12-1-2023

Learning heterogeneous subgraph representations for team discovery

Radin Hamidi Rad
Toronto Metropolitan University

Hoang Nguyen
Toronto Metropolitan University

Feras Al-Obeidat
Zayed University

Ebrahim Bagheri
Toronto Metropolitan University

Mehdi Kargar
Toronto Metropolitan University

See next page for additional authors

Recommended Citation

Hamidi Rad, Radin; Nguyen, Hoang; Al-Obeidat, Feras; Bagheri, Ebrahim; Kargar, Mehdi; Srivastava, Divesh; Szlichta, Jaroslaw; and Zarrinkalam, Fattane, "Learning heterogeneous subgraph representations for team discovery" (2023). *All Works*. 6117.
https://zuscholars.zu.ac.ae/works/6117

Radin Hamidi Rad, Hoang Nguyen, Feras Al-Obeidat, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, Jaroslaw Szlichta, and Fattane Zarrinkalam

Learning Heterogeneous Subgraph Representations for Team Discovery

Hoang Nguyen (hoang.cam.nguyen@ryerson.ca)
Toronto Metropolitan University
Radin Hamidi Rad
Toronto Metropolitan University
Feras Al-Obeidat
Zayed University
Ebrahim Bagheri
Toronto Metropolitan University
Mehdi Kargar
Toronto Metropolitan University
Divesh Srivastava
AT&T Chief Data Office
Jaroslaw Szlichta
York University
Fattane Zarrinkalam
University of Guelph

Research Article

Keywords: Expert Search, Heterogeneous Graph Embeddings, Task Assignment, Team Discovery

Posted Date: December 1st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2318594/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Information Retrieval Journal on October 9th, 2023. See the published version at https://doi.org/10.1007/s10791-023-09421-6.

Learning Heterogeneous Subgraph Representations for Team Discovery

Hoang Nguyen[1†], Radin Hamidi Rad[1†], Feras Al-Obeidat[2*], Ebrahim Bagheri[1*], Mehdi Kargar[1*], Divesh Srivastava[3*], Jaroslaw Szlichta[4*] and Fattane Zarrinkalam[5*]

[1]Toronto Metropolitan University, Toronto, ON, Canada.
[2]Zayed University, Dubai, United Arab Emirates.
[3]AT&T Chief Data Office, Bedminster, NJ, USA.
[4]York University, Toronto, ON, Canada.
[5]University of Guelph, Guelph, ON, Canada.

†These authors contributed equally to this work.

Abstract

The team discovery task is concerned with finding a group of experts from a collaboration network who would collectively cover a desirable set of skills. Most prior work for team discovery either adopt graph-based or neural mapping approaches. Graph-based approaches are computationally intractable often leading to sub-optimal team selection. Neural mapping approaches have better performance, however, are still limited as they learn individual representations for skills and experts and are often prone to overfitting given the sparsity of collaboration networks. Thus, we define the team discovery task as one of learning subgraph representations from heterogeneous collaboration network where the subgraphs represent teams which are then used to identify relevant teams for a given set of skills. As such, our approach captures local (node interactions with each team) and global (subgraph interactions between teams) characteristics of the representation network and allows us to easily map between any homogeneous and heterogeneous subgraphs in the network to effectively discover teams. Our experiments over two real-world datasets from different domains, namely the DBLP bibliographic dataset with $\mathbf{10,647}$ papers and IMDB with $\mathbf{4,882}$ movies, illustrate that our approach outperforms the state-of-the-art baselines on a range of ranking and quality metrics. More specifically, in terms

[*]These authors are ordered alphabetically.

of ranking metrics, we are superior to the best baseline by approximately **15%** on the DBLP dataset and by approximately **20%** on the IMDB dataset. Further, our findings illustrate that our approach consistently shows a robust performance improvement over the baselines.

Keywords: Expert Search, Heterogeneous Graph Embeddings, Task Assignment, Team Discovery

1 Introduction

The problem of team discovery from an expert network, distinct from tasks such as expert finding [1] and witness discovery [2], was first introduced by [3]. The major objective of this problem to find a group of collaborative experts who are able to collectively address a task that requires a set of desirable skills. Since it is proved that the team discovery problem for finding teams in the form of subgraphs from a collaboration network is NP-hard, [3] proposed two optimization functions, namely the diameter communication cost and the minimum spanning tree communication cost, to find locally optimum subgraphs from the collaboration network to serve as teams. Subsequent studies primarily focused on defining desirable characteristics for teams, such as having minimal communication cost [4] or including heterogeneous node types to support a wider range of applications [5]. However, the limiting aspect of these approaches is that they are computationally-intractable graph optimization problems as the cost functions are tailored for each team discovery criteria and they are NP-hard.

More recently, researchers have adopted neural architectures, such as autoencoders and variational Bayesian models in order to learn mappings between different node types within a collaboration network to discover teams [6, 7]. Despite showing promising results, these methods overlook different types of interactions between different node types in the collaboration network and solely rely on learning a specific mapping function between the collaboration network node types (e.g., mapping from skill nodes to expert nodes). In practice, collaborations between experts can be viewed through multiple subgraphs in the collaboration network, which necessitates preserving both local (interactions between skills and team members within each team) and global (interactions between different teams through overlaps between their skills and team members) characteristics of the network when finding teams [8]. Therefore, our work goes beyond the limited mappings of existing neural team discovery techniques by learning subgraph representations based on both local (node interactions with each team) and global (subgraph interactions between teams) network characteristics for the sake of team discovery.

More specifically, we learn subgraph representations within heterogeneous collaboration networks that are then used to identify relevant teams for a given set of skills. The crux of our approach is its ability to procure representations for heterogeneous subgraphs. This allows us to map between homogeneous and heterogeneous subgraphs of the collaboration network (e.g., mapping from a skill subgraph consisting of only skill nodes to a team subgraph consisting of experts, skills, and other node types).

1.1 Research Objectives and Contributions

The main objective of this paper is to design a team discovery method from heterogeneous collaboration networks in such a way that it would take the intricacies of expert collaboration into account. We are specifically focused on several key characteristics when designing our team discovery method, which we enumerate as follows:

1. a successful team would need to be able to deliver and accomplish the goals of the task that it is formed to fulfil. In other words, the main purpose of a team is to accomplish a task; therefore, the team members, either individually or collectively, need to possess the right skill sets to accomplish the goals of the task;
2. an efficient team would be one that not only possesses the right skill sets but also consists of team members who are able to work with each other efficiently and have a collaborative team work spirit. A sign of a potentially successful team is one that incorporates members that have effectively worked with each other in the past. As such, a desirable team would be one that has members with fruitful past collaborations;
3. finally, most collaboration networks have two key characteristics: (1) they are quite sparse, i.e., the number of past collaborations between experts as well as the number of skills per expert is low compared to the number of experts and skills in the collaboration network; (2) the size of collaboration networks in terms of the total number of skills, experts and past collaboration is large. These two characteristics make a collaboration network to be a large yet sparse network, which makes designing efficient methods for such graphs difficult. For a team discovery method to be useful in practice, it has to have a reasonable execution time despite having to work on such a large-sparse network structure.

On the basis of these key characteristics, the goal of this paper is to design a team discovery method that is able to efficiently work with large sparse collaboration networks in order to identify teams (subgraphs) whose nodes would collectively satisfy a set of requirements (i.e., cover a set of skills) and that these nodes have effective past interactions (i.e., show past collaboration history). As such, we define the problem of team discovery in collaboration networks as one of learning subgraph representations from heterogeneous graphs where the heterogeneous graph denotes the collaboration network and the subgraphs represent teams. We will both theoretically and empirically show that our work offers the following key contributions:

- While existing neural mapping approaches for team discovery learn individual representations for skills and experts, we learn subgraph representations for teams who have collaborated in the past and skills that were observed in tandem in past teams. For this reason, our approach is able to learn team structure through subgraph representation learning, which is not possible when using neural mapping approaches.
- Our approach learns subgraph representations based on a heterogeneous graphical model rather than directly learning a mapping from the skill space to the expert space in a supervised way. For this reason, our approach is able to overcome the collaboration network sparsity problem and avoid overfitting; hence, addressing the challenge that neural mapping techniques face with regard to overfitting in the face of sparsity.
- Our proposed method also distinguishes itself from existing heterogeneous subgraph representation learning techniques [9, 10] in that it generates embedding vectors for each subgraph by considering node interactions within each subgraph and the interaction of subgraphs with other subgraphs, which has not been captured before.
- Through a range of experiments over real-world datasets, we report that our proposed approach provides significant performance improvements for the team discovery task over the state-of-the-art neural, graph-based and subgraph representation-based techniques. We demonstrate that these observed gains are consistent across different ranking and quality-based metrics.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes the preliminaries and problem definition. Our proposed team discovery learning method is explained in Section 4. Section 5 presents our experiments. Finally, Section 6 sheds light on the future work and concludes the paper.

2 Related Work

In this paper we propose a heterogeneous subgraph representation learning technique for team discovery. In this section, we review the related works in two areas: *team discovery* and *subgraph representation learning*.

2.1 Team Discovery

Heuristic-based approaches with multiple objective functions are applied as the first attempts to solve the team discovery problem. In this group of approaches, the objective function needed to be optimized using either a linear or non-linear programming (IP) which was based on human or non-human factors along with scheduling preferences [11–16]. For example, [11] have used multiple fuzzy objective functions to solve the team discovery problem. They have optimized the suitability metric by maximizing members' fit to the team according to their levels of expertise (skills). They have also optimized the team size by applying human factors (e.g., salary and availability of the candidate experts) and non-human factors (e.g., schedule limitations). This line of work focuses on suggesting experts as the team members individually. Further, in these studies, the selection of an expert is independent of the selection of other experts. Therefore, the optimization objectives do not consider social ties and collaboration between members as effective factors while suggesting team members. Considering the past collaboration can add significantly important information to the team proposal, for instance, an unsuccessful or successful collaboration of the proposed experts in the past can be a good measure of the future project's productivity. However, a team is innately a collaboration among team members and the coherency in the team can effectively define the performance of the team.

Community Search as a well-established problem in network science [17], aims to search for a community of vertices in the graph that can address an input query. Mostly, the queries contain a certain vertex or a group of them. In general, the team formation problem can be interpreted as a special variation of the community search a community, namely attributed community search [18]. This is because in the team formation problem our objective is to find a subgraph from the collaboration network that covers the given set of skills. This can be done using attributed community search where we use nodes to represent experts and each individual's skills as the attributes. Using this analogy, we can search for a community that will cover the input query in form of a group of attributes [18]. Then, the retrieved communities can be considered as our potential teams.

In a more recent study, based on [17]'s cocktail party, [19], utilized a monotone optimization function for team discovery. They tried to minimize the sum of distances between the candidate experts in the induced connected subgraphs. In contrast, [3] used MST and shortest diameter variations to discover potential teams. However, [19] argued that such indexes namely MST and diameter as communication cost can only reflect a limited aspect of a team's communication. [20] proposed a weighted collaboration network where edges and nodes are weighted with respect to the attributes. Similar to [3] and [17], weighted edges can be used to demonstrate the level of successful collaboration in the past and helps minimize the communication cost between proper candidates. Moreover, nodes in the collaboration networks are also weighted. [19] used the h-index in author-network, to represent the importance and seniority of a candidate in the team. Hence, an optimum team is a subgraph from a collaboration network that minimizes the communication cost among weighted edges and maximized the candidates' weights within the subgraph based on their h-index. In another extension to the [3] method, [21, 22] tried to solve the team discovery problem by predicting explicit candidate collaboration in a team based on the potential link utilizing a link prediction technique. Following heuristic approach proposed by [3], they detected the minimum spanning subgraph as the optimum team. This is a promising direction for research as link prediction based on neural representation of graphs is receiving wider attention from the community [23].

There is another line of research in team discovery that can be applied to large-scale collaboration networks. For instance, in community detection algorithms [24–26], compact attributed group detection [27] and keyword search over attributed graph [28, 29]. In these studies, given a skill set as a query, the compact groups will consist of a connected subgraph that contains a set of nodes as experts that are connected via their shortest path. Further, the size of the groups is often limited by the upper and lower bounds.

Another approach that is applied to solve the team discovery problem is by defining the problem as a collaborative filtering task [30, 31] in which the team members are suggested based on the required skills. For example, [32] utilized an autoregressive model based on LSTM to learn behavioural patterns in their Recurrent Recommender Networks (RRN) method. Furthermore, recently, [33] tried to generate feature vectors for the entities based on Bayesian Group Ranking (BGR) and then used them as the input for learning-to-rank architecture for their recommendation.

There are also some studies that have utilized neural networks to solve team discovery problem. For example, [34] used a variational inference neural network to map the given skill set to the experts. Afterwards, using the generated embeddings, they used the learning to rank method to search over the collaboration network for potential teams using their embedded vectors. Moreover, [7] trained an autoencoder that gets trained on the adjacency matrix for team discovery. The model is able to link experts to each other at the inference phase to build up a team. Lastly, [35] proposed a neural network for generating representation vectors for experts. Their approach, namely ExEm, uses the generated embedding later to retrieve experts by calculating the similarity between required skills and experts.

Our approach is the first work employing the subgraph representation learning in the task of team discovery to deal with problems caused by the sparsity of the skill set and the high dimensionality of the expert set while mining the information from historical collaboration records.

2.2 Subgraph Representation Learning

The graph representation learning problem that seeks to learn a low-dimensional feature representation of nodes or entire graphs has been well studied. For example, DeepWalk [36] and Node2Vec [37] learn feature representation of nodes by applying skip-gram model on the sequences of nodes generated by random walk strategy. SDNE [38] and LINE [39] learn feature representation of nodes by optimizing an objective function that preserves first-order and second-order proximities of the graph. Further, due to the powerful ability of graph neural networks (GNNs) to learn feature representations, more recent studies have applied GNN-based methods for graph representation learning [40, 41]. For example, Graph Attention Network (GAT) [41] leverages masked self-attentional layers to learn the importance of nodes by considering the features of neighbors. Similarly, Heterogeneous Attention network [42] leverages both node-level and semantic-level attentions in a hierarchical manner to learn the importance of nodes and meta-paths, respectively. Personalized Propagation of Neural Predictions (PPNP) [43] is derived by incorporating personalized PageRank into graph convolutional networks (GCNs).

However, although some studies have focused on learning representation of specific subgraphs (e.g. rooted subgraphs [44] and Graphlet kernels [45]) or have utilized subgraphs to train GNNs [46, 47], few studies have studied the problem of subgraph representation learning. For example, Sub2Vec [48], learns feature representation of arbitrary subgraphs by applying Paragraph2vec [49] over the samples generated by random walks in subgraphs. The usability of Sub2Vec is shown for community detection and graph classification. Recently, SubGNN [8] is a subgraph-level GNN that propagates neural messages between the subgraph's components and randomly samples patches from the whole graph and aggregates their features to learn feature representations of subgraphs. To improve SubGNN by distinguishing nodes inside and outside the subgraph, GLASS [50], GNN with LAbeling trickS for Subgraph, utilizes an expressive and scalable labelling trick to enhance GNNs for subgraph representation learning. Very recently, to address the scalability issue in the subgraph representation learning problem via GNNs, SUREL [51] reduces the redundancy of subgraph extraction and supports parallel processing by decoupling the graph structure into sets of walks and reusing the walks to form subgraphs.

While most subgraph/graph representation learning techniques are designed for homogeneous graphs and don't consider different types of nodes and relations, recently, a number of studies aimed at designing techniques for heterogeneous graphs [52–54]. For example, MetaGraph2Vec [10] generates heterogeneous node sequences by applying a meta-graph based random walk strategy and then the feature representation of nodes are learned by employing a heterogeneous skip-gram technique over the node sequences. DHNE [9] is a deep hypernetwork-based method that considers a non-linear tuple-wise similarity function in its embedding space while capturing both the local and global structures of a heterogeneous graph. HetGNN [55] applies a random walk with restart strategy to sample a fixed size of strongly correlated heterogeneous neighbors for each node and group them based on node types. Then, a neural network architecture is designed to aggregate feature information of the sampled nodes.

Our approach distinguishes itself from existing heterogeneous subgraph representation learning techniques by generating feature representations for each subgraph by considering node interactions within each subgraph and the interaction of subgraphs with other subgraphs, which has not been captured before.

Table 1 Summary of Frequently Used Notations.

Symbol	Meaning
$\mathcal{G}$	the collaboration network
$\mathcal{V}$	a set of nodes in $\mathcal{G}$
$\mathcal{E}$	a set of edges in $\mathcal{G}$
$\mathcal{Y}$	a set of node types in $\mathcal{G}$
$\mathcal{U}$	a set of expert nodes in $\mathcal{G}, \mathcal{U} \subseteq \mathcal{V}$
$\mathcal{S}$	a set of skill nodes in $\mathcal{G}$, $\mathcal{S} \subseteq \mathcal{V}$
$\mathcal{X}$	the features of all nodes in $\mathcal{V}$
x_i	the features of node $v_i \in \mathcal{V}$
$\mathcal{T}_i$	the team i^{th} in $\mathcal{G}$, $\mathcal{T}_i \subseteq \mathcal{G}$
$\mathcal{A}$	a set of anchor nodes in $\mathcal{G}$
$\mathcal{AS}$	a set of anchor subgraphs in $\mathcal{G}$
C	the number of of anchor subgraph (or anchor nodes) in $\mathcal{G}$
A_c	the anchor node c^{th} in $\mathcal{A}$, i.e. $A_c \in \mathcal{A}$
AS_c	the anchor subgraph c^{th} in $\mathcal{AS}$, i.e. $AS_c \in \mathcal{AS}$
a_c	the representation of the anchor subgraph AS_c, $AS_c \in \mathcal{AS}$
$\mathbf{A}$	the packed representation of all anchor subgraphs in $\mathcal{AS}$
Ω	a neural encoder for the nodes (node sequences) of AS_c, $AS_c \in \mathcal{AS}$
Γ	a neural encoder for subgraph representation learning
t_i	the final representation of the team $\mathcal{T}_i$
msg_i^{intra}	the local representation of the team $\mathcal{T}_i$
msg_i^{inter}	the global representation of the team $\mathcal{T}_i$

3 Preliminaries and Problem Definition

We first introduce notational conventions and then formulate the team discovery problem. Frequently used symbols are summarized in Table 1 for reference.

3.1 Notations

Definition 1 (Collaboration Network) The collaboration network $\mathcal{G}(\mathcal{V}, \mathcal{X}, \mathcal{E})$ is a heterogeneous graph where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ denotes the set of N nodes, $\mathcal{X} \in \mathbb{R}^{Nxd}$ denotes the node features with d-dimension, and $\mathcal{E}$ denotes the edges. Furthermore, each $v_i \in \mathcal{V}$ is affiliated with a type y denoted as v_i^y, which is mapped using the transfer function $\psi(v) : \mathcal{V} \to \mathcal{Y}$, where $\mathcal{Y}$ represents the set of all possible node types.

The heterogeneous collaboration network $\mathcal{G}$ may consists of different types of nodes, including two mandatory types: (i) skills, $\mathcal{S} = \{s_i | s_i \in \mathcal{V}\}$, and (ii) experts, $\mathcal{U} = \{u_j | u_j \in \mathcal{V}\}$; and other additional types: (iii) the product of collaboration (e.g., a paper, a movie, a software product), (iv) the type of the product (e.g., the field of the paper or the genre of the movie), (v) the venue in which the team collaborated (e.g., a conference, a movie studio, or a software lab).

DBLP is a typical example of a collaboration network in the computer science bibliography whose experts correspond to the authors (researchers) and the skills can be considered as the terms extracted from the papers. In addition, there are two other types of nodes in the DBLP dataset which are the paper and the venue. In this dataset, three types of edges can be formed such as paper-author to represent the co-authorship relation, paper-skill representing the skills required to write the paper and paper-venue to represent the place where the paper is published.

Definition 2 (Team) A team $\mathcal{T}_i = \{v \in \mathcal{V}\}$ is the i^{th} subgraph in the collaboration network $\mathcal{G}$ that contains a set of n experts denoted as $\mathcal{U}(\mathcal{T}_i) = \{u_1, u_2, \ldots, u_n\}$; $\mathcal{U}(\mathcal{T}_i) \neq \emptyset$ who collectively cover a predefined set of m required skills denoted as $\mathcal{S}(\mathcal{T}_i) = \{s_1, s_2, \ldots, s_m\}$; $\mathcal{S}(\mathcal{T}_i) \neq \emptyset$.

The collaboration network $\mathcal{G}$ can also be represented as a set of M teams, i.e., $\mathcal{G} = \{(\mathcal{T}_i)\}_{i=1}^{M}$, where $M \leq N$. It is noted that, these teams may have some inter-team connections based on their shared nodes.

Given the DBLP dataset as a collaboration network and a set of terms as the required skills to write a specific paper, a team is formed as a minimal set of author nodes who have co-authored a paper related to the input skills. Figure 1a depicts an example on the DBLP dataset. The collaboration network $\mathcal{G}$ consists of four different node types (e.g., paper, author, term and venue). In the network $\mathcal{G}$, there are three teams $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ (shown as grey dashed circles in Figure 1a) which is defined based on the collaboration project (e.g., an academic paper). Each team is formed by a group of experts/authors (yellow nodes) and it covers a set of skills/terms (green nodes). We may observe that the team $\mathcal{T}_1$ and $\mathcal{T}_2$ both possess skills (a) and (b). Meanwhile, the expert (4) works in two teams, $\mathcal{T}_2$ and $\mathcal{T}_3$.

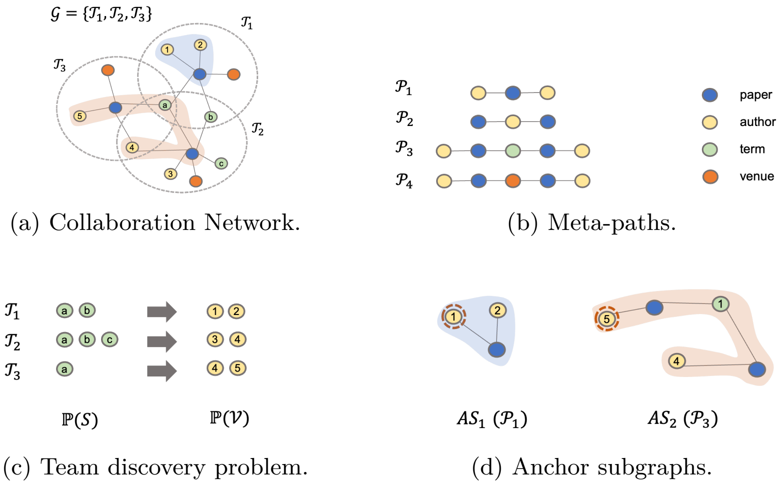
Definition 3 (Meta-path-based subgraph) Given a network $\mathcal{G}$ and a meta-path $\mathcal{P}$, a meta-path-based subgraph in $\mathcal{G}$, denoted as $\mathcal{AS}_i$, is a graph constructed by a set of nodes $\{v \in \mathcal{V}\}$ which are connected to each other via the meta-path $\mathcal{P}$.

A set of some possible meta-paths of the DBLP dataset is depicted in Figure 1b. Considering the two meta-paths $\mathcal{P}_1$ (author-paper-author) and $\mathcal{P}_3$ (author-paper-term-paper-author) with lengths 3 and 5 in this set, two corresponding meta-path-based subgraphs, AS_1 starting from node (1) and AS_2 initiating at node (5), can be formed, as shown in Figure 1d. The length of the path and the size of its corresponding subgraph are equal.

3.2 Team Discovery Problem

Given the collaboration network, $\mathcal{G}$, the team discovery problem is defined as a non-linear mapping function $f : \mathbb{P}(\mathcal{S}) \rightarrow \mathbb{P}(\mathcal{U})$ that maps skill powerset to expert powerset. More specifically, this problem can be formalized as follows.

Definition 4 (Team Discovery Problem) Given the collaboration network $\mathcal{G}$ and a project $\mathcal{Q}$ that requires a pre-defined set of skills $\mathcal{S}_{\mathcal{Q}}$, team discovery problem is



(a) Collaboration Network.

(b) Meta-paths.

(c) Team discovery problem.

(d) Anchor subgraphs.

Fig. 1 Example of definitions in the DBLP dataset.

defined to assign a group of experts $\mathcal{U}_\mathcal{Q}$ in the collaboration network that covers the skills $\mathcal{S}_\mathcal{Q}$ to the project $\mathcal{Q}$.

An illustration of the team discovery problem in the DBLP dataset is shown in Figure 1c. Given a set of desirable skills, the team discovery problem tries to form a set of experts whose skills are similar to the given skill set. An optimal solution for retrieving experts from the skill sets $\{a, b\}, \{a, b, c\}$ and $\{a\}$, is the group of experts $\{1, 2\}, \{3, 4\}$ and $\{4, 5\}$, respectively.

Simply put, we define the problem of team discovery as retrieving subgraphs from within a collaboration network, such that each of the subgraphs represents a team whose experts cover the specified skills. Prior work has already shown that the search for optimal subgraphs is an NP-hard problem [56, 57] and hence opting for solutions which are heuristic-based and sub-optimal [58, 59] to perform graph traversal. For this reason, we propose to learn subgraph representations from heterogeneous collaboration networks, such that relevant subgraphs can be seamlessly identified and retrieved (see Section 4.2).

4 Team Discovery Learning

In this section, we first give an overview of our framework for team discovery learning. Then, We provide more details about two modules of our framework: *team representation learning* and *team retrieval*.

4.1 Overview

Existing neural team discovery techniques from heterogeneous collaboration networks rely only on learning a specific mapping function between the node types (e.g., mapping from skill nodes to expert nodes) and overlook the interactions between nodes. To go beyond such limited mapping, our team discovery approach learns subgraph representations within heterogeneous collaboration networks based on both local (node interactions with each team) and global (subgraph interactions between teams) characteristics of the network that are then used to retrieve relevant teams for a given set of skills.

More specifically, we formulate the team discovery problem as a low-dimensional subgraph representation learning problem. The representation learning generates similar embedding for teams that share similar semantics in the collaboration network $\mathcal{G}$. Intuitively, the semantics of the team can be the profile of the experts in the team, the skills required in the project or the topic of the project. Therefore, given a project with a set of required skills, the latent representation should be encoded in such a way that it allows retrieving the team with a similar profile covering the specified skills and the formed team should be more likely to provide the output (e.g., a published paper) in the specific domain.

We hypothesize that the representation of teams with similar semantics should be highly correlated to each other regardless of their position in the collaboration network. For example, if two groups covering the set of skills which are related, even if they stay multiple hop away, they should still have similar embeddings. Conventional GNNs fail to capture this semantic, as they are based on one-hop message passing, where nodes receive latent representations from their immediate neighbors [60]. Thus, to reduce the bias caused by Conventional GNNs, our idea is to encode a set of sample subgraphs (called *anchor subgraphs*) to capture the semantics of a team and then propagate the neural messages between the team components. Hence, for team representation learning, we propose to utilize a *message passing* technique [61] to share the messages generated by a *meta-path guided random walker* [62, 63] among the anchor subgraphs to effectively capture properties of the team. As a result, the message including the features can be propagated from the source subgraphs which are the anchor subgraphs to the targets which are the subgraphs represented for the teams. On this basis, we generate embedding vectors for each team by employing a hierarchical meta-path aware random walker [64, 65], which is used for generating messages for each team (local interactions within a team or the intra-level team information) and for routing the messages throughout the collaboration network between teams (global interactions across teams or the inter-level team information). More details about our team representation learning approach is explained in Section 4.2.

Next, to retrieve the relevant experts to a project that requires a set of predefined skills, we consider the set of skills as a subgraph and infer its representation via the learned encoder in the former phase. Finally, given the representation of the past teams generated in our team representation learning, we retrieve similar teams to the skills and then select the top-k experts involved in these teams as potential experts for the project.

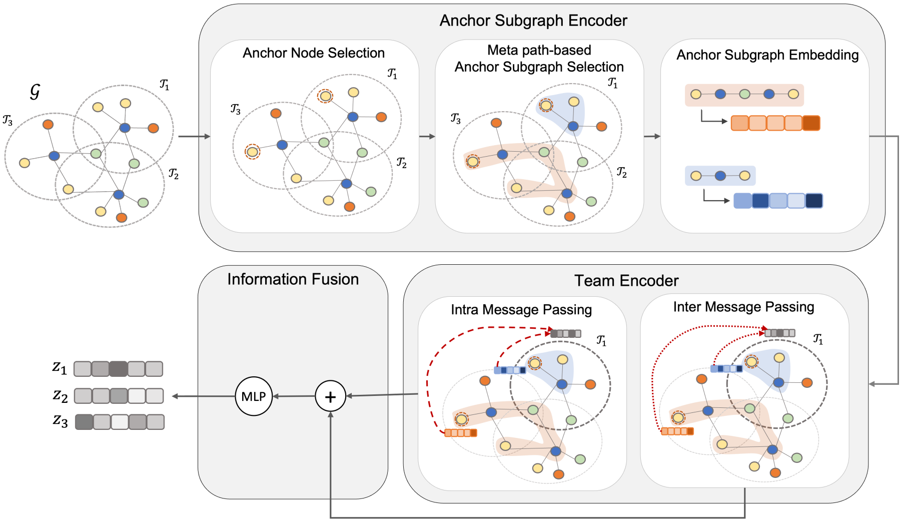


Fig. 2 The overview of our team representation learning approach. **(1) Anchor Subgraph Encoder:** is a three-phase encoder. The anchor nodes shown in the red dashed circles are selected in the first phase. It is then followed by the meta-path-based anchor subgraph selection phase. Two subgraphs highlighted by the blue and pink regions in the figure are generated by the anchor nodes and meta-paths illustrated in Figure 1d. The final phase involves the sequential encoder to embed the meta-path-based subgraphs into latent space. **(2) Team Encoder:** includes intra and inter message passing schemes. The red dashed arrows show an example of how to gather the messages from the two sampled anchor subgraphs to the team $\mathcal{T}_1$ in two different levels. **(3) Information Fusion:** aggregates the two types of representation to produce the final representation for the team

4.2 Team Representation Learning

Given a collaboration network $\mathcal{G}$ represented as a set of M teams, i.e., $\mathcal{G} = \{(\mathcal{T}_i)\}_{i=1}^{M}$, our goal is to map each team to its representation in latent space, i.e., $\Phi(\mathcal{T} \rightarrow \mathbb{R}^{d'})$, where $d' \ll N$. The overview of our team representation learning approach is illustrated in Figure 2. As Figure 2 shows, our proposed encoder Φ consists of three main components: (i) *Anchor Subgraph Encoder*: in this component, we first select a set of anchor nodes; then sample a set of anchor subgraphs starting from the sampled anchor nodes by a meta-path random walker; and finally build a meta-path based neural encoder to learn the embeddings of the anchor subgraphs. (ii) *Team Encoder*: in this component, the embedding features are propagated from the anchor subgraphs to the targeted teams (or subgraphs) in intra-level and inter-level interactions for encoding the intra-team information and inter-team information, respectively. (iii) *Information Fusion*: This component is used to fuse the two types of information to get the unified embeddings for each team.

4.2.1 Anchor Subgraph Encoder

Our anchor subgraph encoder includes three main tasks: (i) Anchor node selection, (ii) Meta path-based anchor subgraph selection, and (iii) Anchor subgraph embedding.

Anchor node selection. Given $\mathcal{G} = \{(\mathcal{T}_i)\}_{i=1}^{M}$, we first select a set of sampled nodes, which are referred to as *anchor nodes A*. For *anchor node selection*, we employ a meta-path aware random walker that crawls through each team individually to find the set of anchor nodes in $\mathcal{G}$. By this strategy, the anchor nodes are selected such that their relative position are corresponding to the position of the teams and they also widely spread across entire the network [66]. These sampled anchor nodes are crucial since they acts like the seed nodes for selecting the anchor subgraphs which are used in order to navigate the messages through the collaboration network $\mathcal{G}$.

The random walker applies the following probability to determine its next node:

$$p\left(v^{i+1} \mid v_y^i, \mathcal{P}\right) = \begin{cases} \frac{1}{\left|N_{y+1}\left(v_y^i\right)\right|} & \left(v^{i+1}, v_y^i\right) \in \mathcal{E}, \psi\left(v^{i+1}\right) = y+1 \\ 0 & \left(v^{i+1}, v_y^i\right) \in \mathcal{E}, \psi\left(v^{i+1}\right) \neq y+1 \\ 0 & \left(v^{i+1}, v_y^i\right) \notin \mathcal{E} \end{cases} \qquad (1)$$

where v_y^i denotes the i^{th} node in a given meta-path $\mathcal{P}$ with node type y. Here, $N_{y+1}\left(v_y^i\right)$ specifies all the next (neighbor) nodes of the node v_y^i based on the specific meta-path $\mathcal{P}$. For the sake of a continuous walk, we design a *symmetric* meta-path schema with the intent that the ending node can be also a starting node for further walks. Thus, in our meta-path patterns, given a metapath $\mathcal{P}$ with length L, the first node type $\psi(v^1)$ is the same as the last node in the meta-path $\psi(v^L)$ [67, 68].

Meta path-based anchor subgraph selection. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_{\mathrm{C}}\}$ be a set of anchor nodes selected in the anchor node selection task, where C is size of the set. For each anchor node $A_c \in \mathcal{A}$, in order to select an anchor subgraph AS_c corresponding to it, a meta-path-based random walker following the same strategy in Equation 1 is employed to generate a length L walk sequence $(A_{c_1}, A_{c_2}, \ldots, A_{c_L})$ initiated at the anchor node A_c ($A_{c_1} \equiv A_c$) and followed a specific meta-path $\mathcal{P}$. The subgraph $AS_c \in \mathcal{AS}$, which is also referred as a meta-path-based anchor subgraph, is then formed by the node sequence $(A_{c_1}, A_{c_2}, \ldots, A_{c_L})$. We denote a set of (meta-path-based) anchor subgraphs corresponding to the set of anchor nodes $\mathcal{A}$ as $\mathcal{AS} = \{AS_1, AS_2, \ldots, AS_{\mathrm{C}}\}$. It is worth to note that the number of anchor nodes equals to the number of anchor subgraphs. In this scenario, the meta-paths are critical in the way that they can capture the semantic relationship between different types of entities in the heterogeneous network which provides more insights for network search and mining. For example, the meta-path 'A(uthor)-P(aper)-S(kill)-P-A' allows to search the authors who share the same skill set or the meta-path 'P-A-P' semantically captures the works of an author. Subsequently, these anchor subgraphs are responsible for sharing properties of each team to others through message passing in the team encoder to achieve more inclusive and efficient team representations.

Anchor subgraph embedding. Once we have the meta-path-based anchor subgraphs, their node sequence information and initial features, we learn an encoder Ω to embed the anchor subgraphs which can capture their semantic meaning. To encode messages of each anchor subgraph or the features represented the messages at the subgraph, we learn an encoder that first iteratively generates sequential messages received by each node within the node sequence (or subgraph), and then aggregate the message from all nodes to use as the anchor subgraph representation. More specifically, the encoder $\Omega : \mathbb{R}^{Lxd} \rightarrow \mathbb{R}^{d_1}$ encodes the sequence of messages represented as $\{x_{c_1}, x_{c_2}, \ldots, x_{c_L}\}$ (where x_{c_L} represents the features of A_{c_L} residing L hops away from A_c) to produce the representation h_{c_i} of the i^{th} node in the sampled sequence as follows:

$$h_{c_i} = \Omega(x_{c_1}, x_{c_2}, \ldots, x_{c_L}) \tag{2}$$

where Ω can be any predefined function that provides for aggregating incremental messages from previously visited nodes. Function Ω can be implemented through an attention mechanism [69] or a long short-term memory network [8]. In this work, we follow the work of [8] that propose to adopt a long short-term memory for this purpose. Now, in order to fuse node embeddings into one representation a_c for anchor subgraph AS_c, we incorporate a READOUT(.) function [70].

$$a_c = READOUT(\left\{ \overrightarrow{h}_{c_l} \parallel \overleftarrow{h}_{c_l} \right\}_{t=1}^{L}) \tag{3}$$

where $\overrightarrow{h}_{c_l}$, $\overleftarrow{h}_{c_l}$ are the output of forward and backward of the message sequences, respectively, and $\parallel$ denotes the aggregation operation. The $READOUT(.)$ function can be any aggregation function, such as summation, maximum, last or average.

We pack the representation of all anchor subgraph computed by Equation 3 and denote as $\mathbf{A}$.

4.2.2 Team Encoder

Given the collaboration network $\mathcal{G} = \{(\mathcal{T}_i)\}_{i=1}^{M}$ and the embedding of selected anchor subgraphs $\mathbf{A}$ generated by the anchor subgraph encoder Ω, we define a *team encoder* which is composed of two separate components to utilize two types of message passing techniques (i.e., Γ^{intra} for *intra-level message passing* and Γ^{inter} for *inter-level message passing*) in order to learn the representation of each team $\mathcal{T}_i$ based on the local and global structural characteristics of the teams. Intuitively, the team representation is characterized locally by the nodes within the team and globally by the connections with the other nodes. Thus, our proposed approach for learning the team representation relies on the messages propagated from two distinct sources, intra and inter team, by defining two separate properties of anchor subgraphs for a team: *intra connection* and *inter connection*, to capture its local and global interactions, respectively.

Definition 5 (Intra Connection Property) Given a team $\mathcal{T}_i$ in a collaboration network $\mathcal{G}$, the intra connections of $\mathcal{T}_i$, denoted by $\mathrm{CP}_{\mathcal{T}_i}^{intra}$, are defined as the set of connections within the team $\mathcal{T}_i$.

Definition 6 (Inter Connection Property) Given a team $\mathcal{T}_i \in \mathcal{G}$, the direct edges connecting the team $\mathcal{T}_i$ and the nodes in the network that do not belong to it forms the set of inter connections of that team, denoted by $\mathrm{CP}_{\mathcal{T}_i}^{inter}$.

The definition of intra and inter connection properties can be generalized to the concept of subgraphs in the network. Therefore, we note that for each anchor subgraph $AS_j \in \mathcal{AS}$ in the network $\mathcal{G}$, there are also two corresponding types of connection property which are intra connections denoted as $\mathrm{CP}_{AS_j}^{intra}$ and inter connections $\mathrm{CP}_{AS_j}^{inter}$.

Formally, a single component in the *team encoder*, $\Gamma : \mathbb{R}^{Cxd_1} \rightarrow \mathbb{R}^{Mxd_1}$, acquires the message msg_i for the i^{th} team from the anchor subgraphs' representation $\mathbf{A}$ using the following message passing scheme:

$$msg_i = \Gamma(\mathcal{T}_i) = \sum_{j=1}^{C} \gamma\left(\mathrm{CP}_{\mathcal{T}_i}, \mathrm{CP}_{AS_j}\right) \cdot a_j \tag{4}$$

where msg_i is the aggregated message from the set of anchor subgraphs $\mathcal{AS}$, γ is a weight function which models the relation between the connection properties of the team (i.e., $\mathrm{CP}_{\mathcal{T}_i}$) and the anchor subgraph (CP_{AS_j}), and a_j is the messages (features) in regard to the j^{th} anchor subgraph. The function γ controls the amount of which can be propagated from the anchor subgraph (AS_j) to the team ($\mathcal{T}_i$) based on their connection properties. We adopt the dynamic time warping function for γ as suggested in [8].

In order to maintain the specific characteristics of different information source, we rather build two separate projections for two types of connection within a team. In other words, for each team $\mathcal{T}_i$, we train two distinct projections: (i) **Intra Message Passing** denoted by Γ^{intra} for computing the similarity of intra connection property (i.e., CP^{intra}) between the ordinary team (i.e., $\mathcal{T}_i$) and the sampled anchor subgraphs (i.e., $AS_j \in AS$) in Equation 5 and (ii) **Inter Message Passing** denoted by Γ^{inter} for computing the similarity between $\mathrm{CP}_{\mathcal{T}_i}^{inter}$ and $\mathrm{CP}_{AS_j}^{inter}$ in Equations 6, to map the anchor subgraphs' representation to a specified team interaction level (i.e., local or global) distribution. Each unique projection measures the importance of a specific type of information regarding the team. As the result, there are two types of encoded message, intra message msg_i^{intra} capturing local information and inter message msg_i^{inter} calculating global information.

$$msg_i^{intra} = \Gamma^{intra}(\mathcal{T}_i) = \sum_{j=1}^{C} \gamma^{intra}\left(\mathrm{CP}_{\mathcal{T}_i}^{intra}, \mathrm{CP}_{AS_j}^{intra}\right) \cdot a_j \tag{5}$$

$$msg_i^{inter} = \Gamma^{inter}(\mathcal{T}_i) = \sum_{j=1}^{C} \gamma^{inter}\left(\mathrm{CP}_{\mathcal{T}_i}^{inter}, \mathrm{CP}_{AS_j}^{inter}\right) \cdot a_j \tag{6}$$

4.2.3 Information Fusion

The final team representation is composed by the aggregation of two types of team information, local information and global information.

$$t_i = AGGREGATE(msg_i^{intra}, msg_i^{inter}) \tag{7}$$

where t_i, msg_i^{intra} and msg_i^{inter} are the final, local and global representation for the team i^{th} respectively; and $AGGREGATE(.)$ denotes aggregation operation. The type of aggregation will be discussed in the experiments section (see Section 5.4).

The fused team embeddings are finally fed into a two-layer Multilayer Perceptron (MLP) as in Equation 8.

$$z_i = \sigma(\mathbf{W}t_i + b) \tag{8}$$

where σ is a non-linear activation function, $\mathbf{W}$ is the learnable weight matrix, and b is bias. The model is trained using a cross-entropy loss function and the Adam optimizer [71].

Supervised Training. We assume that the team representation can be inferred via learning or differentiating the characteristic of the different teams (e.g., research topic of the team or the main theme of the movie). We formulate our proposed team representation learning approach as a supervised learning problem, under this assumption, we use the labels of the nodes in each team as the labels for their corresponding team (e.g., the publication venue node for a paper and the movie genre). It is worth to note that we train our neural network model to predict the team labels (e.g., venue/genre prediction) as a multi-label/multi-class classification task depending on the number of predicted classes of a single instance. For example, a movie can has more than one genre, then predicting a movie is a multi-label classification. We use the following cross entropy ($\mathcal{L}_{CE}$) and binary cross entropy ($\mathcal{L}_{BCE}$) loss function for multi-class and multi-label problem respectively during training step:

$$\mathcal{L}_{CE} = -\sum_{\mathcal{T}_i} \sum_{j=1}^{R} y_{ij} log(\hat{y}_{ij}) \tag{9}$$

where R is the number of labels, $\hat{y}_{ij}$ is the predicted probability of class j^{th} of the i^{th} team and $\hat{y}_i = softmax(z_i)$ is the predicted probability distribution.

$$\mathcal{L}_{BCE} = -\sum_{\mathcal{T}_i} \sum_{j=1}^{R} y_{ij} log(\hat{y}_{ij}) + (1 - y_{ij}) log(1 - \hat{y}_{ij}) \tag{10}$$

where R is the number of class labels, $\hat{y}_{ij} = sigmoid(z_{ij})$ is the predicted probability of the j^{th} class of the i^{th} team and z_{ij} is the logit at j of z_i.

4.3 Team Retrieval

The objective of this step is to retrieve a group of experts to work on a project given a set of desirable skills. Specifically, for a project $\mathcal{Q}$, team discovery involves a mapping function $f : \mathcal{S}_{\mathcal{Q}} \to \mathcal{U}_{\mathcal{Q}}$. The *team retrieval* consists of three main steps: (1) generate the representation of the target team, (2) generate the representation of all the past teams and (3) retrieve the potential team.

In order to generate a team representation for a project $\mathcal{Q}$ given a set of $\mathcal{S}_{\mathcal{Q}}$, we first construct a subgraph $\mathcal{T}_{\mathcal{Q}}$ based on the skill nodes in the $\mathcal{S}_{\mathcal{Q}}$ set. Then, we adopt the encoder Φ learned in the *team representation learning* phase to obtain the embedding $t_{\mathcal{Q}}$. Formally, $t_{\mathcal{Q}} = \Phi(\mathcal{T}_{\mathcal{Q}})$ is computed in Equation 7. Hence, $t_{\mathcal{Q}}$ is considered as the embedding vector for the potential team $\mathcal{T}_{\mathcal{Q}}$ based on the given skill-set $\mathcal{S}_{\mathcal{Q}}$. Ideally, the generated embedding vector $t_{\mathcal{Q}}$ also represents the potential team members $\mathcal{U}_{\mathcal{Q}}$ who are most suitable to be part of the project $\mathcal{Q}$ based on the past collaborations.

In the second step of team retrieval, the representation of each past team collaboration is obtained similarly. The subgraph $\mathcal{T}_i$ is constructed by all the nodes in the team and then its representation t_i is inferred by the encoder Φ. An illustration of how to form the past teams is shown as grey dashed circles in Figure 1a. It is worth to note that the representation of these past teams are optimized during the training phase of team representation learning. Hence, given the collaboration network $\mathcal{G}$ and all the known collaborations $\{\mathcal{T}_i\}$, we can infer their corresponding representations $\{t_i\}$.

Finally, to retrieve the relevant experts $\mathcal{U}_{\mathcal{Q}}$ for the project, all the past collaborations are get sorted out based on the similarity between the representation of the target team (i.e., $t_{\mathcal{Q}}$) and the representation of each team in the set of the historical teams (i.e., $\{t_i\}$). The similarity function can be defined by any similarity distance measures such as Cosine similarity, Jaccard distance or Hadamard product. In this work, we have used Euclidean distance to measure the distance between embedding vectors. After sorting the teams, the top-k teams are chosen as the potential candidates and the experts collaborating in these teams will be listed as the potential team members for the prospective team. In section 5, the top-k relevant experts for each skill-set are used for evaluation purposes.

5 Experiments

In this section, we describe our experiments in terms of the dataset and experimental setup and performance compared to the state-of-the-art.

5.1 Dataset and Experimental Setup

Dataset. We conduct our experiments on two datasets from different domains, namely the DBLP and IMDB datasets. The standard dataset that is commonly used for the team discovery task and has been extensively used in prior literature [3, 6, 72] is based on the DBLP bibliographic database [73]. The assumption in the papers that use this dataset is that the authors associated with paper can be considered to have formed a team with the objective of writing a scientific article. Based on a similar intuition, we use the DBLP dataset released by [6]. This dataset consists of four types of nodes including $1,878$ authors, $10,674$ papers that have at least 2 authors, $2,000$ skills and 21 venues. The teams in the training set are built using the author, paper and skill nodes while the venue nodes are used as the label for the training phase. The edges in each team represent the relationship between papers and authors, as well as papers and skills. In the test set, each team is represented by a paper node, and its corresponding skill nodes. For the sake of evaluation, the edges between papers and authors are predicted.

Similarly, in the IMDB dataset, which covers information about films, each movie is viewed as being the output of a collaborative effort by a set of people, such as actors. Without loss of generality, we assume the set of actors who play in a movie to form a team. This dataset includes $4,882$ movies, which are grouped into 21 genres; and require the involvement of $6,202$ actors as well as $2,532$ skills. Similar to the DBLP dataset, in the training set, we construct a team by connecting each movie node with its neighbor actor nodes and skill nodes. The genre is used as the label for the supervised training. The edges between the movies and the actors in the test set are removed.

For both datasets, we use 10-fold cross-validation for training and evaluating our models against the baselines. We randomly sampled 5% of the training data for validation. We evaluated using top-10 predicted experts for each fold.

Dataset statistics and other information used in the experiments are summarized in Table 2.

Table 2 Statistics of datasets used in our experiments.

Dataset	# Nodes	# Edges	Label node	Meta-paths
IMDB	# actor (A): 6,202 # movie (M): 4,882 # skill (S): 2,532 # genre (G): 21	# M-A: 14,646 # M-S: 42,661 # M-G: 14,040	G	A-M-A M-A-M M-S-M S-M-S A-M-S-M-A
DBLP	# author (A): 1,840 # paper (P): 10,674 # skill (S): 2,000 # venue (V): 21	# P-A: 26,405 # P-S: 70,809 # P-V: 10,674	V	A-P-A P-A-P P-S-P S-P-S A-P-S-P-A

Code and Data. We note that the source code and data for our work is publicly available for reproducibility[i].

Meta-paths. As suggested in [63, 74], in the DBLP dataset, we use 'A(uthor)-P(aper)-A' meta-path which introduces co-authorship, 'A-P-S(kill)-P-A' which denotes relationship between authors with common skills, 'S-P-S' which denotes the required skills of the paper, 'P-S-P' which represents papers focusing on the same skills and 'P-A-P' representing papers published by the same author as the set of meta-paths. In the IMDB dataset, to capture similar relations, we use five different meta-paths, namely 'A(ctor)-M(ovie)-A', 'A-M-S(kill)-M-A', 'S-M-S', 'M-S-M' and 'M-A-M'. These correspond to the five meta-paths for the DBLP dataset. All the meta-paths are reported in Table 2.

Expanding to Other Datasets We took extra notice while developing the code to ensure it is easy to expand the experiments to a new dataset. Thus, while researchers can reproduce the results for datasets studied in this paper, they can study the performance of the proposed method over new datasets. We suggest that researchers take the following pathway to perform their own dataset experiment. The procedure can be done in three steps, (1) starting with the dataset format, using the preprocessing module in the source code, they can prepare the dataset in (sample ID, skill occurrence vector, expert occurrence vector) triplet format. This format is used by other baselines [75] as well. Thus, the preprocessed dataset can base later used to evaluate baselines. (2) In the next step, researchers can run the model and generate representation vectors, and (3) finally, they can evaluate the model using the provided evaluation module in the source code. For a complete walkthrough, we recommend researchers follow the detailed instructions on the project's Github page.

Experimental Setup. For the subgraph embedding methods (i.e., `Metagraph2vec` and `DHNE`), we adopt the implementation in OpenHINE library[ii]. For the remaining baselines, we use the implementation provided by the authors. For all methods, we retain the default hyper-parameter settings proposed by the authors.

The two most important hyper-parameters in our proposed method, the number of anchor subgraphs and the learning rate, are tuned by grid search. The learning rate is searched within the range $[10^{-3}, 10^{-2}]$ and the range of the number of anchor subgraphs is $[35, 80]$. The performance of our proposed approach with varying values of these two hyper-parameters is reported in Section 5.4.

We note that, for a fair comparison between the models, the dimension of the final representations is set to 128 for all methods.

5.2 Baselines

We compare the performance of our methods to several state-of-the-art methods from four different categories:

[i] https://github.com/hoangntc/heterogeneous_subgraph_representation_for_team_discovery

[ii] https://github.com/BUPT-GAMMA/OpenHINE

Graph-based methods. Traditionally, team discovery over a collaboration network has been seen as mining sub-graphs or sub-trees from the graph. Identifying sub-graphs has been shown to be an NP-hard problem. Hence, most studies in this area are based on approximation algorithms. We include the following studies:

- [72], as the strongest work in this group, have modeled the team discovery as keyword search in the graph and tried to find subgraphs from the collaboration network as potential teams.
- [3] is the representative baselines from this category. In this work, authors used minimum subtree diameter as objective function to find the most suitable team based on maximizing the collaboration among members.

Recommender methods. Moreover, the team discovery problem can be defined as a recommender system problem, where for a given set of skills, the recommender system is supposed to find a team from the collaboration network. Collaborative filtering as one of the most common and well-known techniques in this domain can be a promising solution for the team discovery problem. We have included the following methods for this group of studies:

- [76] have proposed a RRN method that utilizes an LSTM-based autoregressive model to capture future behavioral trajectories and also factorization over the skill-expert relations.
- `SVD++` [77] is based on matrix factorization. In this approach, skill-expert interactions are implicitly captured and used in matrix calculation in order to find potential teams.
- `GERF` [33] is based on Bayesian Group Ranking (BGR). In this method, a Bayesian interface is used to optimize the weights of the model which is responsible for generating feature vectors. The generated feature vectors will be later used for learning to rank problem to find relevant experts for a given skill-set.

Neural methods. More recent studies on team discovery have exploited neural architectures to learn a mapping between skill and expert spaces. We adopt two state-of-the-art neural methods as baselines:

- [6] have proposed a method based on variational Bayesian neural network that maps the skill to the expert domain. They have used meta-path-based random walker to craft initial embedding vectors as the input for the model.
- [7] have used an auto-encoder that in essence learns the collaboration between experts implicitly. They used adjacency matrix as the input to represent the link between experts.
- `ExEm` [35] is a neural architecture that generates expert embedding vectors based on their collaborations. Later, they used the embedding vectors to calculate the similarity between experts-skills and rank the experts based on their scores.

Subgraph embedding methods. Given the focus of our method, we also compare our work with the state-of-the-art methods on heterogeneous subgraph representation learning. All the methods are implemented by *OpenHINE*[iii] library. We compare our method against the state-of-the-art methods including:

- `Metagraph2vec` [10] uses meta-paths in form of patterns to guide the random walker traverse though the subgraphs. Due to the consideration of meta-paths that provide semantics to the relationships between the collaboration network node types and edges, this method is able to capture meaningful relations from the graph.
- `DHNE` [9] takes past collaboration information into account when traversing the heterogeneous graph. This approach yields richer representation vectors since the collaboration network is being updated dynamically.

5.3 Evaluation Metrics

We compute several metrics to measure the effectiveness of the proposed model from two perspectives, *ranking-based* and *quality-based.* In the former perspective, we employ four widely adopted information retrieval metrics that have been used in the past for this task [7, 34], including mean average precision (`map`), mean reciprocal rank (`mrr`), normalized discounted cumulative gain (`ndcg`) and `recall`.

Furthermore, we also compute two team quality metrics, namely skill coverage (`sk`) and team comparability (`tc`) [6, 34].

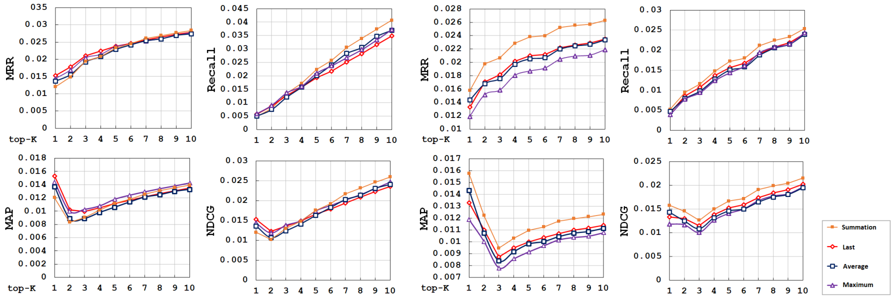
- The skill coverage (`sk`) metric measures to what extent the recommended team covers the set of skills that are specified. The purpose of this metric is to reward those teams that while they do not consist of the exact set of expected experts but still consist of experts that have the required expertise.
- The team comparability (`tc`) metric measures the difference between the average performance of the members of the proposed team and that of the members of the expected team. In the DBLP dataset, h-index for each author is used as the performance index and in the IMDB dataset, the ratio of each movie's gross sales to budget of the movies for each actor is used as the performance indicator.

It is worth noting that a higher `sk` and a lower `tc` values are desirable.

5.4 Ablation Study

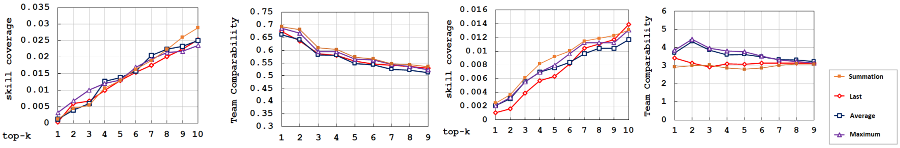
Given the architecture of our proposed method, it is possible to specify three variation points in our work. The first two variations refer to the aggregation functions that are mentioned in Equations 3 and 7. These aggregation functions are used in two corresponding stages, (1) the `anchor subgraph`, and (2) the `subgraph information fusion` embedding generation steps. As

[iii] https://github.com/BUPT-GAMMA/OpenHINE



(a) DBLP. (b) IMDB.

Fig. 3 Anchor subgraph embedding generation variations. The performance results are based on ranking measures.



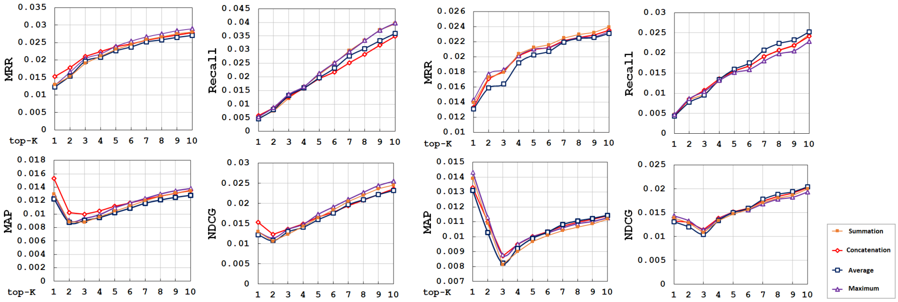
(a) DBLP. (b) IMDB.

Fig. 4 Qualitative results for anchor subgraph embedding generation variations.

noted earlier, these aggregation functions can be implemented using different operators. For $READOUT(.)$ function in Equation 3 which is used for anchor subgraph (node sequence) embedding, we have implemented and evaluated four different aggregation functions, namely, (1) summation, (2) maximum, (3) average and (4) last. Meanwhile, we have conducted the experiment to test graph information fusion embedding with $AGGREGATION(.)$ function in Equation 7 with (1) summation, (2) maximum, (3) average and (4) concatenation.

Starting with the anchor subgraph embedding vectors aggregation method, the results related to ranking-based metrics are shown in Figure 3. As shown in this figure, the summation variation is showing a slightly better performance compared to the other variations on all four ranking metrics. Moreover, the consistency of this trend over both datasets can indicate that in general we can consider the *summation* function to be the preferred method for the aggregation of anchor subgraph embedding vectors. To make the experiments inclusive, we also evaluated the variations over the qualitative metrics. As shown in Figure 4, while the variations are performing close to each other, we observe that, consistent with the performance seen over ranking metrics, the summation operator is slightly better performing.

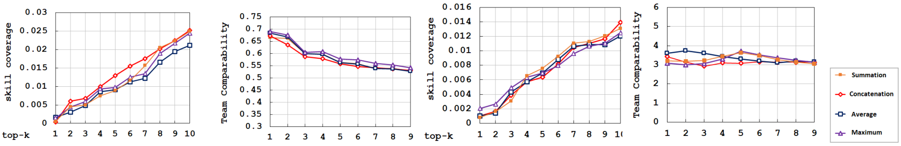
The second variation of our work is due to the possible implementations of the fusion of inter-subgraph and intra-subgraph embedding vectors which is referred to as information fusion. This aggregation function is introduced in



(a) DBLP.

(b) IMDB.

Fig. 5 Fusion embedding generation variations. The performance results are based on ranking measures.



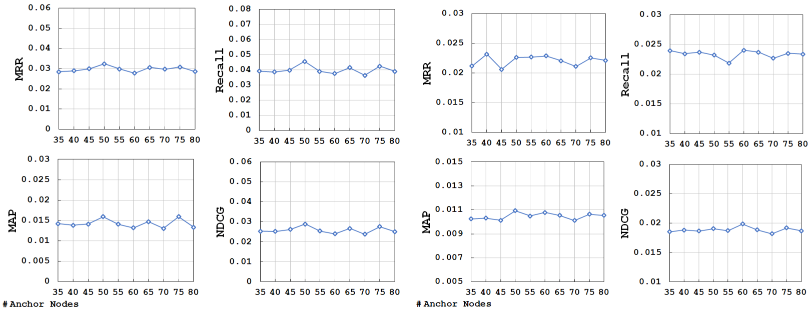
(a) DBLP.

(b) IMDB.

Fig. 6 Information fusion variations performance based on qualitative measures.

Equation 7. Similar to the anchor subgraph aggregation method, four possible implementations can be used to realize Equation 7. The results for the four implementations of the fusion aggregation function are reported in Figure 5. Based on the results of ranking-based metrics, we can conclude that there is no consistent trend across any of the variations. All the variations are performing quite similar to each other and there is no major difference between them. Considering the qualitative metrics in Figure 6, a similar trend can be observed. Hence, we conclude that the proposed approach is robust against different aggregation methods for its fusion stage. For the sake of consistency, we adopt the summation operator to operationalize the fusion aggregation function.

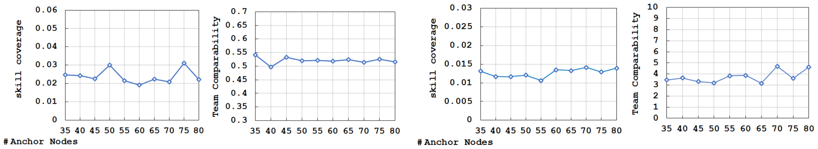
The final variation point of our work relates to the number of anchor subgraphs (or anchor nodes) that is used for routing through the graph. We have evaluated our proposed method with different number of anchor subgraphs in the range of 35 to 80, at cut-off 10.The results for the ranking-based metrics are shown in Figure 7. As shown in this figure, for both of the DBLP and IMDB datasets, the performance does not change noticeably with the change in the number of anchor nodes. This can be considered to be an indication that our proposed approach is robust against changes to the number of anchor subgraphs. We further report the performance of the variations of our proposed approach and its variations based on the qualitative metrics. The results for the qualitative metrics are reported in Figure 8. The results depict a similar



(a) DBLP.

(b) IMDB.

Fig. 7 Impact of number of anchor subgraphs (or anchor nodes) on performance based on ranking measures.



(a) DBLP.

(b) IMDB.

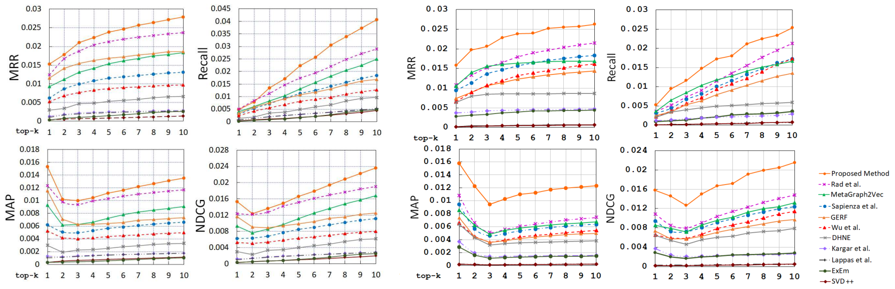
Fig. 8 Qualitative metric results on the DBLP and IMDB datasets. This figure shows impact of number of anchor subgraphs (or anchor nodes) on performance.

consistency in the performance of the various of our work and points to its robustness to the number of anchor nodes.

5.5 Comparison With Baselines

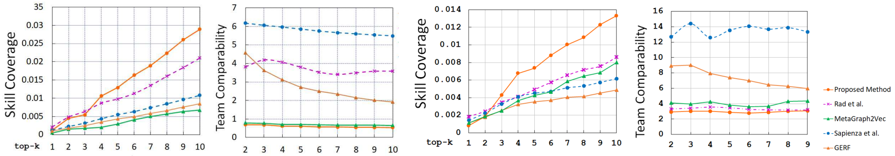
Findings based on Ranking Metrics. The results of experiments based on ranking metrics are reported in Figure 9. Based on this figure, our observations are as follows:

1. We find that our method is able to outperform all of the baselines by a noticeable margin across all metrics. More specifically, we are superior to the best baseline, `Rad et al.` by approximately 18%, 15%, 20%, and 24% for the `mrr`, `map`, `recall` and `ndcg` metrics, respectively on the DBLP dataset and by approximately 22%, 65%, 20%, and 46% for the `mrr`, `map`, `recall` and `ndcg` metrics, respectively on the IMDB dataset. For both comparisons, the top-k cut-off is 10. In other words, 10 experts are retrieved for building a team. For the recall metric, the best baseline model shows comparable results when the size of the team is small, however, our approach achieves noticeable improvements as the size of the team increases.
2. We also observe that neural-based team discovery methods including `Sapienza et al.` and `Rad et al.` show a reasonable performance across



(a) DBLP. (b) IMDB.

Fig. 9 The performance of our proposed model against the baselines on ranking metrics.



(a) DBLP. (b) IMDB.

Fig. 10 Qualitative measures of performance on the DBLP and IMDB datasets.

all metrics (albeit being weaker than our proposed approach on all metrics). This can be due to the fact that neural architectures such as the variational Bayesian neural architecture have shown to be robust even for sparse graphical structures such as collaboration networks and therefore can effectively learn mappings from the skill space to the expert space.

3. Amongst the methods that learn subgraph embeddings, the `Metagraph2vec` method performs better than the `DHNE` method. We believe that similar to our approach that adopts meta-paths, the strength of `Metagraph2vec` is also due to the consideration of meta-paths that provide explicit semantics for the relationships between the collaboration network node types and edges.
4. Finally, our experiments show that even with state-of-the-art graph-based team such as `Kargar et al.`, the performance of these methods do not show competitive performance to other methods such as our approach and methods based on neural architectures. This can be due to these methods being based on sub-optimal local graph search heuristics.

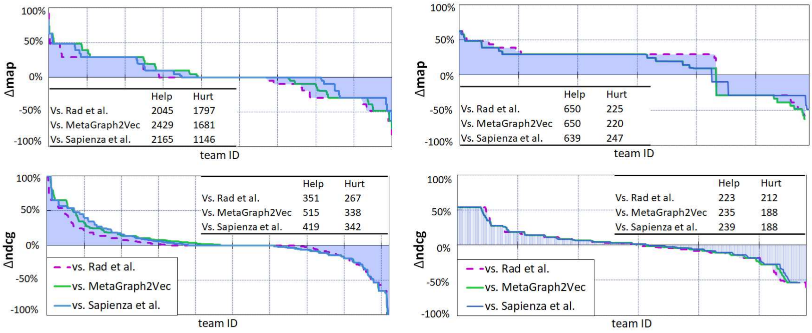
Findings based on Quality Metrics. In order to perform qualitative assessment of the formed teams, we measure quality in terms of the two metrics that were introduced earlier in this section, namely skill coverage (`sk`) and team comparability (`tc`). We compare our proposed approach based on these two metrics against the four most competitive baseline methods that were identified based on ranking metrics.

Based on Figure 10 that shows the performance of our work and the baselines, we make the following observations:

1. From the perspective of skill coverage, we observe that our method has been able to discover teams that have a higher coverage of the required skills compared to the other baselines. This means that even for cases when our method has not been able to find the expected experts, it has been able to find alternative experts that have appropriate skills and this coverage of skills is superior to the coverage provided by the baselines.
2. Unlike the skill coverage metric, for which larger values are more desirable, lower values indicate a more suitable team when measuring the team comparability metric. Based on the findings for the `tc` metric, we observe that our proposed approach has either been able to find teams that consist of the expected experts (those experts that actually were observed in the team present in the test set) or has included experts in the formed teams that have similar stature (h-index in DBLP and discounted gross movie sales to its budget) compared to the expected team. Our method is able to consistently outperform all other baselines.
3. We highlight the fact that while on the skill coverage metric the method by `Rad et al.` showed the second best performance, on the team comparability metric, it does not show competitive performance. Similarly, the `Metagraph2vec` method that shows strong performance on team comparability does not show good performance on skill coverage. However, our method shows good performance on both metrics indicating that teams that are formed by our method are able to provide greater coverage of the required skills and also consist of experts that are either the expected experts or are comparable to the expected experts. This indicates that our proposed method has a more robust performance compared to other strong baselines across the different qualitative metrics.

Robustness of Findings. To show that our proposed approach is able to consistently show a robust performance improvement over the baselines, we also compare the performance of our approach with the best three baselines, i.e., `Rad et al.`, `Metagraph2vec` and `Sapienza et al.` on an individual team basis. The objective of this study is to show that the performance improvements shown by our method over the baselines in earlier experiments were not due to only a limited set of teams in the test set and that such performance improvements can be seen consistently over a large number of teams.

In order to achieve this, without loss of generality and by noting that other metrics show a similar trend, we report help-hurt diagrams over both `map` and `ndcg` metrics. A help-hurt diagram shows (1) the number of, and (2) the percentage degree of improvement or decline of the performance over each given team. A positive value reported in a help-hurt diagram indicates that the performance of the proposed approach is superior to that of the baseline while a negative value indicates poorer performance by the proposed method



(a) DBLP.

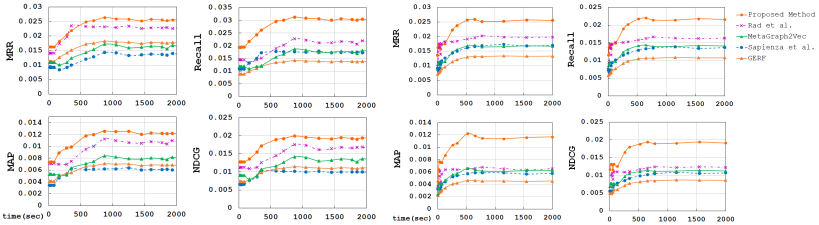
(b) IMDB.

Fig. 11 Performance comparison on a per-team basis over the DBLP and IMDB datasets.

compared to the baseline. The results of the robustness test through help-hurt diagrams have been reported in Figure 11. As seen, the number of teams that have been helped (improved) by our proposed approach has been greater than the number of teams that have been negatively impacted by it. Furthermore, the degree of improvement on those teams that were helped is greater than the extent to which other teams were hurt. This is an indication that our proposed approach has been able to show a robust performance improvement over a range of teams, and hence the performance improvements shown in all experiments have been a result of systematic impact by our proposed approach.

Execution Time Analysis. In order to compare the execution time efficiency of our proposed method compared to the baselines, we studied the performance of our proposed method versus four of the most competitive baseline methods that were identified based on ranking metrics from the *training time* point of view. We trained our proposed method and other baselines for the same amount of time and compared their efficacy using our ranking metrics (i.e., mrr, map, ndcg and recall). The results for both DBLP and IMDB datasets are shown in Figure 12. In this figure, the X-axis shows the time in seconds and the Y-axis shows the performance scores, which are the average of results @5 and @10.

Based on the results, we can observe that in a designated fixed time, all models converge and their performance remains quite stable once they reach the convergence point. However, we find that our proposed method obtains higher performance compared to the other baseline while trained for the same amount of time. The two methods, `Rad et al.` and `Sapienza et al.`, employ an Autoencoder-based architecture which is composed of stacked MLPs. Since the dimension of the input data and the models are relatively small, these methods are highly efficient during training. The graph-based method, `Metagraph2vec`, only involves the computation of the gradient based on the occurrence frequency of the context node pairs in a sampled set of meta-paths random walks. Therefore, the training time is significantly reduced



(a) DBLP.

(b) IMDB.

Fig. 12 Impact of training time on method efficiency for DBLP and IMDB datasets.

compared to graph-based methods which require the computation over the whole network. When comparing the time of our method with the baselines, the results demonstrate that on both datasets, our proposed method is reaching a higher performance in less execution time. It can be concluded that our proposed method is computationally efficient on both datasets. Regarding the model complexity, our proposed method does not perform neural computing for all the nodes in the network. Instead, it undertakes message passing from the sampled anchor subgraphs to the team. Therefore, the scalability of our proposed model depends on the number of teams and the number of anchor subgraphs. The number of teams is smaller than the number of total nodes. Further, our method only requires a small number of anchor subgraphs to achieve a good performance as shown in Figure 7 and 8. Therefore, our approach is able to scale better compared to other methods.

5.6 Discussion

Based on the reported experiments in the previous section, we make a set of observations with regards to our proposed approach and some of the state-of-the-art baselines, which we summarize as follow:

1. Most traditional approaches towards team discovery have primarily focused on designing heuristic methods based on deterministic graph traversal mechanisms with some desirable characteristics. While effective on small-scale graphs, they are not effective in terms of execution time and team discovery quality on realistic collaboration networks, which are large and sparse. The reason for this is the fact that most of these deterministic approaches are designed based on subgraph optimization methods, which have been shown to be a reduced version of the Steiner-tree problem, which is an NP-hard by nature [78]. Methods based on this approach will be heuristics-based by nature and therefore may lack scalability and/or effectiveness.
2. The second observation is with regards to the comparative performance of the baselines from neural mapping approaches compared to subgraph representation learning methods. We find that overall subgraph representation learning methods are quite effective for the team discovery task especially

when considering the `Metagraph2vec` method. This method is consistently among the top performing methods despite it not being designed specifically for team discovery. It even perform better than two of the neural mapping baselines that are specifically introduced for team discovery, namely `ExEm` and `Sapienza et al.` This can be due to the fact that graph representation learning methods consider interaction between graph nodes, which implicitly captures past expert collaborations and their skills. However, in neural mapping approaches, past collaboration history and interaction between expert's expertise is not explicitly captured. This is more clearly observed over the quality metrics where the `Metagraph2vec` method shows a very competitive performance.

3. Finally, we have shown based on our experiments that our proposed approach is (1) *stable*, i.e., its performance is not sensitive the architectural variations shown through our ablation studies, (2) is accurate and effective from both perspective of quantitative ranking metrics as well as quality metrics, (3) is robust by showing that it improve the performance of the team discovery task over a large number of test samples compared to several strong state-of-the-art baselines. This is achieved over two real-world large datasets, namely DBLP and IMDB, and (4) diverges to a higher performance score in terms of ranking metrics compared to the baselines while it is trained for the same amount of time.

We believe that a key to an effective team discovery method is to consider the interaction between skills, experts and skills-experts. Our proposed approach effectively captures these interactions by learning subgraph representation from the collaboration network that encode the interactions explicitly. This is specifically why our proposed approach shows better performance compared to other team discovery methods especially neural mapping methods. While neural mapping approaches learn effective skill and expert representations and useful mapping across these representations, they fall short by not capturing the interaction between the expert-skill spaces when learning the representations.

6 Concluding Remarks

We propose a team representation learning approach for finding teams of experts based on a desirable set of skills from collaboration networks. The novelty of our work is in its formulation of the team discovery problem as one of learning team representations from heterogeneous collaboration networks. Our method adopts a meta-path guided random walker as well as a message passing schema to capture local and global structural characteristics of the teams. Through comparing with a range of the state-of-the-art team discovery and heterogeneous subgraph representation learning baselines over the DBLP and IMDB datasets, we show that our proposed approach is effective and robust for finding expert teams from the collaboration network in terms of both ranking and quality metrics.

Our proposed model has two main limitations which we aim to address in our future work: **(1)** Our method ignores the development or the change of experts' skills. Currently, most of the team discovery methods capture the past history of expert skills and expert collaboration as a static collaboration network. The fact is that the experts may change their interests or acquire new skills, and furthermore, they establish new collaborations and abandon some past collaborations. Therefore, as our future work, we are particularly interested in looking into the dynamics of expert skills and expert collaborations to discover more relevant teams. **(2)** Our method retrieves the relevant current teams based on the representation of the past teams encoded via the representation of the the anchor subgraphs. Hence, its performance depends on the goodness of the sampled anchor subgraphs. In our work, we select these subgraphs randomly without considering their importance. Therefore, in our future work, we would like to investigate the method to capture the importance of the anchor subgraph in order to select the striking ones for learning the team representation.

Declarations

- Funding: The research leading to these results received funding from NSERC
- Conflict of interest: The authors have no competing interests to declare that are relevant to the content of this article.
- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Availability of data and materials: Data for our work is publicly available at https://github.com/hoangntc/heterogeneous_subgraph_representation_for_team_discovery.
- Code availability: The source code of our work is publicly available for reproducibility at https://github.com/hoangntc/heterogeneous_subgraph_representation_for_team_discovery.
- Authors' contributions: Hoang Nguyen and Radin Hamidi Rad implemented the method and ran experiments. Fattane Zarrinkalam and Ebrahim Bagheri wrote and revised the manuscript. Feras Al-Obeidat, Mehdi Kargar, Divesh Srivastava and Jaroslaw Szlichta revised manuscript and analyzed results.

References

[1] Neshati, M., Fallahnejad, Z., Beigy, H.: On dynamicity of expert finding in community question answering. Information Processing & Management **53**(5), 1026–1042 (2017)

[2] Liu, W., Cao, J., Yang, Y., Liu, B., Gao, Q.: Category-universal witness discovery with attention mechanism in social network. Information Processing & Management **59**(4), 102947 (2022)

[3] Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009, pp. 467–476. ACM, ??? (2009). https://doi.org/10.1145/1557019.1557074. https://doi.org/10.1145/1557019.1557074

[4] An, A., Kargar, M., Zihayat, M.: Finding affordable and collaborative teams from a network of experts. In: Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA, pp. 587–595. SIAM, ??? (2013). https://doi.org/10.1137/1.9781611972832.65. https://doi.org/10.1137/1.9781611972832.65

[5] Yang, J., Li, M., Wu, B., Xu, C.: Forming a research team of experts in expert-skill co-occurrence network of research news. In: Kumar, R., Caverlee, J., Tong, H. (eds.) 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016, pp. 1143–1150. IEEE Computer Society, ??? (2016). https://doi.org/10.1109/ASONAM.2016.7752383. https://doi.org/10.1109/ASONAM.2016.7752383

[6] Rad, R.H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: Retrieving skill-based teams from collaboration networks. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pp. 2015–2019. ACM, ??? (2021). https://doi.org/10.1145/3404835.3463105. https://doi.org/10.1145/3404835.3463105

[7] Sapienza, A., Goyal, P., Ferrara, E.: Deep neural networks for optimal team composition. Frontiers Big Data **2**, 14 (2019). https://doi.org/10.3389/fdata.2019.00014

[8] Alsentzer, E., Finlayson, S.G., Li, M.M., Zitnik, M.: Subgraph neural networks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual (2020). https://proceedings.neurips.cc/paper/2020/hash/5bca8566db79f3788be9efd96c9ed70d-Abstract.html

[9] Yin, Y., Ji, L., Zhang, J., Pei, Y.: DHNE: network representation learning method for dynamic heterogeneous networks. IEEE Access **7**, 134782–134792 (2019). https://doi.org/10.1109/ACCESS.2019.2942221

[10] Zhang, D., Yin, J., Zhu, X., Zhang, C.: Metagraph2vec: Complex semantic path augmented heterogeneous network embedding. CoRR

abs/1803.02533 (2018) 1803.02533

[11] Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. Cybern. Syst. **38**(2), 155–185 (2007). https://doi.org/10.1080/01969720601139041

[12] Fitzpatrick, E., Askin, R.G.: Forming effective worker teams with multi-functional skill requirements. Comput. Ind. Eng. **48**(3), 593–608 (2005). https://doi.org/10.1016/j.cie.2004.12.014

[13] Durfee, E.H., Jr., J.C.B., Sleight, J.: Using hybrid scheduling for the semi-autonomous formation of expert teams. Future Gener. Comput. Syst. **31**, 200–212 (2014). https://doi.org/10.1016/j.future.2013.04.008

[14] Mirzaei, M., Sander, J., Stroulia, E.: Multi-aspect review-team assignment using latent research areas. Inf. Process. Manag. **56**(3), 858–878 (2019). https://doi.org/10.1016/j.ipm.2019.01.007

[15] Neshati, M., Beigy, H., Hiemstra, D.: Expert group formation using facility location analysis. Inf. Process. Manag. **50**(2), 361–383 (2014). https://doi.org/10.1016/j.ipm.2013.10.001

[16] Dehghan, M., Rahmani, H.A., Abin, A.A., Vu, V.: Mining shape of expertise: A novel approach based on convolutional neural network. Inf. Process. Manag. **57**(4), 102239 (2020). https://doi.org/10.1016/j.ipm.2020.102239

[17] Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: Rao, B., Krishnapuram, B., Tomkins, A., Yang, Q. (eds.) Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010, pp. 939–948. ACM, ??? (2010). https://doi.org/10.1145/1835804.1835923. https://doi.org/10.1145/1835804.1835923

[18] Fang, Y., Cheng, R., Luo, S., Hu, J.: Effective community search for large attributed graphs. Proc. VLDB Endow. **9**(12), 1233–1244 (2016). https://doi.org/10.14778/2994509.2994538

[19] Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Macdonald, C., Ounis, I., Ruthven, I. (eds.) Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011, pp. 985–994. ACM, ??? (2011). https://doi.org/10.1145/2063576.2063718. https://doi.org/10.1145/2063576.2063718

[20] Zihayat, M., An, A., Golab, L., Kargar, M., Szlichta, J.: Authority-based team discovery in social networks. In: Markl, V., Orlando, S.,

Mitschang, B., Andritsos, P., Sattler, K., Breß, S. (eds.) Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017, pp. 498–501. OpenProceedings.org, ??? (2017). https://doi.org/10.5441/002/edbt.2017.54. https://doi.org/10.5441/002/edbt.2017.54

[21] Keane, P., Ghaffar, F., Malone, D.: Using machine learning to predict links and improve steiner tree solutions to team formation problems. In: Cherifi, H., Gaito, S., Mendes, J.F., Moro, E., Rocha, L.M. (eds.) Complex Networks and Their Applications VIII - Volume 2 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019, Lisbon, Portugal, December 10-12, 2019. Studies in Computational Intelligence, vol. 882, pp. 995–1006. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-36683-4_79. https://doi.org/10.1007/978-3-030-36683-4_79

[22] Keane, P., Ghaffar, F., Malone, D.: Using machine learning to predict links and improve steiner tree solutions to team formation problems - a cross company study. Appl. Netw. Sci. **5**(1), 57 (2020). https://doi.org/10.1007/s41109-020-00306-x

[23] Wu, Z., Zhan, M., Zhang, H., Luo, Q., Tang, K.: Mtgcn: A multitask approach for node classification and link prediction in graph data. Information Processing & Management **59**(3), 102902 (2022)

[24] Fang, Y., Cheng, R., Luo, S., Hu, J.: Effective community search for large attributed graphs. Proc. VLDB Endow. **9**(12), 1233–1244 (2016). https://doi.org/10.14778/2994509.2994538

[25] Fani, H., Jiang, E., Bagheri, E., Al-Obeidat, F.N., Du, W., Kargar, M.: User community detection via embedding of social network structure and temporal content. Inf. Process. Manag. **57**(2), 102056 (2020). https://doi.org/10.1016/j.ipm.2019.102056

[26] Li, X., Sun, C., Zia, M.A.: Social influence based community detection in event-based social networks. Inf. Process. Manag. **57**(6), 102353 (2020). https://doi.org/10.1016/j.ipm.2020.102353

[27] Khan, A., Golab, L., Kargar, M., Szlichta, J., Zihayat, M.: Compact group discovery in attributed graphs and social networks. Inf. Process. Manag. **57**(2), 102054 (2020). https://doi.org/10.1016/j.ipm.2019.102054

[28] Kargar, M., Golab, L., Srivastava, D., Szlichta, J., Zihayat, M.: Effective keyword search over weighted graphs. IEEE Transactions on Knowledge and Data Engineering, 1–1 (2020). https://doi.org/10.1109/TKDE.2020.2985376

[29] Bryson, S., Davoudi, H., Golab, L., Kargar, M., Lytvyn, Y., Mierzejewski, P., Szlichta, J., Zihayat, M.: Robust keyword search in large attributed graphs. Inf. Retr. J. **23**(5), 502–524 (2020). https://doi.org/10.1007/s10791-020-09379-9

[30] Seyedhoseinzadeh, K., Rahmani, H.A., Afsharchi, M., Aliannejadi, M.: Leveraging social influence based on users activity centers for point-of-interest recommendation. Information Processing & Management **59**(2), 102858 (2022). https://doi.org/10.1016/j.ipm.2021.102858

[31] Ramos, G., Boratto, L., Caleiro, C.: On the negative impact of social influence in recommender systems: A study of bribery in collaborative hybrid algorithms. Information Processing & Management **57**(2), 102058 (2020)

[32] Wu, C., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: de Rijke, M., Shokouhi, M., Tomkins, A., Zhang, M. (eds.) Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017, pp. 495–503. ACM, ??? (2017). https://doi.org/10.1145/3018661.3018689. https://doi.org/10.1145/3018661.3018689

[33] Du, Y., Meng, X., Zhang, Y., Lv, P.: GERF: A group event recommendation framework based on learning-to-rank. IEEE Trans. Knowl. Data Eng. **32**(4), 674–687 (2020). https://doi.org/10.1109/TKDE.2019.2893361

[34] Rad, R.H., Fani, H., Kargar, M., Szlichta, J., Bagheri, E.: Learning to form skill-based teams of experts. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, pp. 2049–2052. ACM, ??? (2020). https://doi.org/10.1145/3340531.3412140. https://doi.org/10.1145/3340531.3412140

[35] Nikzad-Khasmakhi, N., Balafar, M.A., Feizi-Derakhshi, M., Motamed, C.: Exem: Expert embedding using dominating set theory with deep learning approaches. Expert Syst. Appl. **177**, 114913 (2021). https://doi.org/10.1016/j.eswa.2021.114913

[36] Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Macskassy, S.A., Perlich, C., Leskovec, J., Wang, W., Ghani, R. (eds.) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pp. 701–710. ACM, ??? (2014). https://doi.org/10.1145/2623330.2623732. https://doi.org/10.1145/2623330.2623732

[37] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks.

In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pp. 855–864. ACM, ??? (2016). https://doi.org/10.1145/2939672.2939754. https://doi.org/10.1145/2939672.2939754

[38] Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pp. 1225–1234. ACM, ??? (2016). https://doi.org/10.1145/2939672.2939753. https://doi.org/10.1145/2939672.2939753

[39] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: Gangemi, A., Leonardi, S., Panconesi, A. (eds.) Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015, pp. 1067–1077. ACM, ??? (2015). https://doi.org/10.1145/2736277.2741093. https://doi.org/10.1145/2736277.2741093

[40] Kipf, T.N., Welling, M.: 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. OpenReview.net, ??? (2017). https://openreview.net/forum?id=SJU4ayYgl

[41] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, ??? (2018). https://openreview.net/forum?id=rJXMpikCZ

[42] Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The World Wide Web Conference. WWW '19, pp. 2022–2032. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3308558.3313562. https://doi.org/10.1145/3308558.3313562

[43] Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, ??? (2019). https://openreview.net/forum?id=H1gL-2A9Ym

[44] Narayanan, A., Chandramohan, M., Chen, L., Liu, Y., Saminathan, S.: subgraph2vec: Learning distributed representations of rooted sub-graphs from large graphs. CoRR **abs/1606.08928** (2016) 1606.08928

[45] Yanardag, P., Vishwanathan, S.V.N.: Deep graph kernels. In: Cao, L., Zhang, C., Joachims, T., Webb, G.I., Margineantu, D.D., Williams, G. (eds.) Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, pp. 1365–1374. ACM, ??? (2015). https://doi.org/10.1145/2783258.2783417. https://doi.org/10.1145/2783258.2783417

[46] Huang, K., Zitnik, M.: Graph meta learning via local subgraphs. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual (2020). https://proceedings.neurips.cc/paper/2020/hash/412604be30f701b1b1e3124c252065e6-Abstract.html

[47] Wang, J., Chen, P., Xie, Y., Shan, Y., Xuan, Q., Chen, G.: Broad learning based on subgraph networks for graph classification. In: Graph Data Mining, pp. 49–71. Springer, ??? (2021)

[48] Adhikari, B., Zhang, Y., Ramakrishnan, N., Prakash, B.A.: Sub2vec: Feature learning for subgraphs. In: Phung, D.Q., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10938, pp. 170–182. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-93037-4_14. https://doi.org/10.1007/978-3-319-93037-4_14

[49] Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. JMLR Workshop and Conference Proceedings, vol. 32, pp. 1188–1196. JMLR.org, ??? (2014). http://proceedings.mlr.press/v32/le14.html

[50] Wang, X., Zhang, M.: Glass: Gnn with labeling tricks for subgraph representation learning. In: International Conference on Learning Representations (2021)

[51] Yin, H., Zhang, M., Wang, Y., Wang, J., Li, P.: Algorithm and system co-design for efficient subgraph-based graph representation learning. CoRR **abs/2202.13538** (2022) 2202.13538

[52] Shi, C., Wang, X., Yu, P.S.: Heterogeneous Graph Representation Learning and Applications. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, ??? (2022). https://doi.org/10.1007/978-981-16-6166-2. https://doi.org/10.1007/978-981-16-6166-2

[53] Li, Z., Zhao, Y., Zhang, Y., Zhang, Z.: Multi-relational graph attention networks for knowledge graph completion. Knowl. Based Syst. **251**, 109262 (2022). https://doi.org/10.1016/j.knosys.2022.109262

[54] Li, Z., Liu, H., Zhang, Z., Liu, T., Xiong, N.N.: Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Trans. Neural Networks Learn. Syst. **33**(8), 3961–3973 (2022). https://doi.org/10.1109/TNNLS.2021.3055147

[55] Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19, pp. 793–803. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3292500.3330961. https://doi.org/10.1145/3292500.3330961

[56] Veldt, N., Benson, A.R., Kleinberg, J.M.: The generalized mean densest subgraph problem. CoRR **abs/2106.00909** (2021) 2106.00909

[57] Liu, B., Zhang, F., Zhang, W., Lin, X., Zhang, Y.: Efficient community search with size constraint. In: 37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021, pp. 97–108. IEEE, ??? (2021). https://doi.org/10.1109/ICDE51399.2021.00016. https://doi.org/10.1109/ICDE51399.2021.00016

[58] Preti, G., Morales, G.D.F., Riondato, M.: Maniacs: Approximate mining of frequent subgraph patterns through sampling. In: Zhu, F., Ooi, B.C., Miao, C. (eds.) KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021, pp. 1348–1358. ACM, ??? (2021). https://doi.org/10.1145/3447548.3467344. https://doi.org/10.1145/3447548.3467344

[59] Liu, X., Pan, H., He, M., Song, Y., Jiang, X., Shang, L.: Neural subgraph isomorphism counting. In: Gupta, R., Liu, Y., Tang, J., Prakash, B.A. (eds.) KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020, pp. 1959–1969. ACM, ??? (2020). https://doi.org/10.1145/3394486.3403247. https://doi.org/10.1145/3394486.3403247

[60] You, J., Ying, R., Leskovec, J.: Position-aware graph neural networks. In: International Conference on Machine Learning, pp. 7134–7143 (2019). PMLR

[61] Xiao, T., Chen, Z., Wang, D., Wang, S.: Learning how to propagate messages in graph neural networks. In: Zhu, F., Ooi, B.C., Miao, C. (eds.) KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021, pp.

1894–1903. ACM, ??? (2021). https://doi.org/10.1145/3447548.3467451. https://doi.org/10.1145/3447548.3467451

[62] Yan, S., Wang, H., Li, Y., Zheng, Y., Han, L.: Attention-aware metapath-based network embedding for HIN based recommendation. Expert Syst. Appl. **174**, 114601 (2021). https://doi.org/10.1016/j.eswa.2021.114601

[63] Fu, X., Zhang, J., Meng, Z., King, I.: MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In: Huang, Y., King, I., Liu, T., van Steen, M. (eds.) WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pp. 2331–2341. ACM / IW3C2, ??? (2020). https://doi.org/10.1145/3366423.3380297. https://doi.org/10.1145/3366423.3380297

[64] Zhang, Z., Huang, J., Tan, Q., Sun, H., Zhou, Y.: Cmg2vec: A composite meta-graph based heterogeneous information network embedding approach. Knowl. Based Syst. **216**, 106661 (2021). https://doi.org/10.1016/j.knosys.2020.106661

[65] Wang, X., Lu, Y., Shi, C., Wang, R., Cui, P., Mou, S.: Dynamic heterogeneous information network embedding with meta-path based proximity. IEEE Transactions on Knowledge and Data Engineering, 1–1 (2020). https://doi.org/10.1109/TKDE.2020.2993870

[66] Liu, C., Li, X., Zhao, D., Guo, S., Kang, X., Dong, L., Yao, H.: Graph neural networks with information anchors for node representation learning. Mob. Networks Appl. **27**(1), 315–328 (2022). https://doi.org/10.1007/s11036-020-01633-0

[67] Sun, Y., Han, J.: Mining heterogeneous information networks: principles and methodologies. Synthesis Lectures on Data Mining and Knowledge Discovery **3**(2), 1–159 (2012)

[68] Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proc. VLDB Endow. **4**(11), 992–1003 (2011)

[69] Wu, M., Pan, S., Du, L., Tsang, I., Zhu, X., Du, B.: Long-short distance aggregation networks for positive unlabeled graph learning. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), pp. 2157–2160 (2019)

[70] Hamilton, W.L.: Graph representation learning. Synthesis Lectures on Artifical Intelligence and Machine Learning **14**(3), 1–159 (2020)

[71] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In:

Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980

[72] Kargar, M., Golab, L., Srivastava, D., Szlichta, J., Zihayat, M.: Effective keyword search over weighted graphs. IEEE Transactions on Knowledge and Data Engineering **34**, 601–616 (2022). https://doi.org/10.1109/TKDE.2020.2985376

[73] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Li, Y., Liu, B., Sarawagi, S. (eds.) Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008, pp. 990–998. ACM, ??? (2008). https://doi.org/10.1145/1401890.1402008. https://doi.org/10.1145/1401890.1402008

[74] Botterman, H., Lamarche-Perrin, R.: Link weights recovery in heterogeneous information networks. CoRR **abs/1906.11727** (2019) 1906.11727

[75] Rad, R.H., Mitha, A., Fani, H., Kargar, M., Szlichta, J., Bagheri, E.: Pytfl: A python-based neural team formation toolkit. In: Demartini, G., Zuccon, G., Culpepper, J.S., Huang, Z., Tong, H. (eds.) CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, pp. 4716–4720. ACM, ??? (2021). https://doi.org/10.1145/3459637.3481992. https://doi.org/10.1145/3459637.3481992

[76] Wu et al., C.: Recurrent recommender networks. In: de Rijke, M., Shokouhi, M., Tomkins, A., Zhang, M. (eds.) Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017, pp. 495–503. ACM, ??? (2017). https://doi.org/10.1145/3018661.3018689. https://doi.org/10.1145/3018661.3018689

[77] Koren, Y.: Collaborative filtering with temporal dynamics. Commun. ACM **53**(4), 89–97 (2010). https://doi.org/10.1145/1721654.1721677

[78] Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. The IBM Research Symposia Series, pp. 85–103. Plenum Press, New York, ??? (1972). https://doi.org/10.1007/978-1-4684-2001-2_9. https://doi.org/10.1007/978-1-4684-2001-2_9