice across groups, but also compute appropriate representations that make her presence in each group more *coherent*.

To achieve these two goals at once, we maximize *user-group mutual information* (MI) to regularize the latent space of user and group representations, and set group-specific relevance weights $w(u, g)$ in proportion to their estimated MI scores. User-group MI measures the contextual informativeness of a member u towards the group decision through the reduction in group decision uncertainty when user u is included in group g . Unlike correlation measures that quantify monotonic linear associations, mutual information captures complex non-linear statistical relationships between covarying random variables. Our proposed user-group MI maximization strategy enables us to achieve our two-fold motivation (Section 6.3.3):

- **Group Preference Co-variance Learning:** Maximizing user-group MI encourages the group embedding \mathbf{e}_g to encode preference covariance across group members, and regularizes the user embeddings \mathbf{E} to capture social associations in group interactions.

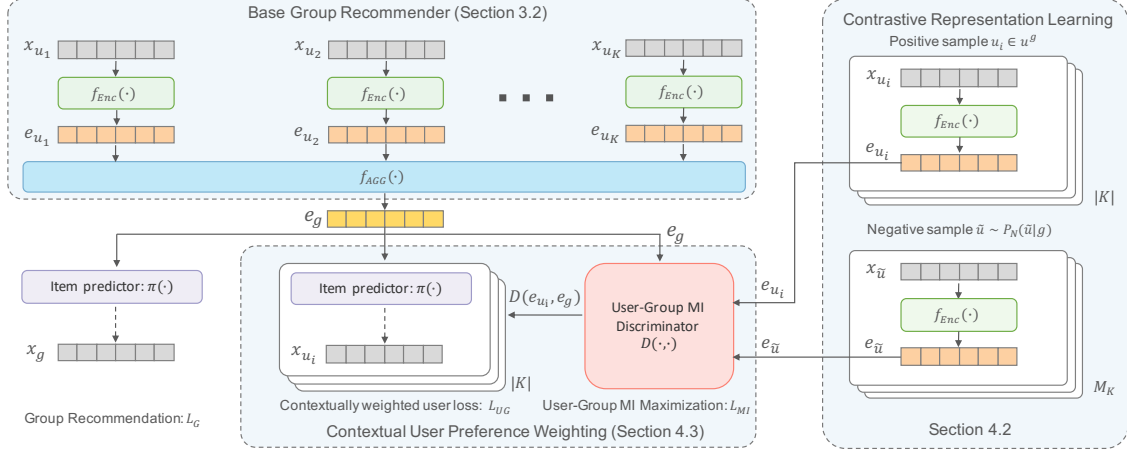


Figure 6.2: Neural architecture diagram of GROUPIM depicting the different model components and loss terms appearing in Equation 6.13.

- **Group-specific User Relevance:** By quantifying $w(u, g)$ through user-group mutual information, we accurately capture the extent of *informativeness* for user u in group g , thus guiding group-adaptive personal preference prioritization for regularization.

6.4.2 User-Group MI Maximization.

Neural MI estimation [164] has demonstrated feasibility to maximize MI by training a classifier \mathbf{D} (a.k.a, discriminator network) to accurately separate *positive* samples drawn from their joint distribution from *negative* samples drawn from the product of marginals.

We maximize user-group MI between group member representations $\{e_u : u \in u^g\}$ and the group representation e_g (computed in equations 6.1 and 6.2 respectively). We train a contrastive *discriminator* network $\mathbf{D} : \mathbb{R}^D \times \mathbb{R}^D \mapsto \mathbb{R}^+$, where $\mathbf{D}(e_u, e_g)$ represents the probability score assigned to this user-group pair (higher scores for users who are members of group g). The positive samples (e_u, e_g) for \mathbf{D} are the preference representations of (u, g) pairs such that $u \in u^g$, and negative samples are derived by pairing e_g with the representations of non-member users sampled from a negative sampling distribution $P_N(u|g)$. The discriminator \mathbf{D} is trained on a noise-contrastive type objective with a binary cross-entropy (BCE) loss between samples from the joint (positive pairs), and the product of marginals (negative pairs), resulting in the following training objective:

$$L_{MI} = -\frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{1}{\alpha_g} \left[\sum_{u \in u^g} \log \mathbf{D}_{ug} + \sum_{j=1}^{M_g} \mathbb{E}_{\tilde{u} \sim P_N} \log(1 - \mathbf{D}_{\tilde{u}g}) \right] \quad (6.5)$$

where $\alpha_g = |g| + M_g$, M_g is the number of negative users sampled for group g and \mathbf{D}_{ug} is a shorthand for $\mathbf{D}(\mathbf{e}_u, \mathbf{e}_g)$. This objective maximizes MI between \mathbf{e}_u and \mathbf{e}_g based on the Jensen-Shannon divergence between the joint and the product of marginals [44].

We employ a *preference-biased* negative sampling distribution $P_{\mathcal{N}}(\tilde{u}|g)$, which assigns higher likelihoods to *non-member* users who have purchased the group items \mathbf{x}_g . These *hard negative examples* encourage the discriminator to learn latent aspects shared by group members by contrasting against other users with similar individual item histories. We define the negative sampling distribution $P_{\mathcal{N}}(\tilde{u}|g)$ as:

$$P_{\mathcal{N}}(\tilde{u}|g) \propto \eta \mathcal{I}(\mathbf{x}_{\tilde{u}}^T \cdot \mathbf{x}_g > 0) + (1 - \eta) \frac{1}{|\mathcal{U}|} \quad (6.6)$$

where $\mathcal{I}(\cdot)$ is an indicator function and η controls the sampling bias. We set $\eta = 0.5$ across all our experiments. In comparison to random negative sampling, our experiments indicate that preference-biased negative user sampling exhibits better discriminative abilities.

When L_{MI} is trained jointly with the base recommender loss L_R (equation 6.4), maximizing user-group MI enhances the quality of user and group representations computed by the encoder $f_{\text{ENC}}(\cdot)$ and aggregator $f_{\text{AGG}}(\cdot)$. We now present our proposed approach to overcome the limitations due to the fixed regularizer L_U (Section 6.3.3).

6.4.3 Contextual User Preference Weighting

In this section, we describe a contextual weighting strategy to identify and prioritize personal preferences of relevant group members, to overcome group interaction sparsity. We avoid degenerate solutions by varying the extent of regularization induced by each \mathbf{x}_u (for user u) across groups through group-specific relevance weights $w(u, g)$. Contextual weighting accounts for user participation in diverse groups with different levels of shared interests.

By maximizing user-group mutual information, the trained discriminator \mathbf{D} outputs scores $\mathbf{D}(\mathbf{e}_u, \mathbf{e}_g)$ that quantify the contextual informativeness of each (u, g) pair (higher scores for informative users). Thus, we set the relevance weight $w(u, g)$ for group member $u \in \mathcal{U}^g$ to be proportional to the discriminator score $\mathbf{D}(\mathbf{e}_u, \mathbf{e}_g)$. Instead of regularizing the user representations \mathbf{E} with \mathbf{x}_u in each group (L_U in eqn 6.4), we directly regularize the group representation \mathbf{e}_g with \mathbf{x}_u in proportion to $\mathbf{D}(\mathbf{e}_u, \mathbf{e}_g)$ for each group member u . Direct optimization of the group representation \mathbf{e}_g (instead of \mathbf{e}_u) results in more effective model regularization with faster flow of gradients, especially with sparse group interactions in ephemeral groups. Thus, we now define the contextually weighted user-loss L_{UG} as:

$$L_{UG} = - \sum_{g \in \mathcal{G}} \frac{1}{|\mathbf{x}_g|} \sum_{i \in \mathcal{I}} \sum_{u \in u^g} \mathbf{D}(\mathbf{e}_u, \mathbf{e}_g) x_{ui} \log \pi_i(\mathbf{e}_g) \quad (6.7)$$

where L_{UG} effectively regularizes the latent group representation \mathbf{e}_g with the individual activities of group member u with contextual relevance weight $\mathbf{D}(\mathbf{e}_u, \mathbf{e}_g)$.

The overall model objective of our framework GROUPIM includes three terms: L_G , L_{UG} , and L_{MI} , which is described in Section 6.4.4 in detail. GROUPIM regularizes the latent representations computed by $f_{\text{ENC}}(\cdot)$ and $f_{\text{AGG}}(\cdot)$ through user-group MI maximization (L_{MI}) to contrastively capture group member associations; and contextual MI-guided weighting (L_{UG}) to prioritize individual preferences.

6.4.4 Model Details

We now describe the architectural details of preference encoder $f_{\text{ENC}}(\cdot)$, aggregator $f_{\text{AGG}}(\cdot)$, discriminator \mathbf{D} , and an alternative optimization approach to train our framework GROUPIM.

User Preference Encoder

To encode individual user preferences \mathbf{X}_U into preference embeddings \mathbf{E} , we use a Multi-Layer Perceptron with two fully connected dense layers, defined by:

$$\mathbf{e}_u = f_{\text{ENC}}(\mathbf{x}_u) = \sigma(\mathbf{W}_2^T (\sigma(\mathbf{W}_1^T \mathbf{x}_u + b_1) + b_2)) \quad (6.8)$$

where \mathbf{e}_u is the latent preference representation for user u , with learnable weight matrices $\mathbf{W}_1 \in \mathbb{R}^{|\mathcal{I}| \times D}$ and $\mathbf{W}_2 \in \mathbb{R}^{D \times D}$, biases $b_1, b_2 \in \mathbb{R}^D$, and $\tanh(\cdot)$ activations for non-linearity σ .

Pre-training: We pre-train the weights and biases of the first encoder layer (\mathbf{W}_1, b_1) on the user-item interaction matrix \mathbf{X}_U with user-loss L_U (equation 6.4). We use these pre-trained parameters to initialize the first layer of $f_{\text{ENC}}(\cdot)$ before optimizing the overall objective of GROUPIM. Our model ablation studies in Section 6.5.5 indicate significant improvements owing to this pre-trained initialization strategy.

Group Preference Aggregators

We consider three preference aggregators MAXPOOL, MEANPOOL and, ATTENTION, which are widely used in graph neural networks [41, 68, 243] and have close ties to aggregators examined in prior work, *i.e.*, MAXPOOL and MEANPOOL mirror the heuristics of maxi-

num satisfaction [125] and averaging [124], while attentions learn varying member contributions [131, 132]. We define the three preference aggregators below:

- **Maxpool:** The preference embedding of each member is passed through MLP layers, followed by element-wise max-pooling to aggregate group member embeddings, given by:

$$\mathbf{e}_g = \text{MAX}(\{\sigma(\mathbf{W}_{\text{AGG}}\mathbf{e}_u + b), \forall u \in u_g\}) \quad (6.9)$$

where MAX denotes the element-wise max operator and $\sigma(\cdot)$ is a nonlinear activation. Intuitively, the MLP layers compute features for each member, and max-pooling over each of the computed features effectively captures different aspects of group members.

- **Meanpool:** We similarly apply an element-wise mean-pooling operation after the MLP layers, to compute the latent group representation \mathbf{e}_g as:

$$\mathbf{e}_g = \text{MEAN}(\{\sigma(\mathbf{W}_{\text{AGG}}\mathbf{e}_u + b), \forall u \in u_g\}) \quad (6.10)$$

- **Attention:** To explicitly differentiate group members' roles, we employ neural attentions [130] to compute a weighted sum of members' preference representations, where the weights are learned by an attention network, parameterized by a single MLP layer.

$$\mathbf{e}_g = \sum_{u \in u_g} \alpha_u \mathbf{W}_{\text{AGG}}\mathbf{e}_u \quad \alpha_u = \frac{\exp(\mathbf{h}^T \mathbf{W}_{\text{AGG}}\mathbf{e}_u)}{\sum_{u' \in u_g} \exp(\mathbf{h}^T \mathbf{W}_{\text{AGG}}\mathbf{e}_{u'})} \quad (6.11)$$

where α_u indicates the contribution of a group member u towards the group decision. We exclude bias terms for the sake of brevity. This formulation can be trivially extended to item-conditioned weighting [131], self-attention [133] and sub-attention networks [132].

Discriminator Architecture

The contrastive discriminator architecture learns a scoring function to assign higher scores to observed (u, g) pairs relative to negative examples, thus parameterizing group-specific relevance weight $w(u, g)$. Similar to existing work on neural mutual information estimation [44], we use a simple bilinear function to score user-group representation pairs. The discriminator function \mathbf{D} is defined by the following equation:

$$\mathbf{D}(\mathbf{e}_u, \mathbf{e}_g) = \sigma(\mathbf{e}_u^T \mathbf{W} \mathbf{e}_g) \quad (6.12)$$

where \mathbf{W} is a learnable scoring matrix and σ is the logistic sigmoid non-linearity function to convert raw scores into probabilities of $(\mathbf{e}_u, \mathbf{e}_g)$ being a positive example. We leave investigation of further architectural variants for the discriminator \mathbf{D} to future work.

Model Optimization

The overall objective of GROUPIM is composed of three terms, the group-loss L_G (Equation 6.3), contextually weighted user-loss L_{UG} (Equation 6.7), and MI maximization loss L_{MI} (Equation 6.5). The combined model training objective is given by:

$$L = \underbrace{L_G}_{\text{Group Recommendation Loss}} + \underbrace{\lambda L_{UG}}_{\text{Contextually Weighted User Loss}} + \underbrace{L_{MI}}_{\text{User-Group MI Maximization Loss}} \quad (6.13)$$

We train GROUPIM using an alternating optimization schedule. In the first step, the discriminator \mathbf{D} is held constant, while optimizing the group recommender on $L_G + \lambda L_{UG}$. The second step trains \mathbf{D} on L_{MI} , resulting in gradient updates for both parameters of \mathbf{D} as well as those of the user encoder $f_{\text{ENC}}(\cdot)$ and preference aggregator $f_{\text{AGG}}(\cdot)$.

Thus, the discriminator \mathbf{D} only seeks to *regularize* the model (*i.e.*, encoder and aggregator) during training through loss terms L_{MI} and L_{UG} . During inference, we directly use the regularized encoder $f_{\text{ENC}}(\cdot)$ and aggregator $f_{\text{AGG}}(\cdot)$ to make group recommendations.

Model Objective Interpretation

To understand our model better, we examine a special case with fixed relevance weights $w(u, g) = 1$ without MI maximization. Here, the overall model objective reduces to:

$$L' = - \sum_{g \in \mathcal{G}} \left[\frac{1}{|\mathbf{x}_g|} \sum_{i \in \mathcal{I}} x_{gi} \log \pi_i(\mathbf{e}_g) + \lambda \sum_{i \in \mathcal{I}} \bar{x}_g \log \pi_i(\mathbf{e}_g) \right] \quad (6.14)$$

where $\bar{x}_{gi} = \frac{1}{|\mathbf{x}_g|} \sum_{u \in u^g} x_{ui}$ is the mean preference vector of users in group g for item $i \in \mathcal{I}$. The first term is a supervised loss to predict label x_{gi} from group embedding \mathbf{e}_g . The second term can be viewed as an unsupervised *group reconstruction* loss that encodes member preferences $\{\mathbf{x}_u : u \in u^g\}$ into the group embedding \mathbf{e}_g to reconstruct their *mean* preference vector \bar{x}_g . In the special case of $w(u, g) = 1$, GROUPIM reduces to a classical semi-supervised learning objective (similar to Equation 6.4) for group recommendation.

Dataset	Yelp	Weeplaces	Gowalla	Douban
# Users	7,037	8,643	25,405	23,032
# Items	7,671	25,081	38,306	10,039
# Groups	10,214	22,733	44,565	58,983
# U-I interactions	220,091	1,358,458	1,025,500	1,731,429
# G-I Interactions	11,160	180,229	154,526	93,635
Average # items/user	31.3	58.83	40.37	75.17
Average # items/group	1.09	2.95	3.47	1.59
Average group size	6.8	2.9	2.8	4.2

Table 6.2: *Statistics of four real-world datasets with ephemeral groups. Group-Item interactions are sparse: average number of interacted items per group < 3.5 .*

Thus, our framework GROUPIM can be overall regarded as jointly performing *self-supervised* learning with *user-group mutual information maximization* (via loss term L_{MI}) and *semi-supervised* learning with *group-adaptive* regularization (via loss term L_{UG}).

6.5 EXPERIMENTS

In this section, we present an extensive quantitative and qualitative analysis of our model. We first introduce datasets, baselines, and experimental setup (Section 6.5.1, 6.5.2, and 6.5.3), followed by our main group recommendation results (Section 6.5.4). In Section 6.5.5, we conduct an ablation study to understand our gains over the base recommender. In Section 6.5.6, we study how key group characteristics (*group size*, *coherence*, and *aggregate diversity*) impact recommendation results. In Section 6.5.7, we qualitatively visualize the variation in discriminator scores assigned to group members, for different kinds of groups. Finally, we discuss the limitations of our approach in Section 6.5.9.

6.5.1 Datasets

First, we conduct experiments on large-scale POI (point-of-interest) recommendation datasets extracted from three location-based social networks. Since the POI datasets do not contain explicit group interactions, we construct group interactions by jointly using the check-ins and social network information: check-ins at the same POI within 15 minutes by an individual and her subset of friends in the social network together constitutes a single group interaction, while the remaining check-ins at the POI correspond to individual interactions. We define the group recommendation task as recommending POIs to ephemeral groups of

users. The datasets were pre-processed to retain users and items with five or more check-ins each. We present dataset descriptions below:

- **Weeplaces**¹: we extract check-ins on POIs over all major cities in the United States, across various categories including Food, Nightlife, Outdoors, Entertainment and Travel.
- **Yelp**²: user check-ins on local restaurants located in the city of Los Angeles, California.
- **Gowalla** [211]: we use restaurant check-ins across all cities in the United States, in the time period upto June 2011.

Second, we evaluate event venue recommendation on Douban, which is the largest online event-based social networking platform in China.

- **Douban** [233]: users organize and participate in social events, where users attend events together in groups and items correspond to event venues. During pre-processing, we filter out users and venues with less than 10 interactions each.

Groups across all datasets are ephemeral since group interactions are sparse (average number of items per group < 3.5 in Table 6.2)

6.5.2 Baselines

We present comparisons against state-of-the-art baselines in two categories: score aggregation methods with predefined aggregators, and data-driven preference aggregators.

- **Popularity** [244]: recommends items to groups based on item popularity, which is measured by its interaction count in the training set.
- **User-based CF + Score Aggregation**: We utilize a state-of-the-art neural recommendation model VAE-CF [90], followed by score aggregation via: averaging (AVG), least-misery (LM), maximum satisfaction (MAX), and relevance-disagreement (RD).
- **COM** [123]: a probabilistic generative model that determines group decisions based on group members' individual preferences and topic-dependent influence.
- **CrowdRec** [129]: a probabilistic generative model that extends COM through item-specific latent variables capturing their global popularity for group recommendation.

¹<https://www.yongliu.org/datasets/>

²<https://www.yelp.com/dataset/challenge>

- **MoSAN** [132]: a neural group recommender that employs a collection of sub-attentional networks to model group member interactions. Since MoSAN originally ignores individual activities \mathbf{X}_U , we include \mathbf{X}_U into \mathbf{X}_G as pseudo-groups with single users.
- **AGREE** [131]: a neural group recommender that utilizes attentional group preference aggregation to compute item-specific group member weights, for joint model training over individual and group interactions.

We tested **GroupIM** by substituting three preference aggregators, MAXPOOL, MEANPOOL, and ATTENTION (Section 6.4.4). All experiments were conducted on a single Nvidia Tesla V100 GPU with *PyTorch* [245] implementations on the Linux platform. Our implementation of **GroupIM** and datasets are publicly available³.

Dataset	Yelp (LA)		Weeplaces		Gowalla		Douban	
Metric	N@20	N@50	N@20	N@50	N@20	N@50	N@20	N@50
Predefined Score Aggregators								
Popularity [244]	0.000	0.000	0.063	0.074	0.075	0.088	0.003	0.005
VAE-CF + AVG [90, 124]	0.142	0.179	0.273	0.313	0.318	0.362	0.179	0.217
VAE-CF + LM [90, 122]	0.097	0.120	0.277	0.311	0.375	0.409	0.221	0.252
VAE-CF + MAX [90, 125]	0.099	0.133	0.229	0.270	0.267	0.316	0.156	0.194
VAE-CF + RD [90, 126]	0.143	0.181	0.239	0.279	0.294	0.339	0.178	0.216
Data-driven Preference Aggregators								
COM [123]	0.143	0.154	0.329	0.348	0.223	0.234	0.283	0.288
Crowdrec [129]	0.082	0.101	0.353	0.370	0.325	0.338	0.121	0.188
AGREE [131]	0.123	0.168	0.242	0.292	0.160	0.223	0.126	0.173
MoSAN [132]	0.470	0.494	0.287	0.334	0.323	0.372	0.193	0.239
Group Information Maximization Recommenders (GroupIM)								
GroupIM-Maxpool	0.488	0.501	0.479	0.505	0.433	0.463	0.291	0.313
GroupIM-Meanpool	0.629	0.637	0.518	0.543	0.476	0.504	0.323	0.351
GroupIM-Attention	0.633	0.647	0.521	0.546	0.477	0.505	0.325	0.356

Table 6.3: Group recommendation results on four datasets, $N@K$ denotes NDCG@ K metric at $K = 20$ and 50. The GROUPIM variants indicate MAXPOOL, MEANPOOL, and ATTENTION as preference aggregators in our MI maximization framework. GROUPIM achieves significant performance gains of 31 to 62% NDCG@20 over competing group recommenders. Notice that MEANPOOL and ATTENTION variants of our model GROUPIM achieve comparable group recommendation performance across all datasets.

³<https://github.com/CrowdDynamicsLab/GroupIM>

6.5.3 Experimental Setup

We randomly split the set of all groups into training (70%), validation (10%), and test (20%) sets, while utilizing the individual interactions of all users for training. Note that each group appears only in one of the three sets. The test set contains *strict ephemeral groups* (*i.e.*, a specific combination of users) that do not occur in the training set. Thus, we train on ephemeral groups and test on strict ephemeral groups. We use NDCG@K and RECALL@K as evaluation metrics. RECALL@K measures the percentage of relevant items in the top-K rank list for each group, while NDCG@K is position-sensitive and assigns higher scores for relevant items at the top. We report model performance at $K = 20, 50$ in Tables 6.3 and 6.4.

We tune the latent dimension (number of topics for COM and Crowdrec) in the range $\{32, 64, 128\}$ for all models and other baseline hyper-parameters in ranges centered at author-provided values on the validation set. In GroupIM, we use two fully connected layers of size 64 each in the MLP preference encoder $f_{\text{ENC}}(\cdot)$ and tune λ in the range $\{2^{-4}, 2^{-3}, \dots, 2^6\}$. We use 5 negative samples for each true user-group pair to train the discriminator. Our experimental results are averaged over 10 random runs with random initializations.

Dataset Metric	Yelp (LA)		Weeplaces		Gowalla		Douban	
	R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50
Predefined Score Aggregators								
Popularity [244]	0.001	0.001	0.126	0.176	0.143	0.203	0.009	0.018
VAE-CF + AVG [90, 124]	0.322	0.513	0.502	0.666	0.580	0.758	0.381	0.558
VAE-CF + LM [90, 122]	0.198	0.316	0.498	0.640	0.610	0.750	0.414	0.555
VAE-CF + MAX [90, 125]	0.231	0.401	0.431	0.604	0.498	0.702	0.339	0.517
VAE-CF + RD [90, 126]	0.321	0.513	0.466	0.634	0.543	0.723	0.379	0.557
Data-driven Preference Aggregators								
COM [123]	0.232	0.286	0.472	0.557	0.326	0.365	0.417	0.436
Crowdrec [129]	0.217	0.315	0.534	0.609	0.489	0.548	0.375	0.681
AGREE [131]	0.332	0.545	0.484	0.711	0.351	0.605	0.310	0.536
MoSAN [132]	0.757	0.875	0.548	0.738	0.584	0.779	0.424	0.639
Group Information Maximization Recommenders (GroupIM)								
GroupIM-Maxpool	0.676	0.769	0.676	0.776	0.628	0.747	0.524	0.637
GroupIM-Meanpool	0.778	0.846	0.706	0.804	0.682	0.788	0.569	0.709
GroupIM-Attention	0.782	0.851	0.716	0.813	0.686	0.796	0.575	0.714

Table 6.4: Group recommendation results on four datasets, $R@K$ denotes the RECALL@K metric at $K = 20$ and 50. The GROUPIM variants indicate MAXPOOL, MEANPOOL, and ATTENTION as preference aggregators. GROUPIM achieves significant gains of 3 to 28% RECALL@20 over competing group recommenders.

6.5.4 Experimental Results

We note the following key observations from our experimental results comparing our framework **GroupIM** with its three aggregator variants, against competing baselines on ephemeral group recommendation (Tables 6.3 and 6.4).

First, heuristic score aggregation with neural user-level recommenders (*i.e.*, **VAE-CF**) performs comparable to (and often beats) probabilistic models (**COM**, **Crowdrec**). Neural methods with multiple non-linear transformations, are expressive enough to identify latent groups of similar users just from their individual item interactions.

Second, there is no clear winner among the different pre-defined score aggregation strategies, *e.g.*, **VAE-CF** + **LM** (least misery) outperforms the rest on Gowalla and Douban, while **VAE-CF** + **LM** (averaging) is superior on Yelp and Weeplaces. This empirically validates the non-existence of a single optimal strategy for all datasets.

Third, **MoSAN** [132] outperforms both probabilistic models and fixed score aggregators on most datasets. **MoSAN** achieves better results owing to the expressive power of neural preference aggregators (such as sub-attention networks) to capture group member interactions, albeit not explicitly differentiating personal and group activities. Notice that naive joint training over personal and group activities via static regularization (as in **AGREE** [131]) results in poor performance due to sparsity in group interactions. Static regularizers on \mathbf{X}_U cannot distinguish the role of users across groups, resulting in models that lack generalization to ephemeral groups with sparse interactions.

GroupIM variants outperform baselines significantly, with **ATTENTION** achieving overall best results. In contrast to neural methods (*i.e.*, **MoSAN** and **AGREE**), **GroupIM** regularizes the latent representations by contextually weighting the personal preferences of informative members, thus effectively tackling group interaction sparsity. The **MAXPOOL** variant is noticeably inferior, due to the higher sensitivity of *max* operation to outlier members.

Note that **MEANPOOL** performs comparably to **ATTENTION**. This is because in **GroupIM**, the discriminator \mathbf{D} does the heavy-lifting of contextually differentiating the role of users across groups to effectively regularize the encoder $f_{\text{ENC}}(\cdot)$ and aggregator $f_{\text{AGG}}(\cdot)$ modules. If $f_{\text{ENC}}(\cdot)$ and \mathbf{D} are expressive enough, efficient **MEANPOOL** aggregation can achieve near state-of-the-art results (Tables 6.3 and 6.4).

An important implication is the *reduced inference complexity* of our model, *i.e.*, once trained using our MI maximizing framework, simple aggregators (such as meanpool) suffice to achieve state-of-the-art performance. This is especially significant, considering that our closest baseline **MoSAN** [132] utilizes sub-attentional preference aggregation networks that scale quadratically with group size.

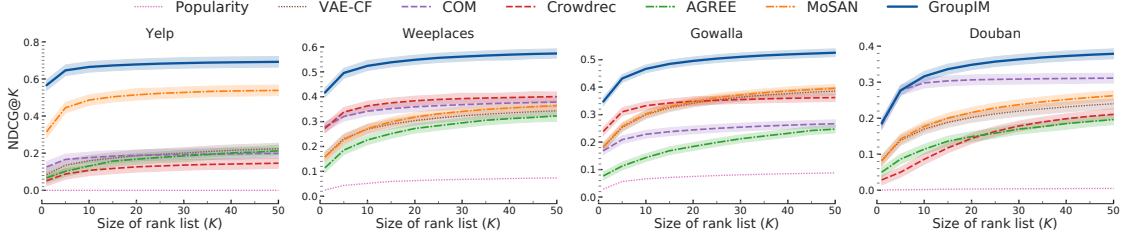


Figure 6.3: NDCG@ K across size of rank list K . Variance bands indicate 95% confidence intervals over 10 random runs. Existing methods underperform since they either disregard member roles (VAE-CF variants) or overfit to the sparse group activities. **GroupIM** contextually identifies informative members and regularizes their representations, to show strong gains.

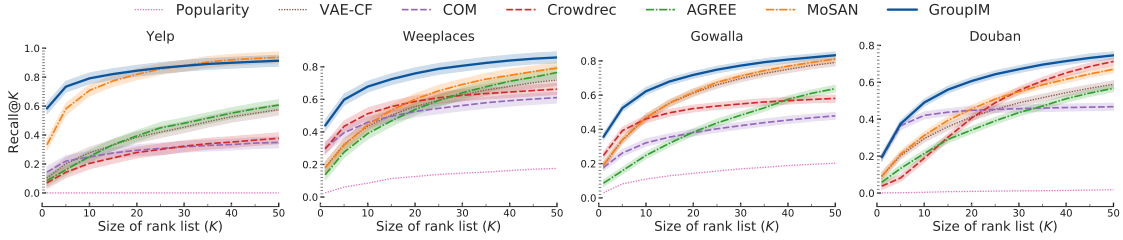


Figure 6.4: RECALL@ K across size of rank list K (1 to 50). Variance bands indicate 95% confidence intervals over 10 independent runs. **GroupIM** has larger recall gains for smaller K (< 20), indicating more accurate recommendations within top ranks.

We compare the variation in NDCG and RECALL scores of all models with size of rank list K (1 to 50) in figures 6.3 and 6.4. We only depict the best aggregator for **VAE-CF**. **GroupIM** consistently generates more *precise* recommendations across all datasets. We observe smaller gains in Douban, where the user-item interactions exhibit substantial correlation with corresponding group activities. **GroupIM** achieves significant performance gains in characterizing diverse groups, evidenced by our results in section 6.5.6.

6.5.5 Model Analysis

In this section, we present an ablation study to analyze several variants of **GroupIM**, guided by our motivations (Section 6.3.3). In our experiments, we choose ATTENTION as the aggregator due to its consistently high performance. We conduct empirical studies on Weeplaces and Gowalla datasets to report NDCG@50 and RECALL@50 in Table 6.5.

First, we examine the model performance of the base group recommender **R** (Section 6.3.2) which does not utilize user-group MI estimation or maximization for model training. The ablation study results are shown in Table 6.5.

Dataset Metric	Weeplaces		Gowalla	
	N@50	R@50	N@50	R@50
Base Group Recommender Variants				
(1) Base (L_G)	0.420	0.621	0.369	0.572
(2) Base ($L_G + \lambda L_U$)	0.427	0.653	0.401	0.647
GroupIM Variants				
(3) GroupIM ($L_G + L_{MI}$)	0.431	0.646	0.391	0.625
(4) GroupIM (Uniform weights)	0.441	0.723	0.418	0.721
(5) GroupIM (Cosine similarity)	0.488	0.757	0.445	0.739
(6) GroupIM (No pre-training)	0.524	0.773	0.472	0.753
(7) GroupIM ($L_G + \lambda L_{UG} + L_{MI}$)	0.543	0.804	0.505	0.796

Table 6.5: GROUPIM *ablation study* (NDCG and RECALL at $K = 50$). *Contrastive representation learning (row 3) improves the base recommender (row 1), but is substantially more effective with group-adaptive preference weighting (row 7).*

Base Group Recommender.

We examine two variants below:

- (1) We train the base recommender \mathbf{R} on group interactions \mathbf{X}_G with loss L_G (equation 6.3).
- (2) We train the base recommender \mathbf{R} jointly on individual \mathbf{X}_U and group \mathbf{X}_G interactions with static regularization on \mathbf{X}_U using joint loss L_R (Equation 6.4).

In comparison to similar neural group preference aggregator **MoSAN**, our base group recommender \mathbf{R} is stronger on NDCG but inferior on RECALL. The difference is likely due to the multinomial likelihood used to train \mathbf{R} , in contrast to the ranking loss in **MoSAN**. Static regularization via \mathbf{X}_U (row 1) results in higher gains for Gowalla (richer user-item interactions) with relatively larger margins for RECALL than NDCG. We now empirically examine the different model variants of **GroupIM** in two parts:

GroupIM: Self-supervised Representation Learning.

We analyze the benefits derived by just training the contrastive discriminator \mathbf{D} to capture group member associations through user-group MI maximization, *i.e.*, we define a model variant (row 3) to optimize just $L_G + L_{MI}$, without the L_{UG} term. Direct user-group MI maximization (row 3) improves over the base group recommender \mathbf{R} (row 1), thus validating the benefits of contrastive self-supervised learning via mutual information maximization, however still suffers from lack of user preference prioritization.

GroupIM: Group-adaptive Preference Prioritization

We analyze the utility of data-driven contextual weighting (via user-group MI), by examining two alternate *fixed* strategies to define $w(u, g)$ in the loss term L_{UG} of GROUPIM:

- (4) **Uniform weights**: We assign the same relevance weight $w(u, g) = 1$ for each group member u in group g , when optimizing the loss term L_{UG} .
- (5) **Cosine similarity**: To model user-group correlation, we set the relevance weight $w(u, g)$ as the cosine similarity between \mathbf{x}_u and \mathbf{x}_g .

From table 6.5 (rows 4 and 5), the uniform weights variant of loss L_{UG} (row 4) surpasses the statically regularized model (row 2), due to more direct feedback from \mathbf{X}_U to the group embedding \mathbf{e}_g during model training. Cosine similarity (row 5) achieves stronger gains owing to more accurate correlation-guided user weighting across groups. Our model **GroupIM** (row 7) has strong gains over the fixed weighting strategies as a result of its regularization strategy to contextually identify informative members across different groups.

GroupIM: Pre-training $f_{\text{ENC}}(\cdot)$ on \mathbf{X}_U

We depict model performance without pre-training (random initializations) in row 6. Our model (row 7) achieves noticeable gains; pre-training identifies good model initialization points for better convergence.

6.5.6 Impact of Group Characteristics

In this section, we examine our results closely to understand the reason for **GroupIM**'s gains over baselines in two datasets, Weeplaces and Gowalla. We study ephemeral groups along three facets: *group size*; *group coherence*; and *group aggregate diversity*.

Group Size

We partition test groups into bins based on five levels of group size (2-3, 4-5, 6-7, 8-9, and ≥ 10). Figure 6.5 depicts the variation in NDCG@50 scores on Weeplaces and Gowalla.

We make three key observations: methods that explicitly distinguish individual and group activities (such as **COM**, **CrowdRec**, **GroupIM**), exhibit distinctive trends *wrt* group size. In contrast, **MoSAN** [132] and **AGREE** [131], which either uniformly mix both behaviors or apply static regularizers, show no noticeable variation; Performance generally increases with group size. Although test groups are previously unseen, for larger groups, subsets of inter-user interactions are more likely to be seen during training, thus resulting

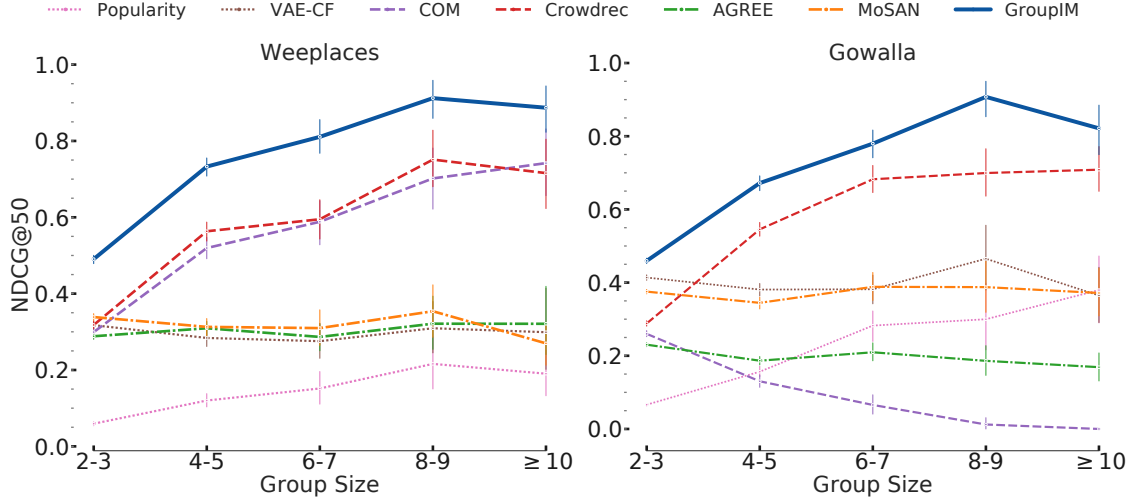


Figure 6.5: Performance ($NDCG@50$), across group sizes. **GroupIM** has larger gains for larger groups due to accurate user associations learnt via MI maximization.

in better performance; **GroupIM** achieves higher (or steady) gains for groups of larger sizes owing to its more accurate prioritization of personal preferences for each member, *e.g.*, **GroupIM** clearly has stronger gains for groups of sizes 8-9 and ≥ 10 in Gowalla.

Group Coherence

We define *group coherence* as the mean pair-wise correlation of personal activities (\mathbf{x}_u) of group members, *i.e.*, if a group has users who frequently co-purchase items, it receives greater coherence. We separate test groups into four quartiles by their coherence scores. Figure 6.6 depicts $NDCG@50$ for groups under each quartile (Q1 - Lower values).

GroupIM has stronger gains for groups with low coherence (quartiles Q1 and Q2), which empirically validates the efficacy of contextual user preference weighting in regularizing the encoder and aggregator, for groups with dissimilar member preferences.

Group Aggregate Diversity

We adapt the classical *aggregate diversity* metric [246] to define *group aggregate diversity* as the total number of distinct items interacted across all group members, *i.e.*, if the set of all purchases of group members covers a wider range of items, then the group has higher aggregate diversity. We report $NDCG@50$ across aggregate diversity quartiles in figure 6.7.

Model performance typically decays (and stabilizes), with increase in group aggregate diversity. Diverse groups with large candidate item sets pose an information overload for

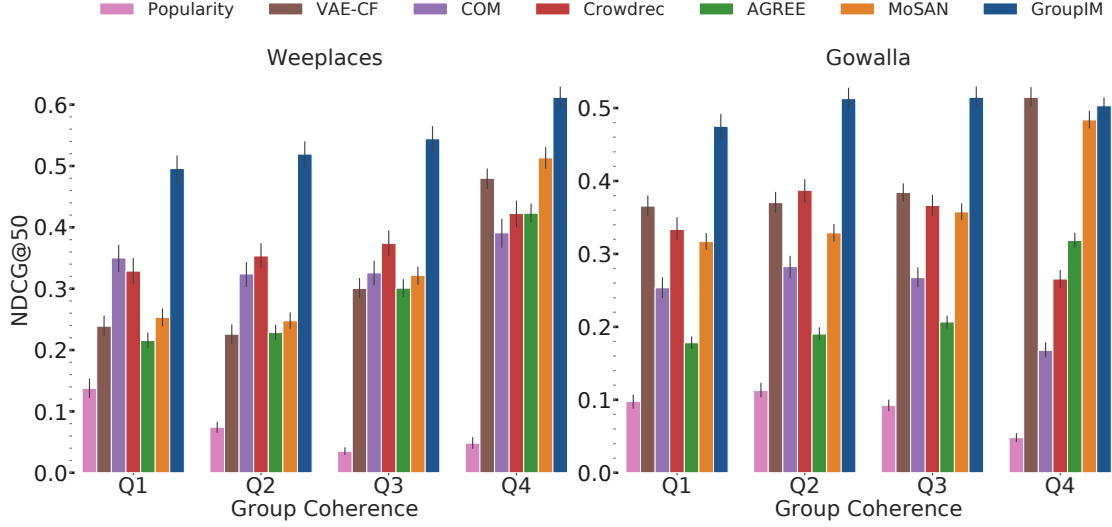


Figure 6.6: Performance (NDCG@50), across group coherence quartiles (Q1: low-est, Q4: highest). **GroupIM** has larger gains in Q1 & Q2 (low group coherence).

group recommenders, leading to worse results. Contextual prioritization with contrastive self-supervised learning benefits diverse groups, as evidenced by the higher relative gains of **GroupIM** highly for diverse groups (quartiles Q3 and Q4).

6.5.7 Qualitative MI Discriminator Analysis

We examine the contextual relevance weights $w(u, g)$ estimated by **GroupIM** over test ephemeral groups, across group size and coherence.

We divide groups into four bins based on group sizes (2-3, 4-6, 7-9, and ≥ 10), and partition them into quartiles based on group coherence per bin. To analyze the variation in *contextual informativeness* across group members, we compute *MI variation* as the standard deviation of scores given by **D** over group members. Figure 6.8 depicts letter-value plots of *MI variation* for groups in corresponding coherence quartiles across group sizes on Weeplaces.

MI variation increases with group size, since larger groups often comprise users with divergent roles and interests. Thus, the learned discriminator generalizes to unseen groups, to discern and estimate markedly different relevance scores for each group member. To further examine the intuition conveyed by the scores, we compare *MI variation* across group coherence quartiles within each group size-range.

MI variation is negatively correlated with group coherence for groups of similar sizes, *e.g.*, *MI variation* is consistently higher for groups with low coherence (quartiles Q1 and Q2). For highly coherent groups (quartile Q4), the discriminator **D** assigns comparable scores

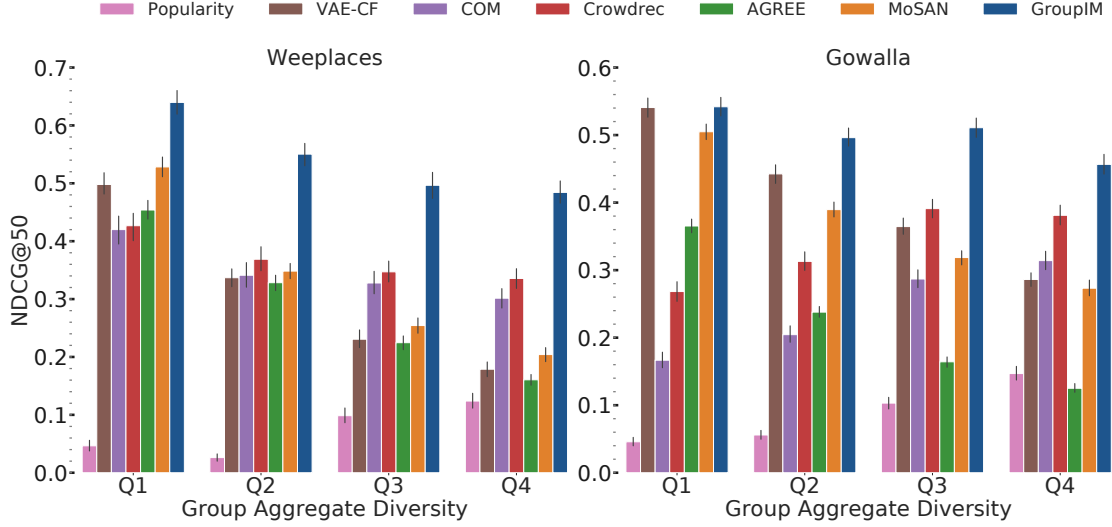


Figure 6.7: Performance ($NDCG@50$), across group aggregate diversity quartiles (Q1: lowest, Q4: highest). GROUPIM has larger gains in Q3 & Q4 (high diversity).

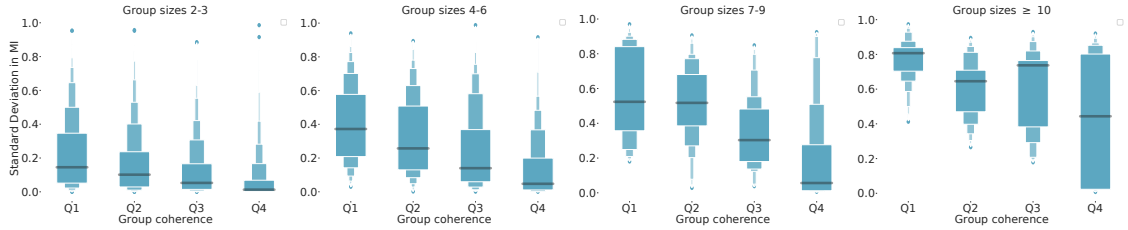


Figure 6.8: MI variation (std. deviation in discriminator scores over members) per group coherence quartile across group sizes. For groups of a given size, as coherence increases, MI variation decreases. As groups increase in size, MI variation increases.

across all members, which is consistent with our intuitions and earlier empirical results on the efficacy of simple averaging strategies for coherent groups.

6.5.8 Sensitivity Analysis

In this section, we analyze parameter sensitivity of our framework GROUPIM with respect to the user-preference weight λ . We show a plot depicting model performance ($NDCG@50$) versus λ in figure 6.9. Varying the hyper-parameter λ in different ranges of values results in performance drops on either side, but for differing reasons. Low λ values result in overfitting to the group activities \mathbf{X}_G , while substantially larger values result in degenerate representations that lack group distinctions. Although we tune λ for each dataset, we find recommendation performance of our model to be stable in a broad range of values that transfer across different datasets.

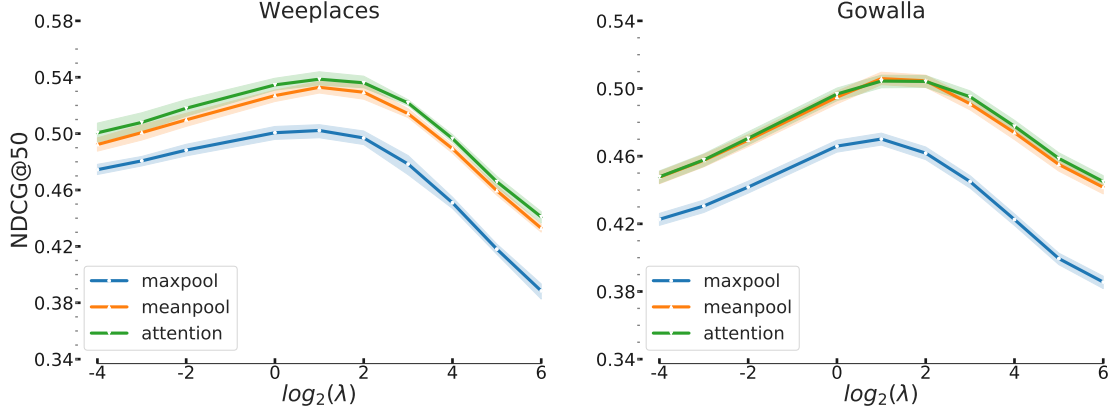


Figure 6.9: λ is varied in $\{2^{-4}, 2^{-3}, \dots, 2^6\}$. Larger λ values result in performance drops due to overfitting on \mathbf{X}_U . GROUPIM is robust in a wide range of values.

6.5.9 Limitations

We identify two limitations of our work. Despite *learning* to contextually prioritize users’ preferences across groups, λ controls the overall strength of preference regularization. Since optimal λ varies across datasets and applications, we plan to explore meta-learning approaches to eliminate such hyper-parameters [247].

GroupIM relies on user-group MI estimation to contextually identify informative members, which might become challenging when users have sparse individual interaction histories. In such a scenario, side information (*e.g.*, social network of users), or contextual factors (*e.g.*, location, interaction time) [105] can prove effective.

6.6 CONCLUSION

This chapter introduces a recommender architecture-agnostic framework GROUPIM that integrates arbitrary neural preference encoders and aggregators for ephemeral group recommendation. To overcome group interaction sparsity, GROUPIM regularizes the user-group representation space by maximizing user-group MI to contrastively capture preference covariance among group members. Unlike prior work that incorporate individual preferences through static regularizers, we dynamically prioritize the preferences of informative members through MI-guided contextual preference weighting. Our extensive experiments on four real-world datasets show significant gains for GROUPIM over state-of-the-art methods.

In this chapter, we explored our second inductive learning application of generating item recommendations to ephemeral groups of users in a multipartite interaction setting involving users, groups, and items. We designed a self-supervised learning framework to specifically

overcome group interaction sparsity challenges. Finally, in the next chapter, we examine a practical industrial application of generating friend suggestions in large-scale social platforms, where we design inductive learning models with the capability to seamlessly incorporate new users and dynamic user interactions.

CHAPTER 7: GRAFRANK: MULTI-FACETED FRIEND RANKING IN SOCIAL PLATFORMS

7.1 INTRODUCTION

Learning latent user representations has become increasingly important in advancing user understanding, and has seen widespread adoption in various industrial settings, *e.g.*, video recommendations on YouTube [3], related pin recommendations on Pinterest [5] etc. The user representations learned using deep models are *effective* at complementing, or even replacing conventional collaborative filtering methods [248], and are *versatile*, *e.g.*, the learned user embeddings can be used to suggest new friendships and also to infer profile attributes (*e.g.*, age, gender) in social networks.

Learning latent representations of nodes in graphs has prominent applications in multiple *academic* settings, such as link prediction [154], community detection [249], and *industrial* recommender systems, including e-commerce [250, 251], content discovery [5, 252], and food delivery [253]. Graph Neural Networks (GNNs) [153] have emerged as a popular paradigm for graph representation learning due to their ability to learn representations combining graph structure and node/link attributes, without relying on expensive feature engineering. GNNs can be formulated as a *message passing* framework where node representations are learned by propagating features from local graph neighborhoods via trainable aggregators. Recently, GNNs have demonstrated promising results in a few industrial systems designed for item recommendations in bipartite [5] or multipartite [252] user-to-item interaction graphs.

Despite their rich representational ability, GNNs have been relatively unexplored in large-scale *user-user social interaction* modeling applications, like friend suggestion. Recommending new potential friends to encourage users to expand their networks, is a cornerstone of social networking, and plays an important role towards user retention, and promoting engagement within the platform.

Prior efforts typically formulate friend suggestion as link prediction (or matrix completion) with a rich literature of graph-based heuristics [254] to quantify user-user affinity, *e.g.*, two users are more likely to connect if they have many common friends. A few GNN models target link prediction in academic settings, learn aggregators over enclosing subgraphs around each candidate link [154, 255, 256]; such models do not scale to industry-scale social graphs with over millions of nodes and billions of edges. Still, GNNs have enormous potential for learning expressive user representations in social networks, due to their intuitive message-passing paradigm that enables attention to social influence from friends in their ego-network.

Yet, designing GNNs for friend recommendations in large-scale social media poses unique

challenges. First, social networks are characterized by *heavy-tailed* degree distributions, *e.g.*, many networks approximately follow power-law distributions [165]. This poses a key challenge of *limited structural information* for a significant proportion of users with very few friends. A related challenge is *activity sparsity* where a very small fraction of users actively form new friendships at any given time. Secondly, contemporary social networking platforms offer a multitude of avenues for users to interact, *e.g.*, users can communicate with friends either by *directly* exchanging messages, pictures, and videos, or *indirectly* through a variety of social actions, including liking, sharing, and commenting on posts. Extracting knowledge from such *heterogeneous* in-platform user actions is challenging, yet extremely *valuable* to address sparsity challenges for a vast majority of inactive users.

Present Work: In this work, we overcome structural and interactional sparsity by exploiting the rich knowledge of heterogeneous in-platform actions. We formulate friend recommendation on social networks as *multi-faceted friend ranking* on an *evolving* friendship graph, with multi-modal user features and link communication features (Figure 7.1). We represent users with heterogeneous feature sets spanning multiple *modalities*, that include a collection of *static* profile attributes (*e.g.*, demographic information) and *time-sensitive* in-platform activities (*e.g.*, content interests and interactions). We also leverage pairwise link features on existing friendships, which capture recent communication activities across multiple direct (*e.g.*, messages) and indirect (*e.g.*, stories) channels within the platform.

To understand the complexity of user interactions and gain insights into various factors impacting friendship formation, we conduct an empirical analysis to investigate *attribute homophily* with respect to different user feature modalities. Our analysis reveals *diverse homophily distributions* across modalities and users, and indicates non-trivial *cross-modality* correlations. Motivated by these observations, we design an end-to-end GNN architecture, GRAFRANK (Graph Attentional Friend Ranker) for multi-faceted friend ranking.

GRAFRANK generates user representations by *modality-specific* neighbor aggregation and *cross-modality* attention. Here, we handle heterogeneity in modality homophily by learning modality-specific message-passing aggregators to compute a set of latent representations for each user. Specifically, the neighbor aggregator is driven by a friendship attention mechanism that captures the influence of individual features and pairwise communications. We introduce a *cross-modality* attention module to compute the effective user representation by attending over the different modality-specific representations for each user, thereby learning non-linear correlations across modalities. We summarize our key contributions below:

Graph-Neural Friend Ranking: To our knowledge, ours is the first work to investigate *graph neural network* usage and design for social user-user interaction modeling

applications. Unlike prior work that typically view friend recommendation as structural link prediction, we present a novel formulation with multi-modal user features and link features, to leverage knowledge of rich heterogeneous user activities in social networking platforms.

GraFRank Model: Motivated by our empirical study that reveals heterogeneity in modality homophily and cross-modality correlations, we design a novel GNN architecture, GRAFRANK, to learn *multi-faceted* user representations. Distinct from conventional GNNs that operate on a single homogeneous feature space, GRAFRANK encapsulates information from multiple correlated feature modalities and user-user interactions.

Robust Experimental Results: Our experiments on two large-scale datasets from a popular social networking platform Snapchat, indicate significant gains for GRAFRANK over state-of-the-art baselines on friend candidate retrieval (relative MRR gains of 30%) and ranking (relative MRR gains of 20%) tasks. Our qualitative analysis indicates stronger gains for the crucial population of *less-active* and *low-degree* users.

7.2 RELATED WORK

We briefly review a few related lines of work on friend recommendations, graph neural networks, and multi-modal learning.

Friend Recommendation: The earliest methods were carefully designed *graph-based heuristics* of user-user proximity in social networks [254], *e.g.*, path-based Katz centrality [60] or common neighbor-based Adamic/Adar [61]. Supervised learning techniques exploited a collection of such extracted pairwise features to train classifiers and ranking models [257, 258]. However, computing pairwise heuristics *on-the-fly* for every potential link, is *infeasible* in large-scale evolving social networks.

More recently, *graph embedding* methods to learn latent node representations in graphs that capture the structural properties of a node and its neighborhoods, gained popularity owing to the scalability and efficacy of skip-gram models [259]. In particular, graph embedding models like node2vec [18] and Deepwalk [17] learn unsupervised node embeddings to maximize the likelihood of co-occurrence in fixed-length random walks, with applications in downstream tasks such as node classification and link prediction.

Graph embedding methods cannot directly incorporate node/link features, and more importantly learn latent embeddings for each node; this implies that number of *model parameters* scales linearly with the *size of the graph*, which is *prohibitive* for large-scale social networks with over multiple millions of users.

Graph Neural Networks: GNNs learn node representations by recursively propagating features (*i.e.*, message passing) from local neighborhoods through the use of aggregation and activation functions [153]. A key feature of GNNs is their ability to learn node representations combining graph topology and node/link attributes. Graph Convolutional Networks (GCNs) [39] learn degree-weighted neighborhood aggregators by operating on the graph Laplacian. Many models generalize GCN with a wide range of learnable aggregators, *e.g.*, self-attentions [40], mean and max pooling functions [41, 260]; these approaches have consistently outperformed embedding techniques based upon random walks [17, 18]. In contrast to most GNN models that store the entire graph in GPU memory, GraphSAGE [41] is an inductive variant that reduces memory footprint by sampling a fixed-size set of neighbors in each GNN layer. A few scalable extensions include minibatch training with variance reduction [261, 262], subgraph sampling [263], and graph clustering [264].

Despite the successes of GNNs in diverse applications, *very few* industrial systems have developed large-scale GNN implementations. One recent system, PinSage [5] extends GraphSAGE for Pinterest recommendation on a user-item bipartite graph. MultiSage [252] extends PinSage to multipartite user-item graphs.

However, GNNs remain unexplored for large-scale *user-user* social modeling applications where users exhibit multifaceted behaviors by interacting with different functionalities on social platforms. In our work, we design GNNs for the important application of *friend suggestion*, through a *novel multi-faceted friend ranking* formulation with *multi-modal user features* and *link communication features*.

Multi-Modal Learning: Deep learning techniques for multi-modal feature fusion over diverse modalities such as text, images, video, and graphs, have diverse applications [265, 266]. Specifically, multi-modal extensions of GNNs have been explored in micro-video recommendation [267] and urban computing [268] applications. In contrast to prior work that regard modalities as largely *independent* data sources, user feature modalities in social networks tend to be correlated. Exploiting knowledge from multiple channels have been shown to benefit recommendations in retail and e-commerce [269]. In this work, we propose a novel *cross-modality attention* layer that captures non-linear modality correlations, to learn *multi-faceted* user representations.

7.3 PRELIMINARIES

We first formulate the problem of *multi-faceted friend ranking* in large-scale social networking platforms (Section 7.3) and then briefly introduce relevant background and notations on *graph neural networks* (Section 7.3.2).

7.3.1 Problem Formulation

In this section, we introduce the different information sources in a social networking platform, that are relevant to friend recommendation. Each registered individual in the platform is denoted by a *user* u or v and a pair of users (u, v) may be connected by a *friendship*, which is an undirected relationship, *i.e.*, if u is a friend of v , v is also a friend of u . We assume that the social network has a set of N users \mathcal{V} introduced until our latest observation time of the platform. The friendship graph \mathcal{G} evolves when new friendships form and when new users join the platform. In this work, we only consider the emerging of new users and friendships while leaving the removal of existing users and edges as future work.

Prior work typically represent a dynamic network as a sequence of static snapshots, primarily due to scaling concerns. However, graph snapshots are coarse approximations of the actual continuous-time network and rely on a user-specified discrete time interval for snapshot creation [270, 271]. We also assume multiple time-aware user-level features (across modalities) and link(edge)-level features capturing pairwise user-user communications. In industrial settings, such features are commonly extracted by routine batch jobs and populated in an upstream database at regular time intervals (*e.g.*, daily batch inference jobs), to facilitate efficient model training.

Thus, we adopt a hybrid data model that achieves the best of both worlds. We formulate the *friendship graph* as a continuous-time dynamic graph (CTDG) with the expressivity to record friendships at the finest possible temporal granularity; and represent *features* as a sequence of daily snapshots where the time-sensitive features (*e.g.*, engagement activity) are recorded at different time scales.

Friendship Graph: Let us consider an observation time window (t_s, t_e) such that friendships created in this window specify the training data for the friend ranking model. We divide this window (t_s, t_e) into a sequence of S daily snapshots, denoted by $1, 2, \dots, S$. Formally, we model the friendship graph \mathcal{G} as a timed sequence of friend creation events over the entire time range $(0, t_e)$, defined as:

Definition 7.1 (Friendship Graph). Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, let \mathcal{V} be the set of users, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathbb{R}$ be the set of friendship links between users in \mathcal{G} . At the finest granularity, each link $e = (u, v, t) \in \mathcal{E}$ is assigned a unique timestamp $t \in \mathbb{R}^+$; $0 < t < t_e$ that denotes the link creation time, and $\mathcal{T} : \mathbb{R}^+ \mapsto [0, S]$ is a function that maps each timestamp t to a corresponding snapshot.

Note that the time window (t_s, t_e) corresponds to snapshots $[1, S]$, and snapshot 0 is a placeholder for any $t < t_s$. However, friendship graph \mathcal{G} includes the entire set of friendships

created with all time-stamped links in $(0, t_e)$. The set of temporal neighbors of user v at time t includes all friends created before t , defined as $N_t(v) = \{w : e = (v, w, t') \in \mathcal{E} \wedge t' < t\}$.

Multi-Modal Evolving User Features: In a social network, users typically use multiple features, such as posting videos, exchanging messages or pictures with friends, or liking and sharing posts and articles, which can be indicative of both their stable traits and mutable interests. Since users with shared interests tend to form social connections due to homophily [11], we extract user attributes spanning a total of $K = 4$ modalities, which include profile attributes, in-app interests, friend creation activities, and user engagement activities, described in detail below:

Profile Attributes: a set of (mostly) static demographic features describing the user, including age, gender, languages, etc., that are listed or inferred from their profile.

Content Interests: a real-valued feature vector describing the textual content (*e.g.*, posts, news) interacted by the user within the platform, *e.g.*, topics of posts liked by the user on Facebook.

Friending Activity: this feature modality records the aggregated number of friend requests sent/received, reciprocated friendships, and viewed suggestions made by the user in different time ranges (*e.g.*, daily, weekly, and monthly).

Engagement Activity: this modality records the aggregated number of in-app direct and indirect engagements made by the user (*e.g.*, direct messages, pictures, and comments on posts) with other friends in different time ranges.

The user feature modalities include a combination of *static* and *time-sensitive* features, *i.e.*, the profile attributes are static while the rest of the modalities are time-sensitive and often evolve at different scales across users, *e.g.*, a long-time active user may frequently communicate with a stable set of friends, while a new user is more likely to quickly add new friends before communicating.

The feature vector of a user $u \in \mathcal{V}$ in snapshot $s \in [1, S]$ is defined by $\mathbf{x}_v^s = [\mathbf{x}_v^{s,1}, \dots, \mathbf{x}_v^{s,K}]$ where $\mathbf{x}_v^{s,k} \in \mathbb{R}_k^D$ is the k -th user feature modality and $[\cdot]$ denotes row-wise concatenation.

Pairwise Link Communication Features: Social networks evolve centering around communication artifacts, with two predominant types of online communication channels: *conversations* and *social actions*. Conversations can include exchanges of text messages, pictures, and video content with friends, which indicate direct user-user interactions via communication. In contrast, social actions facilitate indirect user interactions through a variety of actions, *e.g.*, posting a Snapchat Story (image or video content) or liking a Facebook post, results in a passive broadcast to friends.

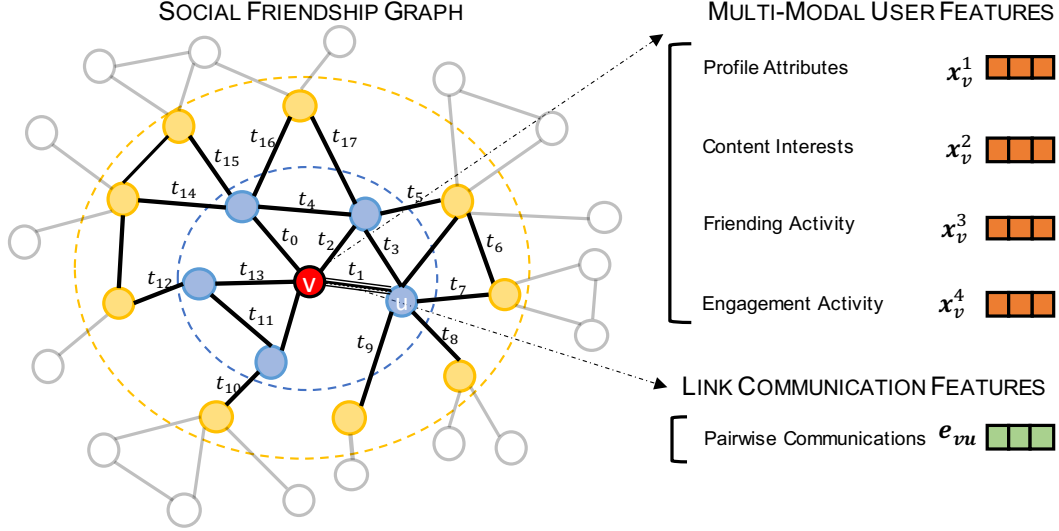


Figure 7.1: *Desiderata for Multi-faceted Friend Ranking: temporally evolving friendship graph with multi-modal user features and pairwise link communication features.*

For conversational channels, we extract bidirectional edge features between users and their friends reflecting the number of communications sent and received by each pair of users. We capture indirect social actions by recording the number of user actions associated with each friend. Similar to user features, we extract edge features per snapshot by aggregating communications at different time intervals. The link feature vector for a pair of users (u, v) at time t (who became friends before t), is denoted by $\mathbf{e}_{uv}^s \in \mathbb{R}^F$ where $s = \mathcal{T}(t)$ is the snapshot corresponding to timestamp t , and E is the number of extracted link communication features.

We now formally define the problem of *multi-faceted friend ranking* in large-scale social platforms, over friendship graph \mathcal{G} with multi-modal user features and pairwise link features.

Problem 7.1 (Multi-Faceted Friend Ranking). Our objective is to leverage the multi-modal user features $\{\mathbf{x}_v^s : v \in \mathcal{V}, 1 \leq s \leq S\}$, link communication features $\{\mathbf{e}_{uv}^s : s = \mathcal{T}(t), (u, v, t) \in \mathcal{E}\}$ and the structure of the friendship graph \mathcal{G} , to generate temporal user representations $\{\mathbf{h}_v(t) \in \mathbb{R}^D : v \in \mathcal{V}\}$ at time t , that can be used for friend ranking, *e.g.*, via nearest-neighbor lookup for candidate retrieval, or friend candidate re-ranking.

7.3.2 Background on GNNs

We briefly introduce a generic formulation of a graph neural network layer with message-passing for neighborhood aggregation.

GNNs use multiple layers to learn node representations. At each layer $l > 0$ ($l = 0$ is the input layer), GNNs compute a representation for node v by aggregating features from its

neighborhood, through a learnable aggregator $F_{\theta,l}$ per layer. Stacking k layers allows the k -hop neighborhood of a node to influence its representation.

$$\mathbf{h}_{v,l} = F_{\theta,l}(\mathbf{h}_{v,l-1}, \{\mathbf{h}_{u,l-1}\}), \quad u \in N(v) \quad (7.1)$$

Equation (7.1) indicates that the node embedding $\mathbf{h}_{v,l} \in \mathbb{R}^D$ for node v at the l -th layer is a non-linear aggregation $F_{\theta,l}$ of the embeddings from layer $l-1$ of node v and the embeddings of immediate network neighbors $u \in N(v)$ of node v . The function $F_{\theta,l}$ defines the message passing function at layer l and can be instantiated using a variety of aggregator architectures, including graph convolution [39], graph attention [40], and pooling [41]. The node representation for v at the input layer is $\mathbf{h}_{v,0}$, where $\mathbf{h}_{v,0} = \mathbf{x}_v$ and $\mathbf{x}_v \in \mathbb{R}^D$. The representation of node v at the final GNN layer is typically trained using a supervised objective.

The above formulation (Equation 7.1) operates under the setting of a static graph and assumes a single static input feature vector for each node. In contrast, our social network has a time-evolving friendship graph \mathcal{G} with pairwise link features and multi-modal node (user features), which poses unique modeling challenges. In the next section, we present a graph neural network model for multi-faceted friend ranking.

7.4 GRAPH NEURAL FRIEND RANKING

In this section, we present our approach to inductively learn user representations in a dynamic friendship graph with pairwise link features and multi-modal node features.

We first conduct an empirical analysis on user feature modalities, to gain insights into various factors impacting friendship formation (Section 7.4.1). We then formulate the design choices of our model GRAFRANK for friend ranking grounded on our acquired insights (Section 7.4.2), followed by model training details (Section 7.4.3)

7.4.1 Motivating Insight: Modality Analysis

We conduct an empirical study that helps us formulate the design choices in our model. We aim to validate the existence and understand the extent and variance of attribute homophily with respect to the different user feature modalities. We begin by analyzing users' ego-networks to characterize *modality homophily*, both overall and broken-down across different user segments. The definition of *modality homophily* echoes the standard definition of attribute homophily [11], but generalized to include a modality of attributes, *i.e.*, the tendency of users in a social graph to associate with others who are *similar* to them along

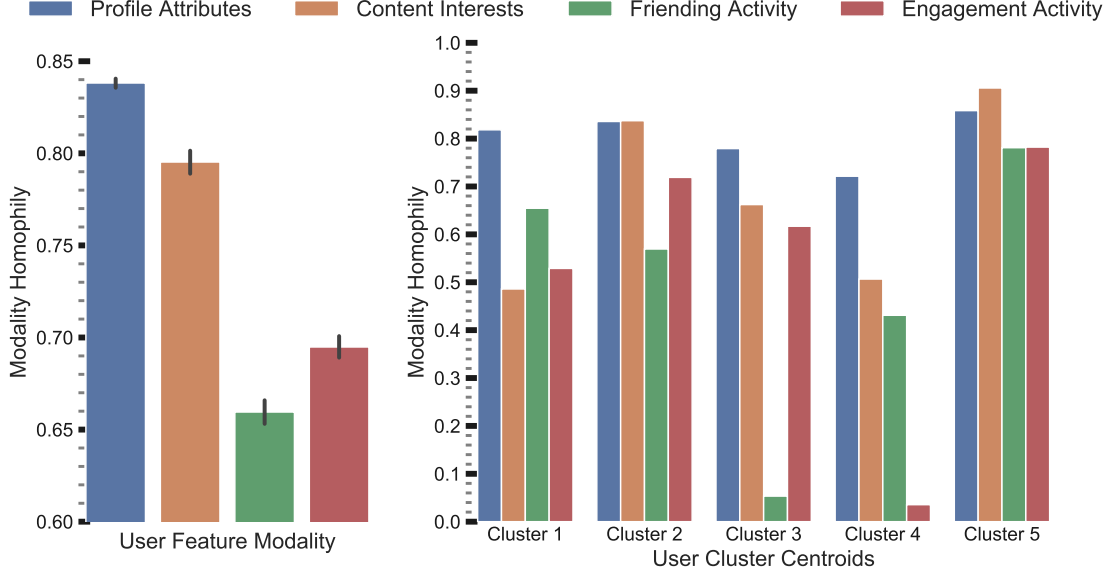


Figure 7.2: *Users exhibit different extents of homophily across feature modalities. (a) Overall modality homophily scores, with 95% confidence interval bands (b) five representative cluster centroids identified by clustering users based on their homophily distributions over the K modalities.*

attributes of a certain modality.

We define a homophily measure \mathbf{m}_{vu}^k between a user v and her friend u on modality k by the standard cosine similarity [272], which is a normalized metric that accounts for heterogeneous activity across users. We compute a modality homophily score \mathbf{m}_u^k for user v on modality k by the mean over all her neighbors, defined by:

$$\mathbf{m}_v^k = \frac{1}{|N_v|} \sum_{u \in N_v} \mathbf{m}_{vu}^k \quad \mathbf{m}_{vu}^k = \cos(\mathbf{x}_u^k, \mathbf{x}_v^k) \quad (7.2)$$

Note that, we omit the snapshot s above since the discussion is restricted to a single feature snapshot. Figure 7.2 (a) shows the overall modality homophily scores (averaged across all users), for each of the K modalities. We observe differing extents of attribute homophily across modalities, with higher variance for the time-sensitive modalities (*e.g.*, friending and engagement activities).

We further extend our analysis to examine the *homophily distribution* over modalities at the granularity of individual users, to understand if modality homophily varies across different users. To study this, we first represent each user u by a K -dimensional *modality vector* $\mathbf{m}_u = [\mathbf{m}_u^1, \dots, \mathbf{m}_u^K]$ that describes the homophily distribution over the K feature modalities. We then use k -means [273] to cluster the set of all users based on their computed modality

vectors; Figure 7.2 (b) shows the centroid vectors of five representative clusters identified by k -means. We observe stark differences in the modality vector centroids across the five clusters, indicating the existence of user segments with diverse homophily distributions over the K modalities. This motivates our first key observation:

Observation 7.1 (Heterogeneity in Modality Homophily). Users exhibit different extents of homophily across feature modalities, and the homophily distribution over modalities varies significantly across diverse user segments.

Each modality enables identification of a subset of friends that exhibit modality homophily. However, this poses a question: do the K modalities induce the same (or disparate) subsets of homophilous friends, or are the friends that exhibit homophily in each modality, correlated? We investigate this relationship in this section.

For every feature modality k , we cluster the ego-network (set of direct friends) $N(v)$ of each user $v \in \mathcal{V}$ by representing it as a set of modality-specific feature vectors $\{\mathbf{x}_u^k \in \mathbb{R}^{D_k} : u \in N(v)\}$; this results in ego-clustering assignments for each modality k . To quantify *cross-modality correlations*, we compute a *correlation* score for each pair of modalities, measured by the *consensus* between their corresponding ego-clustering assignments.

We use two standard measures: Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) to evaluate consensus between clusterings [274]. NMI measures the statistical correlation between two clustering assignments; however, NMI increases with the number of distinct clusters. ARI measures the percentage of correct pairwise assignments, and is chance-corrected with an expected value of zero. Note that NMI and ARI are symmetric metrics.

Figure 7.3 depicts average NMI and ARI scores for each pair of feature modalities. We observe substantial correlation in cluster assignments across a few modalities (*e.g.*, time-sensitive modalities M_1 and M_3) while some (*e.g.*, static modality M_0) are quite distinct from the rest. Our key takeaway regarding modality correlation is:

Observation 7.2 (Cross-Modality Correlation). There exist non-trivial correlations between pairs of feature modalities, indicated by the consensus in their induced ego-network clustering of users.

7.4.2 GRAFRANK: Multi-Faceted Friend Ranking

In this section, we first introduce the key components of our model GRAFRANK (Graph Attention Friend Ranker) for friend ranking. Our modeling choices in designing a multi-modal GNN, follow from our observations. GRAFRANK has three modules (Figure 7.5):

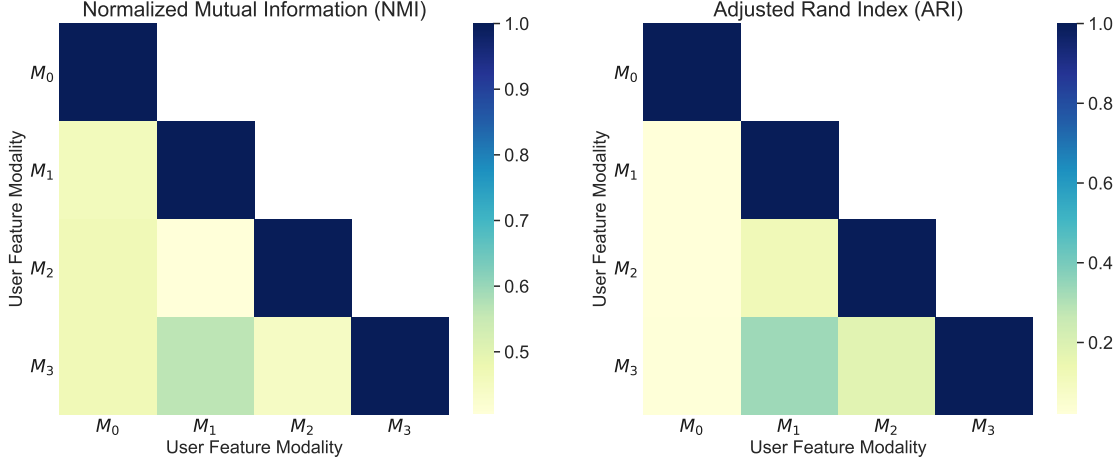


Figure 7.3: *Cross-modality Correlation Study: NMI (a) and ARI (b) metrics for each pair of modalities, quantifying pairwise correlation by consensus in ego-clustering assignments (obtained independently with respect to each modality). We observe substantial correlations across pairs of modalities.*

- Modality-specific neighbor aggregation.
- Cross-modality attention layer.
- Pairwise ranking objective for model training.

Below, we present a detailed description of each module.

Modality-specific Neighbor Aggregation

Since the feature modalities vary in the extent of induced homophily (Observation 7.1), we treat each modality individually as opposed to the popular choice of directly combining multi-modal user features by concatenation. Thus, our objective is to learn a modality-specific representation $\mathbf{z}_v^k(t) \in \mathbb{R}^D$ for each user $v \in \mathcal{V}$ at time $t \in \mathbb{R}^+$, that encapsulates information exclusively from modality k . We endow each user v with the expressivity to flexibly prioritize different friends in her temporal neighborhood $N_t(v)$, hence accounting for the variance in homophily distribution across user segments.

To achieve this, we design a *modality-specific neighbor aggregation* module to compute K representations $\{\mathbf{z}_v^1(t), \dots, \mathbf{z}_v^K(t)\}$, $\mathbf{z}_v^k(t) \in \mathbb{R}^D$ for each user v at time $t \in \mathbb{R}^+$, where each $\mathbf{z}_v^k(t)$ is obtained using an independent and unique message-passing function per modality. We begin by describing a single layer, which consists of two major operations: *message propagation* and *message aggregation*. We then discuss generalization to multiple successive layers.

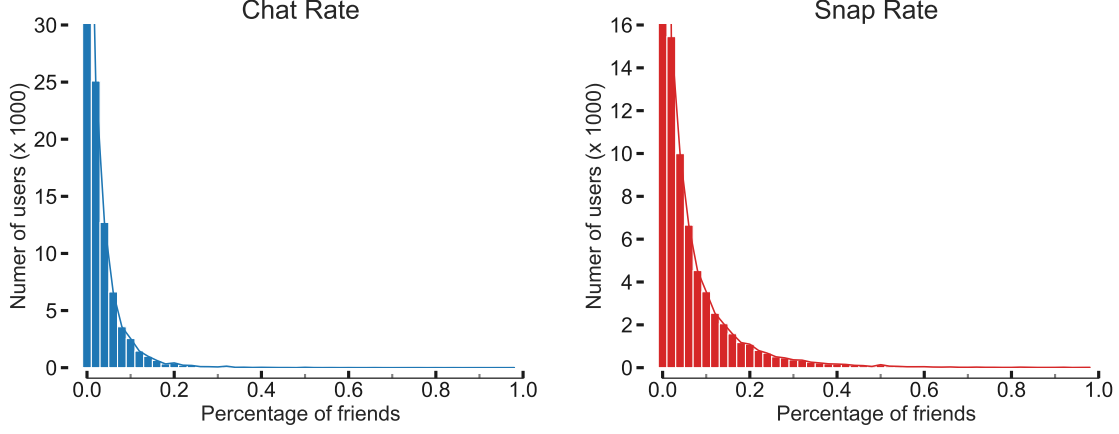


Figure 7.4: *Friend communication rate in two direct channels (Chat, Snap) on Snapchat. Most users communicate frequently only with a subset ($\leq 20\%$) of their friends, making influence modeling critical during neighbor aggregation.*

Message Propagation: We define the message-passing mechanism to aggregate information from the ego-network $N_t(v)$ of user v at time t . Specifically, the propagation step for modality k aggregates the k -th modality features $\{\mathbf{x}_u^{s,k} : u \in N_t(v), s = \mathcal{T}(t)\}$ from the corresponding snapshot $s = \mathcal{T}(t)$ of temporal neighbors $N_t(v)$. To quantify the importance of each friend u in the ego-network, we propose a friendship attention [40] which takes embeddings $\mathbf{x}_u^{s,k}$ and $\mathbf{x}_v^{s,k}$ as input, and computes an attentional coefficient $\alpha^k(v, u, t)$ to control the influence of friend u on v at time t .

$$\alpha^k(v, u, t) = \text{LeakyReLU}\left(\mathbf{a}_k^T (\mathbf{W}_1^k \mathbf{x}_v^{s,k} \parallel \mathbf{W}_1^k \mathbf{x}_u^{s,k})\right) \quad s = \mathcal{T}(t) \quad (7.3)$$

where $\mathbf{W}_1^k \in \mathbb{R}^{D_k \times D}$ is a shared linear transformation applied to each user, \parallel is the concatenation operation, and the friendship attention is modeled as a single feed-forward layer parameterized by weight vector $\mathbf{a}_k \in \mathbb{R}^{2D}$ followed by the LeakyReLU nonlinearity. We then normalize the attentional coefficients across all friends connected with v at time t by adopting the softmax function:

$$\alpha^k(v, u, t) = \frac{\exp(\alpha^k(v, u, t))}{\sum_{w \in N_t(v)} \exp(\alpha^k(v, w, t))} \quad (7.4)$$

We now define an ego-network representation $\mathbf{z}_v^k(t, N_v(t)) \in \mathbb{R}^{D_k}$ for user v in modality k that captures messages propagated from first-order neighbors in the ego-network $N_v(t)$. The message $\mathbf{m}_{v \leftarrow u}^k \in \mathbb{R}^D$ propagated from friend u to user v at time t is defined as the transformed friend embedding, *i.e.*, $\mathbf{m}_{v \leftarrow u}^k = \mathbf{W}_1^k \mathbf{x}_u^{s,k}$. We then compute $\mathbf{z}_v^k(t, N_v(t))$ through a weighted average of message embeddings from each friend $u \in N_v(t)$ and guided by normalized

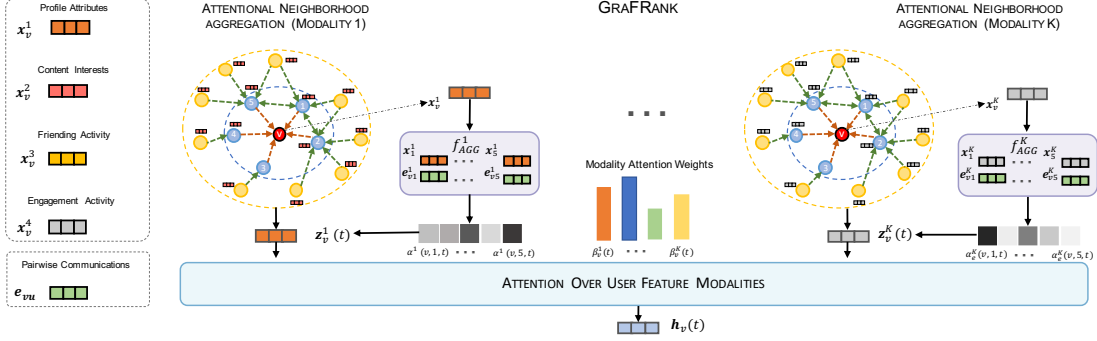


Figure 7.5: Overall framework of GRAFRANK: a set of K modality-specific neighbor aggregators (parameterized by individual modality-specific user features and link communication features) to compute K intermediate user representations; cross-modality attention layer to compute final user representations by capturing discriminative facets of each modality.

friendship weights $\alpha^k(u, v, t)$, which is defined as:

$$\mathbf{z}_v^k(t, N_v(t)) = \sum_{u \in N_v(t)} \alpha^k(v, u, t) \mathbf{m}_{v \leftarrow u}^k \quad (7.5)$$

While the above equation specifies different contributions to friends, these weights are learnt merely based on the connectivity structure of the ego-network. In reality, most users only have a few close friends, and users with many friends only frequently communicate with a few of them. To empirically validate this hypothesis, we compute the *friend communication rate* of all users, defined by the percentage of friends that a user has directly communicated with at least once (directly sent a Chat/Snap with on Snapchat) in a one-month window. From Figure 7.4, we find that a vast majority of users communicate primarily with a small percentage (10-20%) of their friends; thus, we posit that *friendship activeness* is critical to precisely model user affinity.

Towards this goal, we incorporate *pairwise link communication* features to parameterize both the attentional coefficients and the message aggregated from friends in the ego-network. Specifically, we formulate the message $\mathbf{m}_{v \leftarrow u}^k \in \mathbb{R}^D$ from friend u to user v at time t as a function of both friend feature $\mathbf{x}_u^{s,k}$ and link feature \mathbf{e}_{vu}^s .

$$\mathbf{m}_{v \leftarrow u}^k = \mathbf{W}_2^k \mathbf{x}_u^{s,k} + \mathbf{W}_e^k \mathbf{e}_{vu}^s + \mathbf{b} \quad s = \mathcal{T}(t) \quad (7.6)$$

where $\mathbf{W}_2^k \in \mathbb{R}^{D_k \times D}$, $\mathbf{W}_e^k \in \mathbb{R}^{E \times D}$ are trainable weight matrices operating on the user and link features respectively, and $\mathbf{b} \in \mathbb{R}^D$ is a learnable bias vector. The attentional coefficient $\alpha^k(v, u, t)$ is then computed as a function of user feature $\mathbf{x}_v^{s,k}$ and message embedding

$\mathbf{m}_{v \leftarrow u}^k \in \mathbb{R}^D$ from friend u to user v , defined by:

$$\alpha^k(v, u, t) = \sigma \left(\mathbf{a}_k^T \left(\mathbf{W}_1^k \mathbf{x}_v^{s,k} \parallel \mathbf{m}_{v \leftarrow u}^k \right) \right) \quad s = \mathcal{T}(t) \quad (7.7)$$

where σ is a non-linearity such as LeakyRELU. We similarly normalize the attentional coefficients $\alpha^k(v, u, t)$ using Equation 7.4 and compute the ego-network representation $\mathbf{z}_v^k(t, N_v(t))$ for user v on modality k using Equation 7.5 with the message embedding $\mathbf{m}_{v \leftarrow u}^k$ from Equation 7.6 conditioned on link features.

Distinct from conventional graph convolution and attention networks [5, 39, 40, 41] that only consider user $\mathbf{x}_v^{s,k}$ and friend $\mathbf{x}_v^{s,k}$ features to parameterize the neighbor aggregation, we additionally encode the link communication features \mathbf{e}_{vu}^s into the attentional co-efficient $\alpha^k(v, u, t)$ (Equation 7.7) and the message $\mathbf{m}_{v \leftarrow u}^k$ (Equation 7.6) passed from friend u ; this encourages the aggregation to be cognizant of the pairwise communications with friends, *e.g.*, passing more messages from the active friendships. In addition, we observe a boost in friend ranking performance (Section 7.5.4) owing to our novel communication-aware message passing strategy.

Message Aggregation: We refine the latent representation for user v by aggregating the messages propagated from friends in $N_v(t)$. In addition to the messages from $N_v(t)$, we additionally consider the self-connection of v , *i.e.*, $\mathbf{m}_{v \leftarrow v}^k = \mathbf{W}_1^k \mathbf{x}_v^{s,k}$ to preserve knowledge of the original features (\mathbf{W}_1 is the same transformation used in Equation 7.3). Specifically, we concatenate the ego-network and self-representations of user v , and further transform the concatenated embedding through a dense layer F_θ^k , defined by:

$$\mathbf{z}_v^k(t) = F_\theta^k \left(\mathbf{m}_{v \leftarrow v}^k, \mathbf{z}_v^k(t, N_v(t)) \right) \quad (7.8)$$

$$= \sigma \left(\mathbf{W}_a^k \left(\mathbf{z}_v^k(t, N_v(t)) \parallel \mathbf{m}_{v \leftarrow v}^k \right) + \mathbf{b}_a \right) \quad (7.9)$$

where $\mathbf{W}_a^k \in \mathbb{R}^{D \times D}$, $\mathbf{b}_a \in \mathbb{R}^D$ are trainable weight parameters of the message aggregator, and σ denotes the ELU activation function, which allows messages to encode both positive and small negative signals. Empirically we observe significant gains owing to retaining knowledge of self-representation via concatenation, instead of directly using the propagated ego-network representation from the neighborhood, as in GCN [39], and GAT [40].

Higher-order Propagation: We stack multiple aggregation layers to model high-order connectivity information, *i.e.*, propagate features from l -hop neighbors. The inputs to layer l depend on the user representations output from layer $(l - 1)$ where the initial (*i.e.*, “layer

0”) representations are set to the input user features in modality k . By stacking l layers, we recursively formulate the representation $\mathbf{z}_{v,l}^k$ of user v at the end of layer l by:

$$\mathbf{z}_{v,l}^k = F_{\theta,l}^k(\mathbf{m}_{v \leftarrow v,l-1}^k, z_{v,l-1}^k(t, N_v(t))) \quad \mathbf{m}_{v \leftarrow v,l-1}^k = z_{v,l-1}^k \quad (7.10)$$

where $z_{v,l-1}^k$ is the representation of user v in modality k after $(l-1)$ layers and $z_{v,l-1}^k(t, N_v(t))$ denotes the $(l-1)$ -ego-network representation of user v . We apply L neighbor aggregation layers to generate the layer- L representation $\mathbf{z}_{v,L}^k$ of user v in modality k .

Cross Modality Attention

To learn complex non-linear correlations across feature modalities, we design a *cross-modality attention* mechanism. Specifically, we learn modality attention weights $\beta_v^k(t)$ to distinguish the influence of each modality k using a two-layer Multi-Layer Perceptron, by:

$$\beta_v^k(t) = \frac{\exp(\mathbf{a}_m^T \mathbf{W}_m \mathbf{z}_{v,L}^k + b_m)}{\sum_{k'=1}^K \exp(\mathbf{a}_m^T \mathbf{W}_m \mathbf{z}_{v,L}^{k'} + b_m)} \quad (7.11)$$

with weights $\mathbf{W}_m \in \mathbb{R}^{D \times D}$, $\mathbf{a}_m \in \mathbb{R}^D$ and scalar bias b_m . The final representation $\mathbf{h}_v(t) \in \mathbb{R}^D$ of user v is computed by a weighted aggregation of the layer- L modality-specific user representations $\{\mathbf{z}_{v,L}^1, \dots, \mathbf{z}_{v,L}^K\}$, guided by modality weights $\beta_v^k(t)$, defined by:

$$\mathbf{h}_v(t) = \sum_{k=1}^K \beta_v^k(t) \mathbf{W}_m \mathbf{z}_{v,L}^k \quad (7.12)$$

7.4.3 Model Training

We train GRAFRANK using a pairwise ranking objective to differentiate *positive* and *negative* neighborhoods. We assume access to training data described by a set of timestamped links \mathcal{L} created in a time span (t_s, t_e) , where $(v, u, t) \in \mathcal{L}$ indicates a bi-directional link between *source* user u and *target* friend v formed at time $t \in (t_s, t_e)$. To learn the parameters of GRAFRANK, we define a triplet like training objective based on max-margin ranking.

Pairwise Ranking Objective

We define a time-sensitive ranking loss over the user embeddings $(\mathbf{h}_v(t))$ for user v at time t to rank the inner product of positive links $(v, u, t) \in \mathcal{L}$, higher than sampled negatives

(v, n, t) by a margin Δ , as:

$$L = \sum_{(u,v,t) \in \mathcal{L}} \mathbb{E}_{n \sim P_n(v)} \max\{0, \mathbf{h}_v(t) \cdot \mathbf{h}_n(t) - \mathbf{h}_v(t) \cdot \mathbf{h}_u(t) + \Delta\} \quad (7.13)$$

where Δ is a margin hyper-parameter and $P_n(v)$ is the negative sampling distribution for user v . This generic contrastive learning formulation enables usage of the same framework to learn user embeddings for different recommendation tasks such as candidate retrieval and ranking, with diverse negative sampling distributions.

Friend Candidate Retrieval and Ranking Tasks

We learn user embeddings towards two key use-cases in friend ranking: *candidate retrieval* and *candidate ranking*. Candidate retrieval aims to generate a list of top- N (e.g., $N = 100$) potential friend suggestions out of a very large candidate pool (over millions of users), while candidate ranking involves fine-grained re-ranking within a much smaller pool of the generated candidates, to determine the top- n (e.g., $n = 10$) suggestions shown to end users in the platform. We define different negative sampling distributions $P_n(v)$ for each task owing to their different ranking granularities, as follows:

Candidate Retrieval: For the *coarse-grained* task of candidate retrieval, we uniformly sample five *random negative* users for each positive link, from the entire user set \mathcal{V} . Generating random negatives is efficient and effective at quickly training the model to identify potential friend candidates for each user. However, random negatives are often too *easy* to distinguish and may not provide the requisite resolution for the model to learn *fine-grained distinctions* necessary for candidate friend ranking.

Candidate Ranking: To enhance model resolution for candidate ranking, we also consider *hard negative* examples for each positive pair (u, v) that are related to the source u , but not as relevant as the target friend v . For each user u , we generate five hard negatives based on graph proximity. Specifically, users in k -hop ($3 \leq k \leq 4$) graph neighborhoods of u (that are not within the 2-hop neighborhood), are randomly sampled as hard negatives. In practice, we pre-compute a set of hard negatives for each user, to facilitate efficient negative sampling during model training.

We adopt a *two-phase* learning approach for candidate ranking. We pre-train the model on random negatives (as in candidate retrieval), to identify good model initialization points, followed by fine-tuning on hard negatives. Ranking hard negatives is more

challenging, hence encouraging the model to progressively learn friend distinctions at a finer granularity. We empirically show notable gains on candidate ranking due to our two-phase strategy, compared to training individually on random or hard negatives.

Temporal Neighborhood Sampling

We learn a temporal user representation $\mathbf{h}_v(t)$ for user v at time t by selecting a fixed number of friends from $N_v(t)$ for neighbor aggregation at each layer; this controls the memory footprint during training [41].

To efficiently identify and sample neighbors of v at any time t , we represent the time-evolving friendship graph \mathcal{G} as a *temporal adjacency list* at its latest time t_s where each user v has a list of (friend, time) tuples sorted by link creation times. This data representation enables $O(\log d)$ neighbor lookup at an arbitrary timestamp t via binary search where d is the average user degree in the graph.

Multi-GPU Minibatch Training

We train GRAFRANK with minibatches of links from \mathcal{L} using multiple GPUs on a single shared memory machine. The temporal adjacency list of \mathcal{G} and feature matrices \mathbf{X}, \mathbf{E} are placed in shared CPU memory to enable fast parallel neighborhood sampling and feature lookup. We adopt a *producer-consumer* architecture [5] that alternates between CPUs and GPUs for model training. A CPU-bound producer constructs friend neighborhoods, looks up user and link features, and generates negative samples for the links of a minibatch. We then partition each minibatch across multiple GPUs, to compute forward passes and gradients with a PyTorch model over dynamically constructed computation graphs. The gradients from different GPUs are synchronized using PyTorch’s Distributed Data Parallel [275].

7.5 EXPERIMENTS

To analyze the quality of user representations learned by GRAFRANK, we propose five research questions to guide our experiments:

- (RQ₁) Can GRAFRANK outperform feature-based models and state-of-the-art GNNs on candidate *retrieval* and *ranking* tasks?
- (RQ₂) How does GRAFRANK compare to prior work under alternative metrics of *reciprocated* and *communicated* friendships?

Dataset	Region 1	Region 2
# users	3.1 M	17.1 M
# links	286 M	2.36 B
# user features	79	79
# link features	6	6
# test set friend requests	46K	340K

Table 7.1: *Dataset statistics*

- (RQ₃) How do different *architectural* design choices and *training* strategies in GRAFRANK impact performance?
- (RQ₄) How do training strategies and hyper-parameters impact convergence and performance of GRAFRANK?
- (RQ₅) How do the learned user embeddings in GRAFRANK perform across diverse user cohorts?

7.5.1 Experimental Setup

We now present our experimental setup with a brief description of datasets, evaluation metrics, and model training details.

Datasets

We evaluate the user representations learned by GRAFRANK for friend recommendations, on two large-scale datasets from Snapchat. Each dataset is constructed from the interactions among Snapchat users belonging to a specific country (details obscured for privacy reasons). In total, we collect 79 user features spanning four feature modalities and six pairwise link communication features, as described in Section 7.3. All features are standardized using zero-mean and unit-variance normalization before model training. We define the set \mathcal{L} of positive training examples (Equation 7.13) using all timestamped friendships created over a span of one week. Since our objective is to evaluate the quality of friendship recommendations, the test set comprises all *friend add requests* generated by users over the next four days. We use 10% of the labeled examples as a validation set for hyper-parameter tuning. Table 7.1 shows the detailed statistics of the two datasets from Snapchat.

Evaluation Metrics

We experiment on two friend recommendation tasks: candidate retrieval and candidate ranking, as outlined in Section 7.4.3. To evaluate friend recommendation, we use standard ranking metrics Hit-Rate (HR@K), Normalized Discounted Cumulative Gain (NDCG@K) and Mean Reciprocal Rank (MRR). We adopt the standard negative-sample evaluation [88] to generate N negative samples per positive pair of users (u, v) in the test set (user u has sent a friend request to user v). We then compute ranking metrics for each test pair (u, v) by ranking v among the N negative samples via inner products in the latent space.

To evaluate candidate retrieval, we use $N = 10000$ *randomly sampled negative users* for each positive test pair, to emulate retrieval from a large candidate pool. Ideally, candidate ranking should operate over a shortlisted list of potential friends identified by the retrieval system. However, we aim to provide a fair benchmark comparison of different models for candidate ranking that is agnostic to the biases of the upstream retrieval system. Thus, we instead generate $N = 500$ *hard negatives samples* per test pair based on K -hop neighborhoods (Section 7.4.3), to ensure an unbiased comparison.

Training Details

We train GRAFRANK using $L = 2$ message passing layers per feature modality with a hidden dimension size of 64 and output embedding dimension $D = 32$. In each layer, we sample 15 first-order neighbors and 15 second-order neighbors for each sampled first-order neighbor; each user receives messages propagated from upto 225 friends. During model training, we apply dropout with rate of 0.3 in the two aggregation layers. The model is trained for a maximum of 30 epochs with a batch size of 256 positive pairs (apart from 5 random/hard negatives per pair) and learning rate of 0.001 using Adam optimizer. We benchmark our experiments using a machine with 32 cores, 200 GB shared CPU memory, and a single Nvidia Tesla P100 GPU with PyTorch [275] implementations on the Linux platform.

7.5.2 Baselines

We compare the performance of GRAFRANK on friend ranking against strong feature-based machine learning models, and state-of-the-art graph neural networks that learn user representations.

- **LogReg** [276]: Logistic regression classifier for link prediction. The input feature for

each pair of users, is a concatenation of source and target features across the K user feature modalities.

- **XGBoost** [277]: Tree boosting model for pairwise learning to rank, trained using the same input features as LogReg. It is currently deployed at Snapchat for quick-add friend suggestions.
- **MLP** [276]: Two-layer Multi-Layer Perceptron with fully-connected layers and ReLU activations to learn user representations. We train MLP using the same ranking loss as our model (Equation 7.13).
- **GCN** [39]: Graph convolutional networks with degree-weighted neighbor aggregation and neighborhood sampling for scalable training. For each user, we concatenate features across the K user feature modalities into a new nodal feature vector.
- **GAT** [40]: Graph attention networks with self-attentional aggregation and neighborhood sampling for scalable training.
- **SAGE + Max** [41]: Element-wise max pooling for neighbor aggregation and self-embedding concatenation at each layer.
- **SAGE + Mean** [41]: Same as SAGE + Max with element-wise mean pooling function for neighbor aggregation.

Note that graph autoencoders [278] and graph convolutional matrix completion models [279] are not empirically comparable because they cannot scale to our large-scale social network datasets.

For each baseline, we train separate models for candidate retrieval and ranking tasks. We use random negatives for retrieval model training while resorting to hard negatives for ranking; we empirically find training separate models to be vastly superior to a training a single model using a mixture of random and hard negatives, or even curriculum training [5].

7.5.3 Experimental Results

We first present our main empirical results comparing our proposed model GRAFRANK against competing baselines on candidate retrieval and ranking tasks, followed by additional comparisons using alternative measures of friendship quality.

Dataset	Region 1					Region 2				
Metric	N@5	N@50	HR@5	HR@50	MRR	N@5	N@50	HR@5	HR@50	MRR
LogReg	0.1752	0.2460	0.2452	0.5262	0.1751	0.0761	0.1367	0.1134	0.3654	0.0831
MLP	0.1923	0.2679	0.2721	0.5726	0.1903	0.0973	0.1720	0.1466	0.4541	0.1046
XGBoost	0.2099	0.2865	0.2932	0.5957	0.2071	0.1366	0.2097	0.1936	0.4921	0.1409
GCN	0.0934	0.1836	0.1490	0.5154	0.1034	0.1651	0.2634	0.2503	0.6427	0.1678
GAT	0.0851	0.1813	0.1424	0.5352	0.0960	0.1797	0.2794	0.2698	0.6663	0.1812
SAGE + Max	0.1790	0.2736	0.2695	0.6409	0.1797	0.1520	0.2505	0.2315	0.6269	0.1566
SAGE + Mean	0.2378	0.3240	0.3338	0.6757	0.2333	0.2870	0.3805	0.4005	0.7655	0.2790
GraFRank	0.3152	0.3983	0.4318	0.7533	0.3035	0.4166	0.4950	0.5386	0.8395	0.4012
Percent Gains	32.55%	22.93%	29.36%	11.48%	30.09%	45.16%	30.09%	34.48%	9.67%	43.8%

Table 7.2: GRAFRANK outperforms feature-based models and GNNs (relative gains of 30-43 % MRR with respect to the best baseline) on candidate retrieval in Regions 1 and 2. HR@K and N@K denote Hit-Rate@K and NDCG@K metrics for $K = 5, 50$.

Dataset	Region 1					Region 2				
Metric	N@5	N@10	HR@5	HR@10	MRR	N@5	N@10	HR@5	HR@10	MRR
LogReg	0.1521	0.1795	0.2268	0.3116	0.1523	0.1398	0.1711	0.2136	0.3106	0.1449
MLP	0.1873	0.2190	0.2663	0.3644	0.1915	0.1927	0.2241	0.2721	0.3695	0.1967
XGBoost	0.1714	0.2002	0.2394	0.3287	0.1779	0.1844	0.2174	0.2605	0.363	0.1911
GCN	0.1345	0.1698	0.2039	0.3136	0.1462	0.1758	0.2147	0.2619	0.3826	0.1831
GAT	0.1416	0.1776	0.2197	0.3313	0.1503	0.2028	0.2445	0.2984	0.4276	0.2077
SAGE + Max	0.2063	0.2441	0.2980	0.4151	0.2094	0.2426	0.2818	0.3443	0.4654	0.2426
SAGE + Mean	0.2232	0.2607	0.3165	0.4330	0.2255	0.2766	0.3164	0.3835	0.5064	0.2744
GraFRank	0.2684	0.3098	0.3772	0.5051	0.2669	0.3342	0.3767	0.4529	0.5841	0.3282
Percent Gains	20.25%	18.83%	19.18%	16.65%	18.36%	20.82%	19.06%	18.1%	15.34%	19.61%

Table 7.3: GRAFRANK achieves significant improvements (relative gains of 18-20% MRR with respect to the best baseline) over both feature-based models and prior GNNs in all ranking metrics on friend candidate ranking in both Region 1 and Region 2.

Friend Candidate Retrieval and Ranking (RQ₁)

We compare friend recommendation performance (based on add requests) of various approaches on retrieval and ranking in Tables 7.2 and 7.3 respectively. Interestingly, we find that SAGE based models variants often outperform popular GNN models GCN and GAT. A possible explanation is the impact of feature space heterogeneity in social networks and stochastic neighbor sampling; this results in noisy user representations for GNN models (GCN, GAT) that recursively aggregate neighbor features without emphasizing self-connections. Preserving knowledge of the original user features by explicitly concatenating the self-embedding in each layer results in noticeable gains (SAGE variants).

GRAFRANK significantly outperforms state-of-the-art approaches with over 20-30% relative MRR gains. In contrast to singular aggregation over the entire feature space by prior GNNs, GRAFRANK handles variance in homophily across different modalities through

Dataset	Add		Reciprocate		Communicate	
Metric	HR@50	MRR	HR@50	MRR	HR@50	MRR
LogReg	0.5262	0.1751	0.5582	0.2029	0.5495	0.1811
MLP	0.5726	0.1903	0.6006	0.2165	0.6001	0.1979
XGBoost	0.5957	0.2071	0.6286	0.2322	0.6407	0.2274
GCN	0.5154	0.1034	0.5329	0.1113	0.5273	0.1038
GAT	0.5352	0.0960	0.5596	0.1045	0.5654	0.0971
SAGE + Max	0.6409	0.1797	0.6653	0.2043	0.6670	0.1834
SAGE + Mean	0.6757	0.2333	0.6984	0.2609	0.7056	0.2446
GraFrank	0.7533	0.3035	0.7756	0.3367	0.7942	0.3152
Percent Gains	11.48 %	30.09%	11.05%	29.05%	12.56%	28.86 %

Table 7.4: Comparison on add, reciprocate and communicate friendship retrieval tasks (reported on Region 1). GRAFRANK has consistent gains across all tasks.

modality-specific communication-aware neighbor aggregation. Further, the final user representations are learnt by a correlation-aware attention layer to capture discriminative facets of each modality.

Alternative Friendship Quality Indicators (RQ₂)

In addition to evaluating friend suggestion based on the generated friend *addition* requests, we consider other metrics to quantify friendship quality, *e.g.*, social platforms often want to incentivize friendships that result in greater downstream engagement. We therefore define friendship *reciprocation* and future bi-directional *communication* as two alternative measures of friendship quality. We present comparisons to evaluate *reciprocated* and *communicated* friend recommendation performance on the retrieval task in Table 7.4.

We observe consistently high gains for GRAFRANK on the reciprocated and communicated friend retrieval tasks; this also demonstrates the generality of our pairwise friend ranking objective (Equation 7.13) in learning user representations that promote downstream engagement. Designing multi-criteria ranking objectives to balance different quality measures is worth exploring in the future.

7.5.4 Ablation Study (RQ₃)

In this section, we present two model ablation studies to analyze the *architectural modeling* choices and *training strategies* in GRAFRANK respectively. We briefly describe the two studies and our empirical findings below.

Dataset	Retrieval		Ranking	
Metric	HR@50	MRR	HR@10	MRR
(a) GraFRank_U	0.6968	0.2346	0.4255	0.2164
(b) GraFRank_{UL}	0.7069	0.2423	0.4450	0.2301
(c) GraFRank_{UM}	0.7239	0.2823	0.4887	0.2480
GraFRank	0.7533	0.3035	0.5051	0.2669

Table 7.5: Model architecture ablation study of GRAFRANK. Removing (c) link communication features, (b) modality-specific aggregation, or (a) both, hurt model performance.

Model Architecture

We design three variants of GRAFRANK to study the utilities of *communication-aware* and *modality-specific aggregation* towards friend suggestion.

- **GraFRank_{UM} (User-Modality):** We analyze the contribution of link features by parameterizing the k -th modality aggregator with *just* the k -th modality user features (Equation 7.3). Note that link features are excluded during neighbor aggregation.
- **GraFRank_{UL} (User-Link):** To study the effectiveness of learning *modality-specific* aggregators, we define a *single modality-agnostic* aggregator over user feature vectors obtained by concatenation across the K modalities and link features.
- **GraFRank_U (User):** We remove link features from the aggregator in **GraFRank_{UL}** to further test the standalone benefits of link features in parameterizing a single neighbor aggregator.

The model performance of all architectural variations are reported in Table 7.5. **GraFRank_{UL}** performs much worse than **GraFRANK**, thus highlighting the benefits of learning multiple modality-specific aggregators to account for varying extents of modality homophily.

Communication-aware neighbor aggregation is effective at identifying actively engaged friends during neighbor aggregation; this is evidenced by the performance gains of **GraFRANK** over **GraFRank_{UM}** (modality-aware user feature aggregation). We find noticeable gains from parameterizing the neighbor aggregator with link features even in the absence of modality-specific aggregation; the comparison between **GraFRank_{UL}** and **GraFRank_U** (single user feature aggregator without link features) further validates this benefits of utilizing link features during message passing.

Training Strategy

We examine different training strategies to learn GNN models for friend recommendation. We train GRAFRANK with random negatives for candidate retrieval, but adopt two-phase hard negative fine-tuning (with random negative pretraining) for candidate ranking. To validate our choices, we examine three model training settings: (a) random negative training, (b) hard negative training, and (c) fine-tuning (after pretraining on random negatives), for two GNN models: SAGE + Mean, and GRAFRANK, across both candidate retrieval and ranking tasks. Note that training with combination of random and hard negatives, as proposed in [5], is excluded since it consistently performs worse than the above three strategies on both retrieval and ranking tasks.

Dataset	Retrieval		Ranking	
Metric	HR@50	MRR	HR@10	MRR
SAGE + Mean (random)	0.6757	0.2333	0.3943	0.1923
SAGE + Mean (hard)	0.3275	0.0766	0.4330	0.2255
SAGE + Mean (fine-tune)	0.3978	0.0965	0.4561	0.2372
GraFRank (random)	0.7533	0.3035	0.4655	0.2254
GraFRank (hard)	0.4542	0.1461	0.4823	0.2594
GraFRank (fine-tune)	0.5283	0.1871	0.5051	0.2669

Table 7.6: *Training strategy comparison of two GNNs across retrieval and ranking tasks. Random negative training achieves best results for retrieval. Random negative pretraining with hard negative fine-tuning benefits ranking.*

We make three consistent observations from the performance comparison (Table 7.6) across all of the compared GNN models:

- Random negative training achieves best results for retrieval, but performs poorly on ranking; such models lack the *resolution* to discriminate amongst potential candidates for re-ranking.
- Training on hard negatives improves candidate ranking as expected, yet results in poor retrieval performance. Learning fine profile-oriented distinctions among graph-based neighbors, is actually detrimental to the coarse-grained task of retrieval.
- Random negative pretraining yields good parameter initialization points that are more conducive for effective fine-tuning on hard negatives. Fine-tuning improves results for all GNNs over direct hard negative training on ranking, but is ineffective for retrieval.

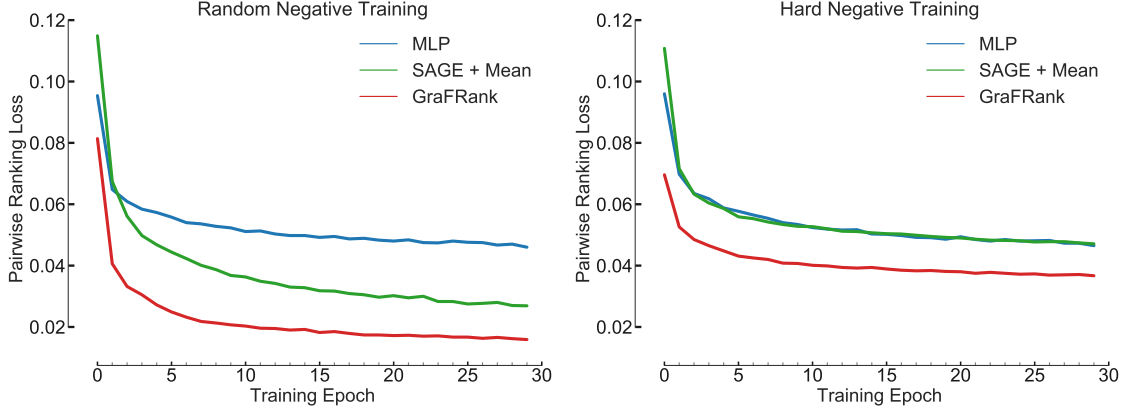


Figure 7.6: GRAFRANK converges faster to better optimization minima in random and hard negative settings, which translates to notable gains on both retrieval and ranking tasks.

7.5.5 Training and Sensitivity Analysis (RQ₄)

In this section, we quantitatively analyze model *convergence* and model *sensitivity* to sampled *neighborhood sizes* in GNN models.

Model Training Analysis

We investigate the relative abilities of different models to optimize the pairwise friend ranking objective (Equation 7.13). We compare the convergence rates of baselines MLP, SAGE + Mean, and GRAFRANK under both random and hard negative training settings, by examining the average training loss per epoch in Figures 7.6 (a) and (b) respectively.

As expected, all models converge to a lower training loss against random negatives (Figure 7.6 (a)) when compared to hard negatives (Figure 7.6 (b)). Interestingly, SAGE + Mean shows similar training convergence as MLP in Figure 7.6 (b), but achieves better test results; this indicates better generalization for GNNs over feature-based models. Compared to baselines, GRAFRANK converges to a better optimization minimum, under both random and hard negative settings, that also generalizes to better test results (Tables 7.2 and 7.3).

Runtime and Sensitivity Analysis

A key trade-off in training scalable GNN models lies in choosing the size of sampled neighborhoods T in each message-passing layer. In our experiments, we train two-layer GNN models for friend ranking. Figure 7.7 shows the runtime and performance of SAGE + Mean and GRAFRANK for different sizes of sampled neighborhoods T from 5 to 20.

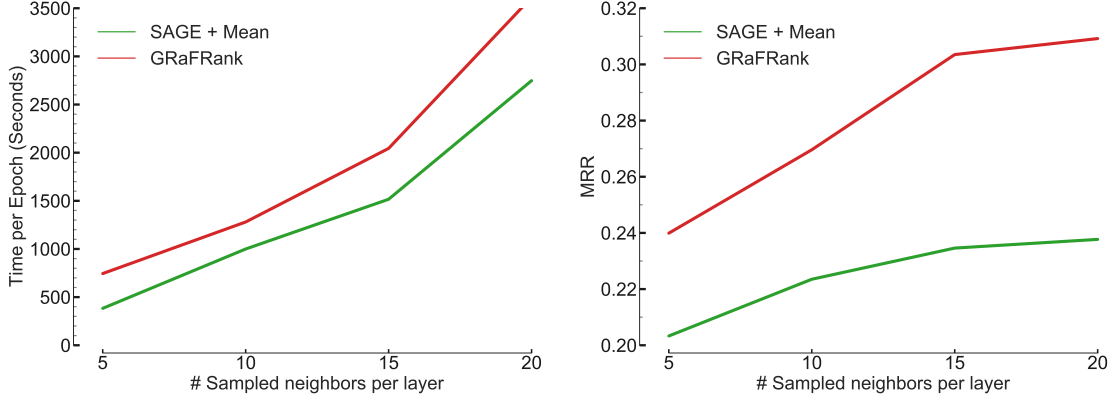


Figure 7.7: We observe diminishing returns in MRR after neighborhood size $T = 15$; GRaFRANK has significant gains over SAGE + Mean, with marginally higher training times.

Model training time generally increases linearly with T , but a greater slope after $T = 15$. We also observe diminishing returns in model performance (MRR) with increase in the size of sampled neighborhood T after $T = 15$. Thus, we select a two-layer GNN model with layer-wise neighborhood size of 15, to provide an effective trade-off between computational cost and performance.

Compared to SAGE + Mean, GRaFRANK has marginally higher training times, yet achieves significant performance gains (20% MRR), justifying the added cost of modality-specific aggregation.

7.5.6 User Cohort Analysis (RQ₅)

In this section, we present multiple *qualitative* analyses to examine model performance across user segments with varied node *degree* and friending *activity* levels, and compare *t-SNE visualizations* of user representations learned by different neural models.

Impact of degree and activity

We examine friend recommendation performance, across users with different node degrees and friending activities. Specifically, for each test user, *degree* is the number of friends, and *activity* is the number of friend requests sent/received in the past 30 days. We divide the test users into groups, independently based on their degree and activity levels. We compare GRaFRANK with feature-based models MLP, XGBoost and the best GNN baseline SAGE + Mean. Figures 7.8(a) and (b) depict friend candidate retrieval performance HR@50 across

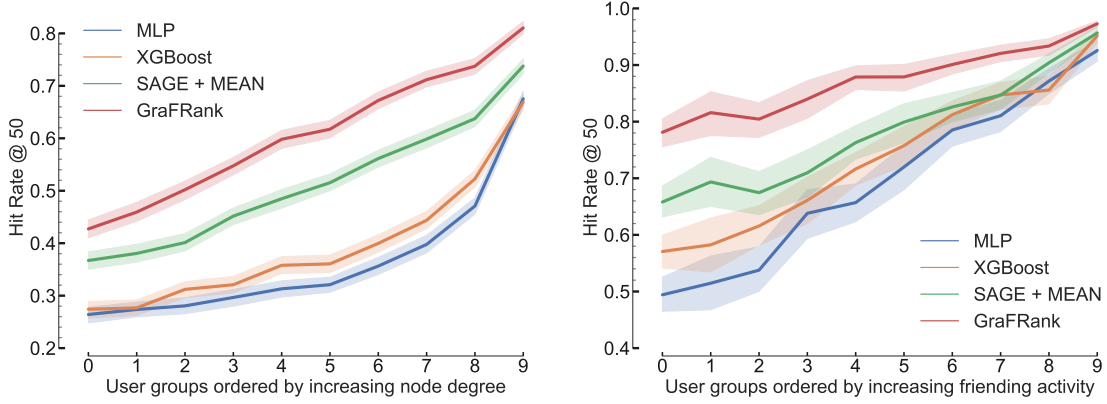


Figure 7.8: GRAFRANK has significant gains across all user segments, with notably larger gains for low-to-medium degree users (a), and less-active users (b).

user segments with different degrees and activities respectively.

From Figure 7.8(a), overall model performance generally increases with node degree, due to the availability of more structural information. GRAFRANK has significant improvements across all user segments, with notably higher gains for *low-to-medium* degree users (relative gains of 20%). GRAFRANK prioritizes active friendships by communication-aware message-passing, which compensates for the lack of sufficient local connectivities in the ego-network.

The performance variation across users with different *activity* levels in Figure 7.8(b), exhibits more distinctive trends with clear gains for GNN models over feature-based MLP and XGBoost for less-active users. Significantly, GRAFRANK has much stronger gains over SAGE + Mean, in *less-active user segments*, owing to its *multi-faceted modeling* of heterogeneous in-platform user actions. GRAFRANK effectively overcomes sparsity concerns for less-active users, through modality-specific neighbor aggregation over multi-modal user features to learn expressive user representations for friend ranking.

Visualization

To analyze the *versatility* of learned user embeddings, we present a qualitative visualization to compare different models on their expressivity to capture geographical user proximity. We randomly select users from three different cities within Region 1 and use t-SNE [280] to transform their learned embeddings into two-dimensional vectors. Figure 7.9 compares the visualization results from different neural models. Evidently, the visualization learned by MLP does not capture geographical proximity, while the GNN models are capable of grouping users located within the same city. Compared to SAGE + Mean, GRAFRANK forms even more well-segmented groups with minimal inter-cluster overlap.

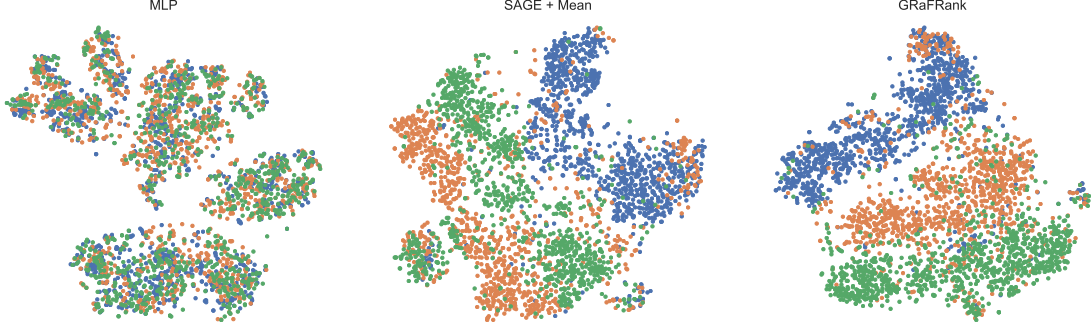


Figure 7.9: *Visualization of two-dimensional t-SNE transformed user representations from feature-based MLP, and GNN models: SAGE + Mean, and GRAFRANK. Users with the same color belong to the same city. Compared to MLP and SAGE + Mean, the friendship relationships learnt by GRAFRANK result in well-separated user clusters capturing geographical proximity.*

7.6 CONCLUSION

This chapter investigates graph neural network usage and design for friend suggestion in large-scale social platforms. We formulate friend suggestion as a multi-faceted friend ranking with multi-modal user features and link communication features. Motivated by our insights from an empirical analysis on user feature modalities, we design a neural architecture GRAFRANK that handles heterogeneity in modality homophily via modality-specific neighbor aggregators, and learns non-linear modality correlations through cross-modality attention. Our experiments on two multi-million social network datasets from Snapchat reveal significant performance gains in friend candidate retrieval (30% MRR gains) and ranking (20% MRR gains). Further, our qualitative analysis indicates stronger gains for a crucial population of less-active and low-degree users.

In this chapter, we concluded our exploration of inductive user modeling scenarios with an industrial application of generating friend suggestions in social platforms. In summary, this dissertation covers inductive user behavior modeling tasks involving new users, user-generated content, and ephemeral groups, across diverse online interaction platforms.

CHAPTER 8: CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, we first briefly summarize the key research contributions of this dissertation, and then discuss immediate extensions and broader research avenues for future work.

8.1 RESEARCH CONTRIBUTIONS

We designed neural frameworks to model user behavior in online platforms (*e.g.*, social networking and e-commerce platforms), involving activities and interactions with other users and functionalities. We examined behavior modeling applications across diverse scenarios, involving graph-structured interactions in social networks, bipartite user-item interactions in e-commerce platforms, and multipartite user-group-item interactions in group activities.

8.1.1 Technical Challenges

Designing neural user modeling frameworks poses several technical challenges in handling the massive scale of behavioral data and the diversity of interaction types involving over millions of entities. Despite the potential to learn neural models that harness massive interaction logs, a central theme of this dissertation is addressing *data sparsity* challenges that manifest in different ways across applications, which are briefly summarized below:

Ground-truth Label Sparsity: In large-scale online platforms, we typically have access to very limited ground-truth labels for entity attributes, *e.g.*, age, gender of users in social networks or aspect ratings of products in e-commerce platforms.

In chapters 3 and 6, to compensate for the lack of sufficient training label information, we designed self-supervised learning strategies that derive auxiliary supervision signals from the intrinsic structure (*e.g.*, higher-order connectivity patterns or group membership structures) of the underlying data.

Entity-level Interaction Sparsity: Heavy-tailed distributions in user interests and interaction patterns are commonly observed in user behavioral data [7] across several online platforms. Despite the availability of behavioral data at massive scales, we have very limited historical records at the granularity of individual entities due to the highly skewed interaction distribution. Thus, learning meaningful trends or insights for a vast majority of entities is challenging due to *entity-level interaction sparsity*.

We explore deep generative modeling (Chapter 5), self-supervised learning (Chapter 6) and meta-learning (Chapter 4) paradigms to enable accurate personalized recommendation and inference for data-poor tail entities (such as long-tailed users, items and ephemeral groups) with severe interaction sparsity.

Feature Diversity and Skew: In inductive learning applications, entities are often represented using a combination of diverse features, *e.g.*, user features in social networking platforms may include static profile attributes, dynamic communication and engagement activities, while item features in e-commerce platforms may include textual descriptions, product reviews, and other attributes (such as brand or price). Here, the key modeling challenges are feature heterogeneity and skew: the features often belong to different modalities and exhibit non-trivial correlations; we also observe skewed occurrences of features across different entities.

To handle attribute diversity in local graph neighborhoods, we introduced graph neural network training objectives that learn local structure and attribute co-variation in chapter 3. We also designed multi-faceted graph neural architectures that can effectively learn from multiple correlated feature modalities in chapter 7.

In this dissertation, we primarily developed architecture-agnostic frameworks for user behavior modeling, with generalizable learning strategies that are targeted towards addressing the aforementioned sparsity challenges. Our learning frameworks enable personalized inference at scale for broad application scenarios across diverse platforms. Furthermore, we also present an example of a specific deep learning architecture design that effectively handles sparsity concerns in a large-scale industrial application.

Our problem settings span transductive and inductive learning scenarios, where transductive learning models behavior of entities seen during training and inductive learning targets unseen entities that are only observed during inference. Thus, our key research contributions are also organized below into *transductive* and *inductive* user behavior modeling applications.

8.1.2 Transductive User Behavior Modeling

We examined two transductive learning settings: inference and recommendation in *graph-structured* interactions (Chapter 3) and *bipartite user-item* interactions (Chapter 4).

First, in Chapter 3, we formulate user profiling in online platforms as semi-supervised (transductive) learning over graphs connecting different entities. To address ground-truth *label sparsity* in node attribute inference, we utilize higher-order connectivity structures (network motifs) to effectively regularize arbitrary graph neural networks through self-supervised

learning objectives. Our framework learns attributed structural roles for nodes to identify structurally similar nodes with co-varying local attributes independent of graph proximity, thus effectively handling *feature sparsity and diversity* in local neighborhoods.

Thereafter, in Chapter 4, we focus on learning to recommend *few-shot* items with neural collaborative filtering models for personalized recommendation. To eliminate the distribution disparity between head items (with *abundant* interactions) and tail items (with *sparse* interactions), our novel meta-learning framework learns-to-compose robust prototype representations for few-shot items via knowledge transfer from arbitrary base recommenders.

8.1.3 Inductive User Behavior Modeling

We examined three inductive learning settings with a focus on prediction and recommendation tasks for entities that are only observed during inference-time. Specifically, we modeled the spread or diffusion of *user-generated content* in social networks (Chapter 5), and enabled item recommendations for *ephemeral (unseen) groups* (Chapter 6). Finally, we designed inductive ranking models for the application of *friend suggestion* in large-scale social platforms (Chapter 7).

First, in Chapter 5, we model the spread (or diffusion) of user-generated content in social networks. Here, we encounter *interaction sparsity* challenges since a vast population of users seldom post content (sparse diffusion actions). To address interaction sparsity, we introduced a deep generative modeling framework that models users as probability distributions in the latent space with variational priors parameterized by graph neural networks. Our social regularization framework incorporates the co-variance of temporal context (recent posting activities) with structural graph connectivity for diffusion prediction.

Then, in Chapter 6, we focus on personalized item recommendations for ephemeral groups with limited or no historical interactions together. To address *group interaction sparsity*, we introduced self-supervised learning objectives that exploit the preference co-variance among group members for group recommender training. To enable inductive generalization to ephemeral groups, our framework relies on mutual information estimation and maximization over group membership structures, to effectively regularize base group recommenders.

Finally, in Chapter 7, we proposed multi-modal inference with graph neural networks for the industrially ubiquitous application of generating friend suggestion in social networking platforms. To address *interaction sparsity* challenges for *less-active* and *low-degree users*, we presented an inductive graph neural network model that captures knowledge from multiple correlated user feature modalities and user-user interactions over a time-evolving friendship graph to achieve multi-faceted friend ranking.

8.1.4 Summary of Learning Paradigms

In this dissertation, we have explored several architectural modeling choices and training strategies to address the sparsity-oriented technical challenges across the different chapters. Here, we provide a brief summary of the central learning paradigms:

Graph Neural Networks (design and training)

Graph Neural Networks (GNNs) are a powerful paradigm to learn node representations combining graph topology and node/edge features for both transductive and inductive learning applications. Their localized message passing framework makes them to extremely effective to directly alleviate entity-level label and interaction sparsity: GNNs can directly consider structural connectivity and interactions with other related entities through local message-passing propagations; GNNs can easily incorporate side information for different entities through node and edge features belonging to diverse modalities.

In this dissertation, we have advanced graph neural network design, training, and applications in several directions, as described below:

- Model training strategies to capture node structural roles via higher-order connectivity structures (Chapter 3).
- Social regularization of sequential models via variational graph autoencoders for diffusion prediction. (Chapter 5).
- Model design to learn from multifaceted interactions in large-scale social networking platforms (Chapter 7).

Deep Generative Modeling

Deep Generative Models are powerful neural networks that aim to approximate any kind of data distribution [281]. We leverage their expressive power to model data-driven priors for entities (*e.g.*, users, items) that are represented as probability distributions in the latent space. By parameterizing latent space priors through deep generative models that can capture different modeling hypotheses, we effectively address entity-level sparsity concerns.

Variational Autoencoders (VAEs) [135] and Generative Adversarial Networks (GANs) [136] are two of the most commonly used frameworks for deep generative modeling. VAEs maximize a variational lower bound of the data log-likelihood, while GANs aim to achieve an equilibrium between a generator and discriminator through alternating optimization.

In chapter 5, we design a variational autoencoder framework for diffusion modeling, where a graph VAE regularizes the user latent space to capture structural graph connectivity information. We have also explored adversarial learning to model item space priors based on co-occurrence statistics to address item-level sparsity [9] (not included in this dissertation).

Self-supervised Learning

The paradigm of self-supervised learning [282] alleviates label and interaction sparsity challenges by setting up new learning objectives based on additional supervision signals extracted from the intrinsic structure of the data. For instance, auxiliary supervision signals for images may be created by rotating, cropping, or colorizing images, each associated with appropriate supervised training objectives. Despite promising recent advances owing to self-supervised learning in computer vision [144] and natural language [147] applications, this paradigm is relatively unexplored in the graph mining and recommendation domains.

In chapter 3, we design self-supervised learning objectives to regularize graph neural networks for transductive node classification by learning attribute correlations in higher-order connectivity structures (network motifs). In chapter 6, we presented a self-supervised learning framework that relies on mutual information estimation and maximization over group memberships, to overcome group interaction sparsity for ephemeral group recommendation.

Meta-Learning for Few-shot Inference

The meta-learning (learning to learn) paradigm intends to rapidly learn new concepts or tasks given a limited number of few training examples. There are three common approaches: metric-based [101], gradient-based [152] and optimization-based models for meta-learning. Notably, meta-learning has achieved considerable success in few-shot supervised learning such as image classification applications in computer vision. Yet, designing meta-learning frameworks for user behavior modeling in online platforms with severe interaction sparsity, poses unique scaling challenges.

In chapter 4, we formulate long-tail recommendations in online platforms as a few-shot learning problem of learning-to-recommend entities (users/items) with very few interactions. We introduced a novel metric-based few-shot item recommendation framework that outperforms state-of-art neural recommendation approaches on overall item recommendation (by 5% Recall@50) while achieving significant performance gains (of 60-80% Recall@50) for tail items (with less than 20 interactions).

8.2 FUTURE WORK

There are multiple avenues of future work that can improve upon our proposed frameworks for sparsity-aware user behavior modeling in online platforms. We first discuss immediate extensions that can directly extend the learning paradigms and applications discussed in this dissertation. Later, we conclude with a discussion on broader research directions (beyond sparsity-oriented challenges) that are critical to online user behavior models.

8.2.1 Extensions to our Frameworks

We first discuss a few extensions to our approaches introduced in the preceding chapters.

Self-supervised Pre-training and Learning over Graphs

In chapter 3, we designed self-supervised learning objectives over graphs to learn attribute correlations in higher-order connectivity structures (network motifs). We can further generalize this self-supervised learning framework beyond motifs to preserve other structural properties and social hypotheses, including pairwise distances, global community structures, attribute homophily, etc. An important direction is leveraging self-supervised learning objectives to pre-train graph neural networks over pretext tasks that explore a variety of different learning perspectives. With the rapid proliferation of e-commerce and social platforms, pre-training graph neural networks over large-scale rich, attributed, and heterogeneous graphs will become critical, similar to BERT [147] models in natural language applications.

Modeling Group Formation and Evolution

We have briefly examined group interaction modeling in Chapter 6 through personalized item recommendations for ephemeral groups with limited or no historical interactions together. In reality, group interactions occur in a variety of different contexts in modern social platforms (*e.g.*, Facebook, Snapchat), where new groups frequently form and existing groups evolve over time with respect to both memberships and interests. Compared to prior dynamic graph evolution literature that capture node-level structural and behavior evolution, group interactions often involve complex dynamics among different members. Thus, an important future research direction is modeling group formation and evolution in social networking platforms, by formulating appropriate temporal link prediction and user recommendation problems.

Deep Generative Modeling for Cold-start Recommendation

In chapter 5, we explored deep generative models [281] for robust behavior modeling in social networks with sparsity in content-posting activities of users. Modeling entity representations as probability distributions in the latent space, has advantages: it allows for flexible distribution-aware and data-driven prior regularization of the latent space with potential gains for long-tail and cold-start entities; flexible probabilistic models better capture uncertainty in the latent space, including data-point variances. Most of the prior neural collaborative filtering approaches are discriminative recommendation models that are effective for previously observed entities. To enable cold-start recommendations, one effective approach is to share probabilistic priors within entities of the same category (*e.g.*, restaurants of the same cuisine), which can be learned from the auxiliary side information (*e.g.*, content, knowledge graphs, etc.) associated with entities. We can build on advances in variational autoencoders (VAEs) [135] and generative adversarial networks (GANs) [136] to design effective deep generative recommendation models in variety of different settings that entail cold-start entities.

Meta-Learning for Cross-Domain Recommendation

In chapter 4, we proposed a metric-based meta-learning framework for few-shot recommendation over user-item interactions in a single domain. Our meta-learning approach enables knowledge transfer by relying on the assumption of similar training and inference interaction distributions, which is expected to be satisfied across items within a single domain. Cross-domain recommendation settings are commonly observed in large-scale e-commerce platforms where domains may describe products belonging to different categories or interactions in different devices (voice versus web shopping) over a shared user space. In such a scenario, we can explore meta-learning frameworks supplemented with appropriate domain adaptation [283] strategies to account for distributional shifts across domains. Apart from metric-based approaches that enable knowledge via shared latent embeddings, we may also explore gradient-based and optimization-based meta-learning models, which can potentially provide more flexibility in handling cross-domain distribution disparities.

8.2.2 Broader Research Challenges beyond Sparsity

We finally discuss a few research directions that go beyond the sparsity-oriented challenges discussed in the other chapters of this dissertation.

Detecting Abuse and Malicious Activity

In this dissertation, we build neural user modeling frameworks that are robust to a variety of data sparsity challenges that manifest in behavioral data. An underlying assumption is that the data utilized for model training and inference is clean, reliable, and of high-quality, *i.e.*, the behavioral data accurately represents the information about the interacting entities. In online platforms, we often observe malicious activities that deny, disrupt, degrade and deceive other participants with a wide range of intents, *e.g.*, fake accounts (which pose potential threats to the safety and security of online communities), and threatening or disparaging content (such as hate-speech, misinformation, bullying and harassment).

One approach to handle malicious activities is to pro-actively detect instances of abusive behavior and remove the corresponding malicious accounts in the data processing stage itself before input to our neural behavior models [284]. Abusive behavior detection can be formulated as a binary classification problem [285], which is severely imbalanced due to label sparsity associated with labeling malicious activities. Our few-shot learning approaches to handle long-tailed distributions can potentially be generalized to learn effective abuse detection models. It is also worthwhile to leverage the power of crowd-sourcing [286] for annotating posts with abuse-related labels in social platforms.

Model Robustness to Adversarial Attacks

Despite the efficacy of abuse detection strategies, malicious interaction data may invariably become a part of training sets that are used for model learning. A few recent studies have demonstrated the vulnerability of deep learning models to adversarial examples [287], which are subtle (but non-random) perturbations that induce erroneous outputs from the model (*e.g.*, misclassification), *e.g.*, adversarial examples in e-commerce applications may include fake user profiles that aim to promote (or demote) specific item(s) in the top- K recommendation list of users, or to recommend irrelevant items (to create a mistrust on a system) [288, 289].

Thus, it is important to design models that are robust to adversarial attacks, including both poisoning (training-time) and evasion (inference-time) attacks. Crafting adversarial examples (*e.g.*, fake user profiles) for graph-structured data or recommender systems is significantly more challenging due to the mostly discrete nature of online interactions. We can explore new learning frameworks for generating adversarial examples, grounded on model compression strategies such as knowledge distillation [208] and deep generative modeling paradigms of GANs [136] and VAEs [135]. Since the details of most deployed machine

learning models are not publicly available, it is also critical to develop black-box adversarial attack strategies that do not have access to the architectural details or model parameters. The final step is the design of specialized training and optimization strategies (such as adversarial training or denoising) to enhance model robustness against adversarial examples.

Privacy-preserving User Modeling

In this dissertation, we have assumed complete access to online interaction data across all users for model training. However, this data often includes personal information about users (*e.g.*, age, gender, location, who likes what etc.), which may reveal sensitive information with potential for misuse [290] and can further result in de-anonymization [291] with the use of sophisticated techniques. With recent government regulations and laws on privacy protection (*e.g.*, General Data Protection Regulation (GDPR) in EU) and increased privacy awareness, it is important to design user modeling techniques that ensure user privacy while generating personalized recommendations.

In contrast to the traditional paradigm of collecting, storing, and processing sensitive user data on a centralized external backend server, we can explore distributed federated learning paradigms [292] which can ensure that sensitive user data never leaves their accounts (or devices). In this paradigm, a master model is distributed to the end clients (users), who in turn can use their locally stored private data and personalized model for both inference and updates. The updates from different users are aggregated on the server to update the master model, which can then be redistributed to the users. Though federated learning can enable a higher level of privacy than conventional machine learning, it is also critical to derive theoretical estimates of information leakage in such privacy-preserving solutions.

Bias and Fairness Issues in Behavior Models

In recent times, several studies have shown the potential for deployed machine learning systems to amplify social inequities and unfairness; a few examples include an automated hiring system that had a higher likelihood of recommending from certain racial, gender, or age groups [293], and an insurance company that involuntarily discriminated against elderly patients when using machine learning to workout insurance premiums. In this context, two central concepts of *bias* and *fairness* are important. In general, a machine learning system exhibits bias if it systematically and unfairly discriminates against certain individuals or groups of individuals (more generally entities) in favor of others. The goal of fairness is to design machine learning algorithms that make fair predictions across various entities or

groups of entities, *e.g.*, groups may be defined based on different protected attributes (*e.g.*, race, gender) of users [294].

In several chapters of this dissertation, we have examined *popularity bias* issues arising due to interaction skew in social networks and recommender systems (popular entities interact frequently, while the majority of other entities do not get much attention). Beyond popularity bias, it is important to quantify and measure other forms of societal biases in online social media and recommendation applications, to ensure appropriate exposure of the protected attributes such as user demographic properties (age, gender, race, ethnicity, location, etc.) and diversity (item category or genres, supplier, etc.). The overall objective is the design fairness-aware recommendation and inference algorithms for mitigating different biases without severely impacting key business metrics; we note that fairness-aware approaches can be applied at several parts of the pipeline: data collection and generation (pre-processing), at the point of modeling (in-processing), or after modeling (post-processing).

REFERENCES

- [1] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.
- [2] R. Borges and K. Stefanidis, “On measuring popularity bias in collaborative filtering data,” in *EDBT/ICDT Workshops*, 2020.
- [3] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in *RecSys*, 2016, pp. 191–198.
- [4] M. Grbovic and H. Cheng, “Real-time personalization using embeddings for search ranking at airbnb,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3219819.3219885> pp. 311–320.
- [5] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3219819.3219890> pp. 974–983.
- [6] A. Pal, C. Eksombatchai, Y. Zhou, B. Zhao, C. Rosenberg, and J. Leskovec, “Pinner-sage: Multi-modal user embedding framework for recommendations at pinterest,” in *KDD*, 2020, pp. 2311–2320.
- [7] A.-L. Barabasi, “The origins of bursts and heavy tails in human dynamics,” *Nature*, vol. 435, pp. 207–11, 06 2005. [Online]. Available: <https://dx.doi.org/10.1038/nature03459>
- [8] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The Adaptive Web, Methods and Strategies of Web Personalization*, 2007. [Online]. Available: https://doi.org/10.1007/978-3-540-72079-9_9 pp. 291–324.
- [9] A. Krishnan, A. Sharma, A. Sankar, and H. Sundaram, “An adversarial approach to improve long-tail performance in neural collaborative filtering,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3269206.3269264> pp. 1491–1494.
- [10] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network motifs: simple building blocks of complex networks,” *Science*, vol. 298, no. 5594, pp. 824–827, 2002. [Online]. Available: <https://dx.doi.org/10.1126/science.298.5594.824>

- [11] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a feather: Homophily in social networks,” *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001. [Online]. Available: <https://www.annualreviews.org/doi/10.1146/annurev.soc.27.1.415>
- [12] S. Aral, L. Muchnik, and A. Sundararajan, “Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21 544–21 549, 2009.
- [13] C. R. Shalizi and A. C. Thomas, “Homophily and contagion are generically confounded in observational social network studies,” *Sociological methods & research*, vol. 40, no. 2, pp. 211–239, 2011. [Online]. Available: <http://arxiv.org/abs/1004.4704>
- [14] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, 2007. [Online]. Available: <https://doi.org/10.1109/TPAMI.2007.250598>
- [15] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*. MIT Press, 2001. [Online]. Available: <https://proceedings.neurips.cc/paper/2001/hash/f106b7f99d2cb30c3db1c3cc0fde9ccb-Abstract.html> pp. 585–591.
- [16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2013. [Online]. Available: <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html> pp. 3111–3119.
- [17] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014. [Online]. Available: <https://doi.org/10.1145/2623330.2623732> pp. 701–710.
- [18] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2939672.2939754> pp. 855–864.
- [19] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, 2015. [Online]. Available: <https://doi.org/10.1145/2736277.2741093> pp. 1067–1077.

- [20] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2939672.2939751> pp. 1105–1114.
- [21] Z. Zhang, P. Cui, X. Wang, J. Pei, X. Yao, and W. Zhu, “Arbitrary-order proximity preserved network embedding,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3219819.3219969> pp. 2778–2786.
- [22] R. A. Rossi, D. Jin, S. Kim, N. K. Ahmed, D. Koutra, and J. B. Lee, “From community to role-based graph embeddings,” *CoRR*, vol. abs/1908.08572, 2019. [Online]. Available: <http://arxiv.org/abs/1908.08572>
- [23] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li, “Rolx: structural role extraction & mining in large graphs,” in *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, 2012. [Online]. Available: <https://doi.org/10.1145/2339530.2339723> pp. 1231–1239.
- [24] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, “*struc2vec*: Learning node representations from structural identity,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3097983.3098061> pp. 385–394.
- [25] R. A. Rossi, N. K. Ahmed, E. Koh, S. Kim, A. Rao, and Y. Abbasi-Yadkori, “A structural graph representation learning framework,” in *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3336191.3371843> pp. 483–491.
- [26] D. Wang, P. Cui, and W. Zhu, “Structural deep network embedding,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2939672.2939753> pp. 1225–1234.
- [27] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *NIPS Workshop on Bayesian Deep Learning*, 2016. [Online]. Available: <http://arxiv.org/abs/1611.07308>
- [28] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017. [Online]. Available: <http://sites.computer.org/debull/A17sept/p52.pdf>

- [29] H. Gao and H. Huang, “Deep attributed network embedding,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, 2018*. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/467> pp. 3364–3370.
- [30] L. Liao, X. He, H. Zhang, and T. Chua, “Attributed social network embedding,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2257–2270, 2018. [Online]. Available: <https://doi.org/10.1109/TKDE.2018.2819980>
- [31] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, “ANRL: attributed network representation learning via deep neural networks,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, 2018*. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/438> pp. 3155–3161.
- [32] X. Zhou and M. Belkin, “Semi-supervised learning,” in *Academic Press Library in Signal Processing*. Elsevier, 2014, vol. 1, pp. 1239–1269.
- [33] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” in *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, 2003. [Online]. Available: <https://proceedings.neurips.cc/paper/2003/hash/87682805257e619d49b8e0dfdc14affa-Abstract.html> pp. 321–328.
- [34] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, “Collective classification in network data,” *AI Mag.*, vol. 29, no. 3, pp. 93–106, 2008. [Online]. Available: <https://doi.org/10.1609/aimag.v29i3.2157>
- [35] Z. Yang, W. W. Cohen, and R. Salakhutdinov, “Revisiting semi-supervised learning with graph embeddings,” in *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 2016. [Online]. Available: <http://proceedings.mlr.press/v48/yanga16.html> pp. 40–48.
- [36] X. Huang, J. Li, and X. Hu, “Label informed attributed network embedding,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3018661.3018667> pp. 731–739.
- [37] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: <http://arxiv.org/abs/1312.6203>
- [38] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016*, pp. 3837–3845.

- [39] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: <https://openreview.net/forum?id=SJU4ayYgl>
- [40] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. [Online]. Available: <https://openreview.net/forum?id=rJXMpikCZ>
- [41] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7e9bea9-Abstract.html> pp. 1024–1034.
- [42] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, “Representation learning on graphs with jumping knowledge networks,” in *ICML*, 2018, pp. 5453–5462.
- [43] J. You, R. Ying, and J. Leskovec, “Position-aware graph neural networks,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019. [Online]. Available: <http://proceedings.mlr.press/v97/you19b.html> pp. 7134–7143.
- [44] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” *CoRR*, vol. abs/1809.10341, 2018. [Online]. Available: <http://arxiv.org/abs/1809.10341>
- [45] J. Wu, J. He, and J. Xu, “Demo-net: Degree-specific graph neural networks for node and graph classification,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3330950> pp. 406–415.
- [46] A. Sankar, X. Zhang, and K. C. Chang, “Motif-based convolutional neural network on graphs,” *CoRR*, vol. abs/1711.05697, 2017. [Online]. Available: <http://arxiv.org/abs/1711.05697>
- [47] J. B. Lee, R. A. Rossi, X. Kong, S. Kim, E. Koh, and A. Rao, “Graph convolutional networks with motif-based attention,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3357384.3357880> pp. 499–508.
- [48] Z. Huang and N. Mamoulis, “Heterogeneous information network embedding for meta path based proximity,” *CoRR*, vol. abs/1701.05291, 2017. [Online]. Available: <http://arxiv.org/abs/1701.05291>

- [49] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “Metagraph2vec: Complex semantic path augmented heterogeneous network embedding,” in *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II*, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-93037-4_16 pp. 196–208.
- [50] Y. Shi, H. Gui, Q. Zhu, L. M. Kaplan, and J. Han, “Aspem: Embedding learning by aspects in heterogeneous information networks,” in *Proceedings of the 2018 SIAM International Conference on Data Mining, SDM 2018, May 3-5, 2018, San Diego Marriott Mission Valley, San Diego, CA, USA*, 2018. [Online]. Available: <https://doi.org/10.1137/1.9781611975321.16> pp. 144–152.
- [51] T. Fu, W. Lee, and Z. Lei, “Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3132847.3132953> pp. 1797–1806.
- [52] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable representation learning for heterogeneous networks,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3097983.3098036> pp. 135–144.
- [53] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3330961> pp. 793–803.
- [54] Z. Hu, Y. Dong, K. Wang, and Y. Sun, “Heterogeneous graph transformer,” in *WWW ’20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3366423.3380027> pp. 2704–2710.
- [55] Y. Zhang, Y. Xiong, X. Kong, S. Li, J. Mi, and Y. Zhu, “Deep collective classification in heterogeneous information networks,” in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3178876.3186106> pp. 399–408.
- [56] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3308558.3313562> pp. 2022–2032.
- [57] X. Fu, J. Zhang, Z. Meng, and I. King, “MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding,” in *WWW ’20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3366423.3380297> pp. 2331–2341.

- [58] A. Sankar, X. Zhang, and K. C. Chang, “Meta-gnn: metagraph neural network for semi-supervised learning in attributed heterogeneous information networks,” in *ASONAM '19: International Conference on Advances in Social Networks Analysis and Mining, Vancouver, British Columbia, Canada, 27-30 August, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3341161.3342859> pp. 137–144.
- [59] G. Kossinets and D. J. Watts, “Empirical analysis of an evolving social network,” *Science*, vol. 311, no. 5757, pp. 88–90, 2006. [Online]. Available: <https://science.sciencemag.org/content/311/5757/88>
- [60] L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
- [61] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Soc. Networks*, vol. 25, no. 3, pp. 211–230, 2003. [Online]. Available: [https://doi.org/10.1016/S0378-8733\(03\)00009-1](https://doi.org/10.1016/S0378-8733(03)00009-1)
- [62] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, “Multi-relational matrix factorization using bayesian personalized ranking for social network data,” in *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*, 2012. [Online]. Available: <https://doi.org/10.1145/2124295.2124317> pp. 173–182.
- [63] L. Zhu, D. Guo, J. Yin, G. V. Steeg, and A. Galstyan, “Scalable temporal latent space inference for link prediction in dynamic social networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2765–2777, 2016.
- [64] L. Zhou, Y. Yang, X. Ren, F. Wu, and Y. Zhuang, “Dynamic network embedding by modeling triadic closure process,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16572> pp. 571–578.
- [65] P. Goyal, N. Kamra, X. He, and Y. Liu, “Dyngem: Deep embedding method for dynamic graphs,” in *IJCAI International Workshop on Representation Learning for Graphs (ReLiG)*, August 2017. [Online]. Available: <http://arxiv.org/abs/1805.11273>
- [66] P. Goyal, S. R. Chhetri, and A. Canedo, “dyngraph2vec: Capturing network dynamics using dynamic graph representation learning,” *Knowl. Based Syst.*, vol. 187, 2020. [Online]. Available: <https://doi.org/10.1016/j.knosys.2019.06.024>
- [67] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, “Evolvegn: Evolving graph convolutional networks for dynamic graphs,” in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 2020. [Online]. Available: <https://aaai.org/ojs/index.php/AAAI/article/view/5984> pp. 5363–5370.

- [68] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, “Dysat: Deep neural representation learning on dynamic graphs via self-attention networks,” in *WSDM ’20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3336191.3371845> pp. 519–527.
- [69] C. Li, J. Ma, X. Guo, and Q. Mei, “Deepcas: An end-to-end predictor of information cascades,” in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3038912.3052643> pp. 577–586.
- [70] D. Kempe, J. M. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, 2003. [Online]. Available: <https://doi.org/10.1145/956750.956769> pp. 137–146.
- [71] M. Granovetter, “Threshold models of collective behavior,” *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978. [Online]. Available: <http://www.jstor.org/stable/2778111>
- [72] N. Barbieri, F. Bonchi, and G. Manco, “Topic-aware social influence propagation models,” in *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, 2012. [Online]. Available: <https://doi.org/10.1109/ICDM.2012.122> pp. 81–90.
- [73] K. Saito, M. Kimura, K. Ohara, and H. Motoda, “Learning continuous-time information diffusion model for social behavioral data analysis,” in *Advances in Machine Learning, First Asian Conference on Machine Learning, ACML 2009, Nanjing, China, November 2-4, 2009. Proceedings*, 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-05224-8_25 pp. 322–337.
- [74] K. Saito, K. Ohara, Y. Yamagishi, M. Kimura, and H. Motoda, “Learning diffusion probability based on node attributes in social networks,” in *Foundations of Intelligent Systems - 19th International Symposium, ISMIS 2011, Warsaw, Poland, June 28-30, 2011. Proceedings*, 2011. [Online]. Available: https://doi.org/10.1007/978-3-642-21916-0_18 pp. 153–162.
- [75] N. Barbieri, F. Bonchi, and G. Manco, “Influence-based network-oblivious community detection,” in *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, 2013. [Online]. Available: <https://doi.org/10.1109/ICDM.2013.164> pp. 955–960.
- [76] Z. Hu, J. Yao, B. Cui, and E. P. Xing, “Community level diffusion extraction,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, 2015. [Online]. Available: <https://doi.org/10.1145/2723372.2723737> pp. 1555–1569.

- [77] S. Bourigault, C. Lagnier, S. Lamprier, L. Denoyer, and P. Gallinari, “Learning social network embeddings for predicting information diffusion,” in *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, 2014. [Online]. Available: <https://doi.org/10.1145/2556195.2556216> pp. 393–402.
- [78] Y. Wang, H. Shen, S. Liu, and X. Cheng, “Learning user-specific latent influence and susceptibility from information cascades,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, 2015*. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9805> pp. 477–484.
- [79] S. Bourigault, S. Lamprier, and P. Gallinari, “Representation learning for information diffusion through social networks: an embedded cascade model,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2835776.2835817> pp. 573–582.
- [80] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, and Y. M. Chee, “Inf2vec: Latent representation model for social influence embedding,” in *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, 2018. [Online]. Available: <https://doi.org/10.1109/ICDE.2018.00089> pp. 941–952.
- [81] J. Wang, V. W. Zheng, Z. Liu, and K. C. Chang, “Topological recurrent neural network for diffusion prediction,” in *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017*, 2017. [Online]. Available: <https://doi.org/10.1109/ICDM.2017.57> pp. 475–484.
- [82] Z. Wang, C. Chen, and W. Li, “A sequential neural information diffusion model with structure attention,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3269206.3269275> pp. 1795–1798.
- [83] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan, “Deepdiffuse: Predicting the ‘who’ and ‘when’ in cascades,” in *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, 2018. [Online]. Available: <https://doi.org/10.1109/ICDM.2018.00134> pp. 1055–1060.
- [84] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, 2001. [Online]. Available: <https://doi.org/10.1145/371920.372071> pp. 285–295.
- [85] Y. Koren, R. M. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009. [Online]. Available: <https://doi.org/10.1109/MC.2009.263>

- [86] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, 2008. [Online]. Available: <https://doi.org/10.1109/ICDM.2008.22> pp. 263–272.
- [87] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: bayesian personalized ranking from implicit feedback,” in *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, 2009. [Online]. Available: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25 pp. 452–461.
- [88] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, “Neural collaborative filtering,” in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3038912.3052569> pp. 173–182.
- [89] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, “Collaborative denoising auto-encoders for top-n recommender systems,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2835776.2835837> pp. 153–162.
- [90] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, “Variational autoencoders for collaborative filtering,” in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3178876.3186150> pp. 689–698.
- [91] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, “Neural graph collaborative filtering,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3331184.3331267> pp. 165–174.
- [92] T. Ebesu, B. Shen, and Y. Fang, “Collaborative memory network for recommendation systems,” in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3209978.3209991> pp. 515–524.
- [93] X. He and T. Chua, “Neural factorization machines for sparse predictive analytics,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3077136.3080777> pp. 355–364.

- [94] Y. Tay, L. A. Tuan, and S. C. Hui, “Latent relational metric learning via memory-based attention for collaborative ranking,” in *Proceedings of the 2018 World Wide Web Conference WWW 2018, Lyon, France, April 23-27, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3178876.3186154> pp. 729–739.
- [95] Y. Park, “The adaptive clustering method for the long tail problem of recommender systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1904–1915, 2013. [Online]. Available: <https://doi.org/10.1109/TKDE.2012.119>
- [96] Y. Wu, X. Liu, M. Xie, M. Ester, and Q. Yang, “CCCF: improving collaborative filtering via scalable user-item co-clustering,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2835776.2835836> pp. 73–82.
- [97] J. Ma, J. Wen, M. Zhong, L. Liu, C. Li, W. Chen, Y. Yang, H. Tu, and X. Li, “Dbrec: Dual-bridging recommendation via discovering latent groups,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3357384.3357892> pp. 1513–1522.
- [98] D. Kim and B. Suh, “Enhancing vaes for collaborative filtering: flexible priors & gating mechanisms,” in *RecSys*, 2019, pp. 403–407.
- [99] D. Chae, J. Kang, S. Kim, and J. Choi, “Rating augmentation with generative adversarial networks towards accurate collaborative filtering,” in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3308558.3313413> pp. 2616–2622.
- [100] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016. [Online]. Available: <https://proceedings.neurips.cc/paper/2016/hash/90e1357833654983612fb05e3ec9148c-Abstract.html> pp. 3630–3638.
- [101] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html> pp. 4077–4087.
- [102] A. Sankar, J. Wang, A. Krishnan, and H. Sundaram, “Protofcf: Prototypical collaborative filtering for few-shot recommendation,” in *Fifteenth ACM Conference on Recommender Systems*, 2021, pp. 166–175.
- [103] X. Li and J. She, “Collaborative variational autoencoder for recommender systems,” in *KDD*, 2017, pp. 305–314.

- [104] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3018661.3018665> pp. 425–434.
- [105] A. Krishnan, M. Das, M. Bendre, H. Yang, and H. Sundaram, “Transfer learning via contextual invariants for one-to-many cross-domain recommendation,” in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401078> pp. 1081–1090.
- [106] A. Krishnan, H. Cheruvu, T. Cheng, and H. Sundaram, “A modular adversarial approach to social recommendation,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3357384.3357898> pp. 1753–1762.
- [107] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, “KGAT: knowledge graph attention network for recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3330989> pp. 950–958.
- [108] M. Volkovs, G. W. Yu, and T. Poutanen, “Dropoutnet: Addressing cold start in recommender systems,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 4957–4966.
- [109] Z. Zhu, S. Sefati, P. Saadatpanah, and J. Caverlee, “Recommendation for new users and new items via randomized training and mixture-of-experts transformation,” in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401178> pp. 1121–1130.
- [110] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, “Melu: Meta-learned user preference estimator for cold-start recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3330859> pp. 1073–1082.
- [111] H. Wang and Y. Zhao, “ML2E: meta-learning embedding ensemble for cold-start recommendation,” *IEEE Access*, vol. 8, pp. 165 757–165 768, 2020. [Online]. Available: <https://doi.org/10.1109/ACCESS.2020.3022796>

- [112] Y. Lu, Y. Fang, and C. Shi, “Meta-learning on heterogeneous information networks for cold-start recommendation,” in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3394486.3403207> pp. 1563–1573.
- [113] M. Dong, F. Yuan, L. Yao, X. Xu, and L. Zhu, “MAMO: memory-augmented meta-optimization for cold-start recommendation,” in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3394486.3403113> pp. 688–697.
- [114] Y. Zhang, D. Z. Cheng, T. Yao, X. Yi, L. Hong, and E. H. Chi, “A model of two tales: Dual transfer learning framework for improved long-tail item recommendation,” in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2021. [Online]. Available: <https://doi.org/10.1145/3442381.3450086> pp. 2220–2231.
- [115] Z. Du, X. Wang, H. Yang, J. Zhou, and J. Tang, “Sequential scenario-specific meta learner for online recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3292500.3330726> pp. 2895–2904.
- [116] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle, “A meta-learning perspective on cold-start recommendations for items,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017*. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/51e6d6e679953c6311757004d8cbbba9-Abstract.html> pp. 6904–6914.
- [117] T. Wei, Z. Wu, R. Li, Z. Hu, F. Feng, X. He, Y. Sun, and W. Wang, “Fast adaptation for cold-start collaborative filtering with meta-learning,” in *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, 2020. [Online]. Available: <https://doi.org/10.1109/ICDM50108.2020.00075> pp. 661–670.
- [118] R. Yu, Y. Gong, X. He, Y. Zhu, Q. Liu, W. Ou, and B. An, “Personalized adaptive meta learning for cold-start user preference prediction,” in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 2021. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/17287> pp. 10 772–10 780.
- [119] Y. Chen, L. Cheng, and C. Chuang, “A group recommendation system with consideration of interactions among group members,” *Expert Syst. Appl.*, vol. 34, no. 3, pp. 2082–2090, 2008. [Online]. Available: <https://doi.org/10.1016/j.eswa.2007.02.008>

- [120] S. Seko, T. Yagi, M. Motegi, and S. Muto, “Group recommendation using feature space representing behavioral tendency and power balance among members,” in *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, 2011. [Online]. Available: <https://doi.org/10.1145/2043932.2043953> pp. 101–108.
- [121] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and W. Cao, “Deep modeling of group preferences for group-based recommendation,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 - 31, 2014, Québec City, Québec, Canada, 2014*. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8458> pp. 1861–1867.
- [122] L. Baltrunas, T. Makcinskas, and F. Ricci, “Group recommendations with rank aggregation and collaborative filtering,” in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, 2010. [Online]. Available: <https://doi.org/10.1145/1864708.1864733> pp. 119–126.
- [123] Q. Yuan, G. Cong, and C. Lin, “COM: a generative model for group recommendation,” in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014. [Online]. Available: <https://doi.org/10.1145/2623330.2623616> pp. 163–172.
- [124] S. Berkovsky and J. Freyne, “Group-based recipe recommendations: analysis of data aggregation strategies,” in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, 2010. [Online]. Available: <https://doi.org/10.1145/1864708.1864732> pp. 111–118.
- [125] L. Boratto and S. Carta, “State-of-the-art in group recommendation and new approaches for automatic identification of groups,” in *Information Retrieval and Mining in Distributed Environments*, 2011, pp. 1–20. [Online]. Available: https://doi.org/10.1007/978-3-642-16089-9_1
- [126] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu, “Group recommendation: Semantics and efficiency,” *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 754–765, 2009. [Online]. Available: <http://www.vldb.org/pvldb/vol2/vldb09-858.pdf>
- [127] F. Brandt, V. Conitzer, and U. Endriss, “Computational social choice,” *Multiagent systems*, pp. 213–283, 2012.
- [128] Z. Yu, X. Zhou, Y. Hao, and J. Gu, “TV program recommendation for multiple viewers based on user profile merging,” *User Model. User Adapt. Interact.*, vol. 16, no. 1, pp. 63–82, 2006. [Online]. Available: <https://doi.org/10.1007/s11257-006-9005-6>
- [129] V. Rakesh, W. Lee, and C. K. Reddy, “Probabilistic group recommendation model for crowdfunding domains,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2835776.2835793> pp. 257–266.

- [130] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *ICLR*, 2015.
- [131] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, “Attentive group recommendation,” in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3209978.3209998> pp. 645–654.
- [132] L. V. Tran, T. N. Pham, Y. Tay, Y. Liu, G. Cong, and X. Li, “Interact and decide: Medley of sub-attention networks for effective group recommendation,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3331184.3331251> pp. 255–264.
- [133] H. Wang, Y. Li, and F. Frimpong, “Group recommendation via self-attention and collaborative metric learning model,” *IEEE Access*, vol. 7, pp. 164 844–164 855, 2019. [Online]. Available: <https://doi.org/10.1109/ACCESS.2019.2953176>
- [134] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020. [Online]. Available: <https://doi.org/10.1007/s10994-019-05855-6>
- [135] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: <http://arxiv.org/abs/1312.6114>
- [136] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2672–2680.
- [137] A. Razavi, A. van den Oord, and O. Vinyals, “Generating diverse high-fidelity images with VQ-VAE-2,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/hash/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Abstract.html> pp. 14 837–14 847.
- [138] S. Tulyakov, M. Liu, X. Yang, and J. Kautz, “Mocogan: Decomposing motion and content for video generation,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Tulyakov_MoCoGAN_Decomposing_Motion_CVPR_2018_paper.html pp. 1526–1535.

- [139] R. A. Yeh, C. Chen, T. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.728> pp. 6882–6890.
- [140] Y. Chen, Y. Wang, M. Kao, and Y. Chuang, “Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Chen_Deep_Photo_Enhancer_CVPR_2018_paper.html pp. 6306–6314.
- [141] W. Nie, N. Narodytska, and A. Patel, “Relgan: Relational generative adversarial networks for text generation,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. [Online]. Available: <https://openreview.net/forum?id=rJedV3R5tm>
- [142] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, “Generative adversarial network for abstractive text summarization,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16238> pp. 8109–8110.
- [143] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, “IRGAN: A minimax game for unifying generative and discriminative information retrieval models,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3077136.3080786> pp. 515–524.
- [144] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. [Online]. Available: <https://openreview.net/forum?id=Bklr3j0cKX>
- [145] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: <http://arxiv.org/abs/1807.03748>
- [146] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, 2020. [Online]. Available: <http://proceedings.mlr.press/v119/chen20j.html> pp. 1597–1607.

- [147] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 2019. [Online]. Available: <https://doi.org/10.18653/v1/n19-1423> pp. 4171–4186.
- [148] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. [Online]. Available: <https://openreview.net/forum?id=H1eA7AEtvS>
- [149] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2020. [Online]. Available: <https://doi.org/10.18653/v1/2020.acl-main.703> pp. 7871–7880.
- [150] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM Comput. Surv.*, vol. 53, no. 3, pp. 63:1–63:34, 2020. [Online]. Available: <https://doi.org/10.1145/3386252>
- [151] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: <https://openreview.net/forum?id=rJY0-Kell>
- [152] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017. [Online]. Available: <http://proceedings.mlr.press/v70/finn17a.html> pp. 1126–1135.
- [153] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021. [Online]. Available: <https://doi.org/10.1109/TNNLS.2020.2978386>
- [154] M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in *Advances in Neural Information Processing Systems*, 2018, pp. 5165–5175.
- [155] Q. Li, Z. Han, and X. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16098> pp. 3538–3545.

- [156] R. A. Rossi and N. K. Ahmed, “Role discovery in networks,” *TKDE*, vol. 27, no. 4, pp. 1112–1131, 2014.
- [157] R. A. Rossi, N. K. Ahmed, A. Carranza, D. Arbour, A. Rao, S. Kim, and E. Koh, “Heterogeneous network motifs,” *arXiv preprint arXiv:1901.10026*, 2019.
- [158] C. Zhuang and Q. Ma, “Dual graph convolutional networks for graph-based semi-supervised classification,” in *WWW*, 2018, pp. 499–508.
- [159] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *ICML*, 2017, pp. 1263–1272.
- [160] A. Paranjape, A. R. Benson, and J. Leskovec, “Motifs in temporal networks,” in *WSDM*. ACM, 2017, pp. 601–610.
- [161] R. A. Rossi, R. Zhou, and N. K. Ahmed, “Estimation of graphlet counts in massive networks,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 1, pp. 44–57, 2019. [Online]. Available: <https://doi.org/10.1109/TNNLS.2018.2826529>
- [162] P. Ribeiro, P. Paredes, M. E. P. Silva, D. Aparício, and F. M. A. Silva, “A survey on subgraph counting: Concepts, algorithms, and applications to network motifs and graphlets,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 28:1–28:36, 2021. [Online]. Available: <https://doi.org/10.1145/3433652>
- [163] L. Peel, J.-C. Delvenne, and R. Lambiotte, “Multiscale mixing patterns in networks,” *PNAS*, vol. 115, no. 16, pp. 4057–4062, 2018.
- [164] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, R. D. Hjelm, and A. C. Courville, “Mutual information neural estimation,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018. [Online]. Available: <http://proceedings.mlr.press/v80/belghazi18a.html> pp. 530–539.
- [165] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.
- [166] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *ICML*, 2018.
- [167] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *CVPR*, 2019, pp. 9268–9277.
- [168] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, “Language modeling with gated convolutional networks,” in *ICML*, 2017, pp. 933–941.
- [169] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017.

- [170] O. Vinyals, S. Bengio, and M. Kudlur, “Order matters: Sequence to sequence for sets,” *arXiv preprint arXiv:1511.06391*, 2015.
- [171] A. Sankar, Y. Wu, Y. Wu, W. Zhang, H. Yang, and H. Sundaram, “Groupim: A mutual information maximization framework for neural group recommendation,” in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401116> pp. 1279–1288.
- [172] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks,” *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.
- [173] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” *Information Retrieval Journal*, vol. 3, pp. 127–163, 2000.
- [174] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2016.
- [175] M. R. Daredddy, M. Das, and H. Yang, “motif2vec: Motif aware node representation learning for heterogeneous networks,” in *BigData*. IEEE, 2019, pp. 1052–1059.
- [176] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, “Learning structural node embeddings via diffusion wavelets,” in *KDD*, 2018, pp. 1320–1329.
- [177] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, “Deep recursive network embedding with regular equivalence,” in *KDD*, 2018, pp. 2357–2366.
- [178] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, “Pitfalls of graph neural network evaluation,” *NeurIPS Workshop*, 2018.
- [179] A. Sankar, X. Zhang, A. Krishnan, and J. Han, “Inf-vae: A variational autoencoder framework to integrate homophily and influence in diffusion prediction,” in *WSDM ’20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3336191.3371811> pp. 510–518.
- [180] A. Krishnan, H. Cheruvu, C. Tao, and H. Sundaram, “A modular adversarial approach to social recommendation,” in *CIKM*. ACM, 2019, pp. 1753–1762.
- [181] K. Narang, C. Yang, A. Krishnan, J. Wang, H. Sundaram, and C. Sutter, “An induced multi-relational framework for answer selection in community question answer platforms,” *arXiv preprint arXiv:1911.06957*, 2019.
- [182] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, “Graph representation learning via graphical mutual information maximization,” in *WWW*, 2020, pp. 259–270.

- [183] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4l: Self-supervised semi-supervised learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1476–1485.
- [184] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” *arXiv preprint arXiv:1906.00910*, 2019.
- [185] Z. Peng, Y. Dong, M. Luo, X.-M. Wu, and Q. Zheng, “Self-supervised graph representation learning via global context prediction,” *arXiv preprint arXiv:2003.01604*, 2020.
- [186] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang, “Node similarity preserving graph convolutional networks,” 2020.
- [187] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [188] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, “Gcc: Graph contrastive coding for graph neural network pre-training,” in *KDD*, 2020, pp. 1150–1160.
- [189] W. Jin, T. Derr, H. Liu, Y. Wang, S. Wang, Z. Liu, and J. Tang, “Self-supervised learning on graphs: Deep insights and new direction,” 2020.
- [190] B. Marr, 2018. [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2018/04/18/netflix-used-big-data-to-identify-the-movies-that-are-too-scary-to-finish/>
- [191] A. Holzapfel, B. Sturm, and M. Coeckelbergh, “Ethical dimensions of music information retrieval technology,” *TISMIR*, vol. 1, no. 1, pp. 44–55, 2018.
- [192] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, “Challenging the long tail recommendation,” *arXiv preprint arXiv:1205.6700*, 2012.
- [193] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, “Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence,” in *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2959100.2959182> pp. 59–66.
- [194] D. Cao, L. Nie, X. He, X. Wei, S. Zhu, and T.-S. Chua, “Embedding factorization models for jointly recommending items and user generated lists,” in *SIGIR*, 2017, pp. 585–594.
- [195] Z. Chen, R. Xiao, C. Li, G. Ye, H. Sun, and H. Deng, “ESAM: discriminative domain adaptation with non-displayed items to improve long-tail performance,” in *SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401043> pp. 579–588.

- [196] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen, “Scalable collaborative filtering using cluster-based smoothing,” in *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, 2005. [Online]. Available: <https://doi.org/10.1145/1076034.1076056> pp. 114–121.
- [197] A. Beutel, K. Murray, C. Faloutsos, and A. J. Smola, “Cobafi: collaborative bayesian filtering,” in *23rd International World Wide Web Conference, WWW ’14, Seoul, Republic of Korea, April 7-11, 2014*, 2014. [Online]. Available: <https://doi.org/10.1145/2566486.2568040> pp. 97–108.
- [198] D. Chae, J. Kim, D. H. Chau, and S. Kim, “AR-CF: augmenting virtual users and items in collaborative filtering for addressing cold-start problems,” in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401038> pp. 1251–1260.
- [199] A. Sankar, Y. Liu, J. Yu, and N. Shah, “Graph neural networks for friend ranking in large-scale social platforms,” in *WWW ’21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2021. [Online]. Available: <https://doi.org/10.1145/3442381.3450120> pp. 2535–2546.
- [200] M. Fan, Y. Bai, M. Sun, and P. Li, “Large margin prototypical network for few-shot relation classification with fine-grained features,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3357384.3358100> pp. 2353–2356.
- [201] M. Luo, F. Chen, P. Cheng, Z. Dong, X. He, J. Feng, and Z. Li, “Metaselector: Meta-learning for recommendation with user-level adaptive model selection,” in *WWW ’20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3366423.3379999> pp. 2507–2513.
- [202] R. S. Sreepada and B. K. Patra, “Mitigating long tail effect in recommendations using few shot learning technique,” *Expert Syst. Appl.*, vol. 140, 2020. [Online]. Available: <https://doi.org/10.1016/j.eswa.2019.112887>
- [203] Q. Suo, J. Chou, W. Zhong, and A. Zhang, “Tadanet: Task-adaptive network for graph-enriched meta-learning,” in *KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3394486.3403230> pp. 1789–1799.
- [204] Y. Wang, D. Ramanan, and M. Hebert, “Learning to model the tail,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/147ebe637038ca50a1265abac8dea181-Abstract.html> pp. 7029–7039.

- [205] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang, “Large-scale few-shot learning: Knowledge transfer with class hierarchy,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Li_Large-Scale-Few-Shot_Learning_Knowledge_Transfer_With_Class_Hierarchy_CVPR_2019_paper.html pp. 7212–7220.
- [206] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, “Large-scale long-tailed recognition in an open world,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_Large-Scale_Long-Tailed_Recognition_in_an_Open_World_CVPR_2019_paper.html pp. 2537–2546.
- [207] S. Gidaris and N. Komodakis, “Dynamic few-shot visual learning without forgetting,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Gidaris_Dynamic-Few-Shot_Visual_CVPR_2018_paper.html pp. 4367–4375.
- [208] G. E. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *CoRR*, vol. abs/1503.02531, 2015. [Online]. Available: <http://arxiv.org/abs/1503.02531>
- [209] W. Park, D. Kim, Y. Lu, and M. Cho, “Relational knowledge distillation,” in *CVPR*, 2019, pp. 3967–3976.
- [210] D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *EMNLP*, 2015, pp. 1422–1432.
- [211] Y. Liu, W. Wei, A. Sun, and C. Miao, “Exploiting geographical neighborhood characteristics for location recommendation,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, 2014. [Online]. Available: <https://doi.org/10.1145/2661829.2662002> pp. 739–748.
- [212] R. He and J. J. McAuley, “VBPR: visual bayesian personalized ranking from implicit feedback,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, 2016. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11914> pp. 144–150.
- [213] J. Leskovec, L. A. Adamic, and B. A. Huberman, “The dynamics of viral marketing,” *ACM Trans. Web*, vol. 1, no. 1, p. 5, 2007. [Online]. Available: <https://doi.org/10.1145/1232722.1232727>

- [214] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, “SEISMIC: A self-exciting point process model for predicting tweet popularity,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, 2015. [Online]. Available: <https://doi.org/10.1145/2783258.2783401> pp. 1513–1522.
- [215] Y. Yang, J. Tang, C. W. Leung, Y. Sun, Q. Chen, J. Li, and Q. Yang, “RAIN: social role-aware information diffusion,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2015. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9384> pp. 367–373.
- [216] J. Zhang, J. Tang, Y. Zhong, Y. Mo, J. Li, G. Song, W. Hall, and J. Sun, “Structinf: Mining structural influence from social streams,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14763> pp. 73–80.
- [217] Y. Wang, H. Shen, S. Liu, J. Gao, and X. Cheng, “Cascade dynamics modeling with attention-based recurrent neural network,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 2017. [Online]. Available: <https://doi.org/10.24963/ijcai.2017/416> pp. 2985–2991.
- [218] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, “Inferring networks of diffusion and influence,” *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 4, pp. 21:1–21:37, 2012. [Online]. Available: <https://doi.org/10.1145/2086737.2086741>
- [219] Y. Zhang, T. Lyu, and Y. Zhang, “Hierarchical community-level information diffusion modeling in social networks,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, 2017. [Online]. Available: <https://doi.org/10.1145/3077136.3080784> pp. 753–762.
- [220] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, “Deepinf: Social influence prediction with deep learning,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3219819.3220077> pp. 2110–2119.
- [221] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, “Social influence locality for modeling retweeting behaviors,” in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013. [Online]. Available: <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6740> pp. 2761–2767.

- [222] A. Sankar, A. Krishnan, Z. He, and C. Yang, “RASE: relationship aware social embedding,” in *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*. [Online]. Available: <https://doi.org/10.1109/IJCNN.2019.8852147> pp. 1–8.
- [223] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, “Dynamic graph representation learning via self-attention networks,” *CoRR*, vol. abs/1812.09430, 2018. [Online]. Available: <http://arxiv.org/abs/1812.09430>
- [224] K. Narang, C. Yang, A. Krishnan, J. Wang, H. Sundaram, and C. Sutter, “An induced multi-relational framework for answer selection in community question answer platforms,” *arXiv preprint arXiv:1911.06957*, 2019.
- [225] C. Xiong, V. Zhong, and R. Socher, “Dynamic coattention networks for question answering,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: <https://openreview.net/forum?id=rJeKjwvclx>
- [226] R. B. Cialdini and N. J. Goldstein, “Social influence: Compliance and conformity,” *Annu. Rev. Psychol.*, vol. 55, pp. 591–621, 2004.
- [227] J. Yang and J. Leskovec, “Modeling information diffusion in implicit networks,” in *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, 2010. [Online]. Available: <https://doi.org/10.1109/ICDM.2010.22> pp. 599–608.
- [228] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [229] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017. [Online]. Available: <http://arxiv.org/abs/1601.00670>
- [230] T. Hogg and K. Lerman, “Social dynamics of digg,” *EPJ Data Sci.*, vol. 1, no. 1, p. 5, 2012. [Online]. Available: <https://doi.org/10.1140/epjds5>
- [231] E. Quintarelli, E. Rabosio, and L. Tanca, “Recommending new items to ephemeral groups using contextual user influence,” in *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2959100.2959137> pp. 285–292.
- [232] A. Delic, J. Masthoff, J. Neidhardt, and H. Werthner, “How to use social relationships in group recommenders: Empirical evidence,” in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, UMAP 2018, Singapore, July 08-11, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3209219.3209226> pp. 121–129.

- [233] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, “Social influence-based group representation learning for group recommendation,” in *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, 2019. [Online]. Available: <https://doi.org/10.1109/ICDE.2019.00057> pp. 566–577.
- [234] D. Cao, X. He, L. Miao, G. Xiao, H. Chen, and J. Xu, “Social-enhanced attentive group recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1195–1209, 2021. [Online]. Available: <https://doi.org/10.1109/TKDE.2019.2936475>
- [235] Y. Zheng, “Identifying dominators and followers in group decision making based on the personality traits,” in *Joint Proceedings of the ACM IUI 2018 Workshops co-located with the 23rd ACM Conference on Intelligent User Interfaces (ACM IUI 2018), Tokyo, Japan, March 11, 2018*, 2018. [Online]. Available: <http://ceur-ws.org/Vol-2068/humanize3.pdf>
- [236] A. Delic, J. Neidhardt, T. N. Nguyen, and F. Ricci, “An observational user study for group recommender systems in the tourism domain,” *J. Inf. Technol. Tour.*, vol. 19, no. 1-4, pp. 87–116, 2018. [Online]. Available: <https://doi.org/10.1007/s40558-018-0106-y>
- [237] X. Liu, Y. Tian, M. Ye, and W. Lee, “Exploring personal impact for group recommendation,” in *21st ACM International Conference on Information and Knowledge Management, CIKM’12, Maui, HI, USA, October 29 - November 02, 2012*, 2012. [Online]. Available: <https://doi.org/10.1145/2396761.2396848> pp. 674–683.
- [238] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada, “Enhancing group recommendation by incorporating social relationship interactions,” in *GROUP*, 2010, pp. 97–106.
- [239] R. Linsker, “Self-organization in a perceptual network,” *Computer*, vol. 21, no. 3, pp. 105–117, 1988. [Online]. Available: <https://doi.org/10.1109/2.36>
- [240] Y. Yeh and Y. Chen, “Qainfomax: Learning robust question answering system by mutual information maximization,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2019. [Online]. Available: <https://doi.org/10.18653/v1/D19-1333> pp. 3368–3373.
- [241] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Comput. Surv.*, vol. 52, no. 1, pp. 5:1–5:38, 2019.
- [242] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, “Deep sets,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html> pp. 3391–3401.

- [243] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. [Online]. Available: <https://openreview.net/forum?id=ryGs6iA5Km>
- [244] P. Cremonesi, Y. Koren, and R. Turrin, “Performance of recommender algorithms on top-n recommendation tasks,” in *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, 2010. [Online]. Available: <https://doi.org/10.1145/1864708.1864721> pp. 39–46.
- [245] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.
- [246] S. Vargas and P. Castells, “Rank and relevance in novelty and diversity metrics for recommender systems,” in *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, 2011. [Online]. Available: <https://dl.acm.org/citation.cfm?id=2043955> pp. 109–116.
- [247] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018. [Online]. Available: <http://proceedings.mlr.press/v80/ren18a.html> pp. 4331–4340.
- [248] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM TOIS*, vol. 22, no. 1, pp. 5–53, 2004.
- [249] Z. Chen, L. Li, and J. Bruna, “Supervised community detection with line graph neural networks,” in *ICLR*. OpenReview.net, 2019.
- [250] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, “Billion-scale commodity embedding for e-commerce recommendation in alibaba,” in *KDD*, 2018, pp. 839–848.
- [251] M. Wang, Y. Lin, G. Lin, K. Yang, and X. Wu, “M2GRL: A multi-task multi-view graph representation learning framework for web-scale recommender systems,” in *KDD*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020, pp. 2349–2358.
- [252] C. Yang, A. Pal, A. Zhai, N. Pancha, J. Han, C. Rosenberg, and J. Leskovec, “Multisage: Empowering gcn with contextualized multi-embeddings on web-scale multipartite networks,” in *KDD*, 2020, pp. 2434–2443.
- [253] A. Jain, I. Liu, A. Sarda, , and P. Molino, “Food discovery with uber eats: Recommending for the marketplace,” 2019. [Online]. Available: <https://eng.uber.com/uber-eats-graph-learning/>
- [254] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

- [255] M. Zhang and Y. Chen, “Weisfeiler-lehman neural machine for link prediction,” in *KDD*, 2017, pp. 575–583.
- [256] M. Zhang and Y. Chen, “Inductive matrix completion based on graph neural networks,” in *ICLR*. OpenReview.net, 2020.
- [257] J. O’Madadhain, J. Hutchins, and P. Smyth, “Prediction and ranking algorithms for event-based network data,” *ACM SIGKDD explorations newsletter*, vol. 7, no. 2, pp. 23–30, 2005.
- [258] D. Ding, M. Zhang, S.-Y. Li, J. Tang, X. Chen, and Z.-H. Zhou, “Baydnn: Friend recommendation with bayesian personalized ranking deep neural network,” in *CIKM*, 2017, pp. 1479–1488.
- [259] P. Cui, X. Wang, J. Pei, and W. Zhu, “A survey on network embedding,” *IEEE TKDE*, vol. 31, no. 5, pp. 833–852, 2018.
- [260] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” *arXiv preprint arXiv:1709.05584*, 2017.
- [261] W. Huang, T. Zhang, Y. Rong, and J. Huang, “Adaptive sampling towards fast graph representation learning,” in *Advances in neural information processing systems*, 2018, pp. 4558–4567.
- [262] J. Chen, J. Zhu, and L. Song, “Stochastic training of graph convolutional networks with variance reduction,” in *ICML*, 2018, pp. 942–950.
- [263] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. K. Prasanna, “Graphsaint: Graph sampling based inductive learning method,” in *ICLR*. OpenReview.net, 2020.
- [264] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, “Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks,” in *KDD*, 2019, pp. 257–266.
- [265] G. Farnadi, J. Tang, M. De Cock, and M.-F. Moens, “User profiling through deep multimodal fusion,” in *WSDM*, 2018, pp. 171–179.
- [266] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, “Multimodal fusion with recurrent neural networks for rumor detection on microblogs,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 795–816.
- [267] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, “Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1437–1445.
- [268] X. Geng, X. Wu, L. Zhang, Q. Yang, Y. Liu, and J. Ye, “Multi-modal graph interaction for multi-graph convolution network in urban spatiotemporal forecasting,” *arXiv preprint arXiv:1905.11395*, 2019.

- [269] I. A. Burhanuddin, P. Bajaj, S. Shekhar, D. Mukherjee, A. Raj, and A. Sankar, “Similarity learning for product recommendation and scoring using multi-channel data,” in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 1143–1152.
- [270] S. Soundarajan, A. Tamersoy, E. B. Khalil, T. Eliassi-Rad, D. H. Chau, B. Gallagher, and K. A. Roundy, “Generating graph snapshots from streaming edge data,” in *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, 2016. [Online]. Available: <https://doi.org/10.1145/2872518.2889398> pp. 109–110.
- [271] N. Shah, D. Koutra, T. Zou, B. Gallagher, and C. Faloutsos, “Timecrunch: Interpretable dynamic graph summarization,” in *KDD*, 2015.
- [272] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, “Friendship prediction and homophily in social media,” *ACM Transactions on the Web (TWEB)*, vol. 6, no. 2, pp. 1–33, 2012.
- [273] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [274] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, “A comparison of extrinsic clustering evaluation metrics based on formal constraints,” *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.
- [275] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damanian et al., “Pytorch distributed: Experiences on accelerating data parallel training,” *arXiv preprint arXiv:2006.15704*, 2020.
- [276] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- [277] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [278] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *NIPS Workshop on Bayesian Deep Learning*, 2016.
- [279] R. v. d. Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” *arXiv preprint arXiv:1706.02263*, 2017.
- [280] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [281] R. Salakhutdinov, “Learning deep generative models,” *Annual Review of Statistics and Its Application*, vol. 2, no. 1, pp. 361–385, 2015. [Online]. Available: <https://doi.org/10.1146/annurev-statistics-010814-020120>

- [282] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, “A survey on contrastive self-supervised learning,” *Technologies*, vol. 9, no. 1, p. 2, 2021. [Online]. Available: <https://www.mdpi.com/2227-7080/9/1/2>
- [283] F. Yuan, L. Yao, and B. Benatallah, “Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019. [Online]. Available: <https://doi.org/10.24963/ijcai.2019/587> pp. 4227–4233.
- [284] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, “PREDATOR: proactive recognition and elimination of domain abuse at time-of-registration,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, 2016. [Online]. Available: <https://doi.org/10.1145/2976749.2978317> pp. 1568–1579.
- [285] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, “Heterogeneous graph neural networks for malicious account detection,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 2018. [Online]. Available: <https://doi.org/10.1145/3269206.3272010> pp. 2077–2085.
- [286] A. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, “Large scale crowdsourcing and characterization of twitter abusive behavior,” in *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, 2018. [Online]. Available: <https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17909> pp. 491–500.
- [287] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, 2019. [Online]. Available: <https://doi.org/10.1109/TNNLS.2018.2886017>
- [288] K. Christakopoulou and A. Banerjee, “Adversarial attacks on an oblivious recommender,” in *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, 2019. [Online]. Available: <https://doi.org/10.1145/3298689.3347031> pp. 322–330.
- [289] Y. Deldjoo, T. D. Noia, and F. A. Merra, “A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 35:1–35:38, 2021. [Online]. Available: <https://doi.org/10.1145/3439729>
- [290] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, “Blurme: inferring and obfuscating user gender based on ratings,” in *Sixth ACM Conference on Recommender Systems, RecSys ’12, Dublin, Ireland, September 9-13, 2012*, 2012. [Online]. Available: <https://doi.org/10.1145/2365952.2365989> pp. 195–202.

- [291] A. Narayanan and V. Shmatikov, “Robust de-anonymization of large sparse datasets,” in *2008 IEEE Symposium on Security and Privacy (S&P 2008)*, 18-21 May 2008, Oakland, California, USA, 2008. [Online]. Available: <https://doi.org/10.1109/SP.2008.33> pp. 111–125.
- [292] Y. Lin, P. Ren, Z. Chen, Z. Ren, D. Yu, J. Ma, M. de Rijke, and X. Cheng, “Meta matrix factorization for federated rating predictions,” in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 2020. [Online]. Available: <https://doi.org/10.1145/3397271.3401081> pp. 981–990.
- [293] S. Wachter-Boettcher, “Ai recruiting tools do not eliminate bias,” *Time Magazine*, 2017. [Online]. Available: <https://time.com/4993431/ai-recruiting-tools-do-not-eliminate-bias/>
- [294] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Comput. Surv.*, vol. 54, no. 6, pp. 115:1–115:35, 2021. [Online]. Available: <https://doi.org/10.1145/3457607>