Graph Neural Team Recommendation: An Integrated Approach

Anonymous Author(s)

Abstract

Team recommendation aims to select an optimal subset of experts that can form an *almost surely* successful collaborative team for a given set of required skills. State-of-the-art methods employ neural multi-label classifiers that *transfer* dense vector representations of skills, pretrained via graph neural networks on an expert collaboration graph, into a multi-hot vector representation of the optimal expert subset. Such methods, however, overlook experts' relational and structural information encoded in the collaboration graph and, thus, fall short of capturing complex inter-dependencies among experts and their associated skills within (un)successful teams. Moreover, the skills' dense vectors are pretrained *disjointly* from the underlying neural classifier, hence, preventing end-to-end optimization. In this paper, we propose to reformulate the team recommendation problem into end-to-end link predictions between expert nodes and the required skill nodes *directly* in the expert collaboration graph to consume multi-hop intra-team and cross-team collaborations among experts while eschewing the unnecessary complexities of the disjoint two-phase training procedure. Our experiments on two large-scale datasets from various domains, with distinct distributions of skills in teams, demonstrate the superiority of our proposed approach against the transfer-based approach.[1]

CCS Concepts

- **Human-centered computing** → *Social recommendation.*

Keywords

Neural Team Recommendation; Social Information Retrieval;

1 Introduction

Collaborative teamwork in which combined skills is integrated through coordinated effort to tackle complex tasks, has become paramount given the expanding scope and complexity of emerging challenges. These problems arise in diverse real-world settings, including assembling ad-hoc medical response teams [35, 43], assigning peer reviewers in scientific publication [1, 2, 31], and forming collaborative student groups in large-scale educational environments [5, 36], where expertise, availability, and coordination constraints must be jointly considered. Consequently, forming teams of experts whose success is *almost surely* guaranteed has been a surge of research for years to replace the tedious, error-prone, and suboptimal manual search process; from operations research [4, 12, 13, 25, 45, 48, 52, 54], social network analysis [19, 26, 44], and more recently, machine learning [7, 15, 38, 40]. Specifically, neural models [7, 8, 37, 38, 41, 42] have brought state-of-the-art efficacy while enhancing efficiency due to the iterative learning procedure and availability of training datasets. Neural models, by and large, frame the team recommendation problem as a *multi-label* Boolean classification task and learn the vector representations of experts and their skills in a training dataset of successful and (virtually) unsuccessful teams to draw future teams that are, more likely than not, successful. Such work, however, was premised on the mutually independent selection of experts and overlooked the collaboration ties among experts in the past (un)successful teams. Indeed, a team is inherently relational, and its success depends on the intra-team and cross-team collaborations among experts.

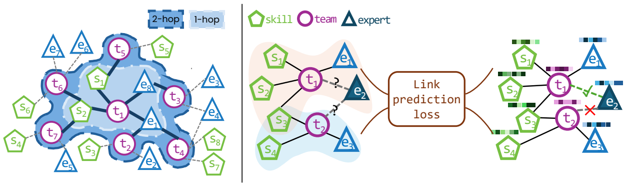

Figure 1: Multi-hop structure in the expert graph (left). Our end-to-end team recommendation approach (right).

In this paper, we propose to model collaborative ties of experts within and across teams in a graph and leverage graph neural networks to capture multi-hop relational and structural information encoded in the graph for complex inter-dependencies among experts and their associated skills within teams as shown in Figure 1 (left). We then propose to reformulate the team recommendation problem into end-to-end link predictions between pairs of required subsets of skill nodes and optimal expert nodes for a team node *directly* within the expert collaboration graph. As highlighted in Figure 1 (right), given a successful team, we map it into a connected subgraph that connects the team's node to its required subset of skills and expert members. After incorporating all successful teams of a training set in the graph, we then apply a graph neural network to learn dense vectors of skills and experts *jointly* based on link prediction loss among such subgraphs using positive (existing) and negative (non-existing) link sampling strategies. During inference, as shown in Figure 1 (right), given a test team with its subset of required skills yet *unseen* expert members, we use the graph neural network to predict links between expert nodes and the team's node.

While employing graphs to model experts' collaborations was mainstream in traditional team recommendation approaches rooted in social network analysis, they have received little attention in recent machine learning–based methods wherein graphs are used only for the prior pretraining and *transfer* of dense vector representations of skills to accommodate a large set of skills and reduce the neural classifiers' input complexity [28, 37, 38], overlooking experts and their collaborations within and across teams. For instance, Rad et al. [38] used `metapath2vec` [10] on a skill-team-expert tripartite graph to pretrain and transfer skill embeddings as the input for a variational Bayesian neural classifier [38]. Alternatively, Kaw et al. [28] employed deep graph infomax [47], a graph convolution network with attention layer, for more effective embeddings of skills. Such approaches entail two key shortcomings: (1) they suffer from independent expert selection probabilities in the multi-label classification framework, and (2) the skill embeddings are learned *disjointly* in a self-supervised manner during a pretraining phase, oblivious to the team recommendation learning process. Our approach, while effectively capturing multi-hop graph structure information for both skills and experts among teams, eschews

[1] The codebase is publicly available; however, we omit the link for double-blind review. The implementation is part of a unified framework that cannot be fully anonymized.

unnecessary two-phase learning complexity yielding improved efficiency, and is inspired by the well-established effectiveness of *end-to-end* graph neural networks in various recommendation and information retrieval tasks [6, 14, 18, 22, 23, 29, 34, 49, 50, 55], which has not been explored in team recommendation before our study.

2 Neural Team Recommendation

Given a set of skills $\mathcal{S} = \{s_i\}$ and a set of experts $\mathcal{E} = \{e_j\}$, a team t is a subset of experts $\mathbf{e} \subseteq \mathcal{E}$ that collectively cover a subset of skills $\mathbf{s} \subseteq \mathcal{S}$ along with its success status y, denoted by $t_{\mathbf{s},\mathbf{e},y\in\{0,1\}}$. Further, $\mathcal{T} = \mathcal{T}^+ \cup \mathcal{T}^- = \{t_{\mathbf{s},\mathbf{e},y=1}\} \cup \{t_{\mathbf{s},\mathbf{e},y=0}\}$ is the set of successful and unsuccessful teams. For a subset of skills $\mathbf{s}$, a neural team recommender aims to identify an optimal subset of experts $\mathbf{e}$ such that their collaboration results in a successful team, i.e., $t_{\mathbf{s},\mathbf{e},y=1}$, while avoiding any subset $\mathbf{e}'$ that leads to an unsuccessful team, i.e., $t_{\mathbf{s},\mathbf{e}',y=0}$. Specifically, the objective is to learn a mapping function f, parameterized by θ, from the powerset of skills to the powerset of experts such that $\forall t \in \mathcal{T}^+$: $f_\theta(\mathbf{s}) = \mathbf{e}$ and $\forall t \in \mathcal{T}^-$: $f_\theta(\mathbf{s}) \neq \mathbf{e}'$.

Existing methods [7, 28, 37, 38, 40] estimate $f_\theta(\mathbf{s}) \approx f_\theta(v_\mathbf{s}) = v_\mathbf{e}$ using a neural multi-label classifier that transfers the *dense* vector representation of a subset of skills, $v_\mathbf{s} \in \mathbb{R}^d$, to the occurrence (multi-hot) vector representation of the optimal (successful) subset of experts, $v_\mathbf{e} \in \{0, 1\}^{|\mathcal{E}|}$, by maximizing the posterior probability of θ in f_θ over $\mathcal{T}$ in a variational Bayesian neural architecture via minimizing Kullback-Leibler divergence [32].

Limitation 1: Existing approaches frame the team recommendation problem as a multi-label Boolean classification task based on the mutually independent selection of experts, ignoring both cross-team relational structures and multi-hop itra-team dependencies.

3 Expert Collaboration Graph

To transfer dense vector representation of a subset of skills $v_\mathbf{s}$, state-of-the-art neural team recommenders [28, 37, 38] construct a heterogeneous tripartite graph $\mathbb{G} = \langle \mathcal{N}, \mathcal{L} \rangle$ whose nodes are skills $n_\mathcal{S}$, successful teams $n_{\mathcal{T}^+}$ and experts $n_\mathcal{E}$, i.e., $\mathcal{N} = n_\mathcal{S} \cup n_{\mathcal{T}^+} \cup n_\mathcal{E}$, and links are undirected $\mathcal{L} = \{n_\mathcal{S} \times n_{\mathcal{T}^+}\} \cup \{n_{\mathcal{T}^+} \times n_\mathcal{E}\}$, and train a graph neural network g, parameterized by ϕ, $g_\phi : \mathcal{N} \to \mathbb{R}^d, g_\phi(n_{s_i}) = v_{s_i}$, via link prediction loss. From Figure 1 (right), each team $t_{\mathbf{s},\mathbf{e},y=1} \in \mathcal{T}^+$ is represented by a *star* subgraph with a team node as the central node and its required skills $\mathbf{s}$ and expert members $\mathbf{e}$ are the leaves, connected to the central team node but not to each other. The dense vector of the required subset of skills $\mathbf{s}$ for a team $t_{\mathbf{s},\mathbf{e},y}$ is then obtained by summing the vectors of its constituent skill nodes, i.e., $v_\mathbf{s} = \sum_{s_i \in \mathbf{s}} v_{s_i} = \sum_{s_i \in \mathbf{s}} g_\phi(n_{s_i})$.

Limitation 2: g_ϕ is estimated to capture structural and semantic information of skills only, completely disregarding the expert and their collaborations within the context of historically successful teams. Also, it is *prior* to and *disjoint* from the main team recommendation task and is kept constant and oblivious to the multi-label classification loss during estimating f_θ.

Table 1: Statistics of the raw and preprocessed datasets.

	dblp		imdb	
teams $\mathcal{T}$	publications		movies	
experts $\mathcal{E}$	authors		cast & crew	
skills $\mathcal{S}$	keywords		(sub) genres	
success y	published		produced	
statistics	raw	filtered	raw	filtered
$\lvert\mathcal{T}\rvert$	4,877,383	99,375	507,034	32,059
$\lvert\mathcal{E}\rvert$	5,022,955	14,214	876,981	2,011
$\lvert\mathcal{S}\rvert$	**89,504**	**29,661**	**28**	**23**

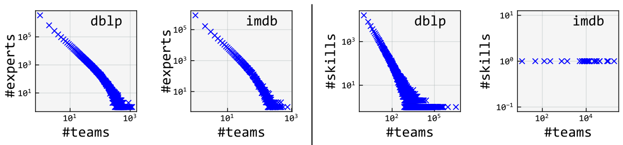

Figure 2: Distribution of teams over experts and skills.

4 Proposed Approach

In this paper, we propose to estimate f directly through the link prediction loss on the expert collaboration graph. Formally, $\forall t_{\mathbf{s},\mathbf{e},y=1} \in \mathcal{T}^+, \forall e_j \in \mathbf{e}$, we estimate $f : \{(n_\mathcal{S}, n_{\mathcal{T}^+}) \times n_\mathcal{E}\} \to \{0, 1\}$ in $\mathbb{G}\langle \mathcal{N}, \mathcal{L} \rangle$ as the perfect cosine similarity between the embeddings of a team node $g_\phi(n_{t_{\mathbf{s},\mathbf{e},y=1}})$ and its expert members $g_\phi(n_{e_j})$:

$$f(\mathbf{s} : \mathbb{G}, g_\phi) = cos(g_\phi(n_{e_j}), g_\phi(n_{t_{\mathbf{s},\mathbf{e},y=1}})) = \texttt{1.0} \qquad (1)$$

Intuitively, $f(\mathbf{s})$ outputs `1.0` to indicate the presence of links between optimally selected experts $e_j \in \mathbf{e}$ for a team $t_{\mathbf{s},\mathbf{e},y}$ with required subset of skills $\mathbf{s}$ such that the team is highly likely to succeed, i.e., $t_{\mathbf{s},\mathbf{e},y=1} \in \mathcal{T}^+$. Our proposed end-to-end formulation directly taps into multi-hop relational and structural information encoded in the graph for complex cross-team interactions and intra-team-dependencies among experts and their associated skills, capturing relations beyond independent expert probabilities assumed in multi-label classification, as shown in Figure 1 (left); a capacity in which feedforward multi-label classifiers are inherently limited. Moreover, it avoids two disjoint learning processes with two sets of parameters; that is, pretraining of g_ϕ and fine-tuning for f_θ, thereby excelling in computational efficiency and resource usage. Herein, ϕ is the sole learnable set of parameters for the entire team recommendation pipeline.

5 Experiments

In this section, we seek to answer our main research question:

RQ1: *Does the end-to-end approach outperform the transfer-based approach for team recommendation?* If affirmative, we further address the following RQs:

RQ2: *Is **RQ1**'s answer consistent across datasets from varied domains with distinct distribution of teams over skills?*

RQ3: *Is **RQ1**'s answer consistent across different graph structures?*

RQ4: *Which graph neural network performs the best (worst) in the end-to-end and transfer-based approaches for team recommendation?*

5.1 Datasets

We used two well-known benchmark datasets as in prior works [7, 27, 30, 39, 40]: **dblp**, a collection of computer science publications [9], where a *published* paper is a successful team, the authors are the expert members, and the keywords are the required skills, and **imdb**, a collection of movies [24], where a *produced* movie is considered as a successful team, the cast and crew are the experts, and the movie's genres and subgenres are the required skills. In contrast to movie recommenders or movie review analysis, herein, the objective is to form a team for a successful movie production.

Teams with fewer than 3 expert members were removed while ensuring no alteration to the statistical distributions, as detailed in Table 1. From Figure 2 (left), both datasets exhibit long-tailed distributions of teams over experts, i.e., few dominant experts have contributed to many teams, while the majority have participated scarcely. Regarding skills, from Figure 2 (right), while dblp follows a similar long-tail distribution of teams over skills, imdb demonstrates a more uniform distribution across a limited set of skills (genres), which are consistently used in many movies.

For scalability in such large-scale datasets, we implemented our approach as well as the baselines on pyg [16, 17], as detailed below.

5.2 Baselines

We compared the end-to-end approach (e2e-*) with the following transfer-based (t-*) baselines:

t-bnn.m2v [38]: It is the pioneer to transfer embeddings of skills by metapath2vec [10] for a variational Bayesian neural classifier.

t-bnn.lant [28]: It employs deep graph infomax [47] for more effective embeddings of skills in fewer training epochs.

To ensure a comprehensive comparison between the end-to-end and transfer-based approaches, we also included strong graph neural networks for a heterogeneous graph[2]:

graphsage (gs) [20]: It aggregates neighbours' vectors by a fixed function, herein, mean, followed by concatenation with the node's current vector.

gin [53]: It learns node vectors via *learnable* aggregation and combination functions, which, theoretically, show the best node embeddings by capturing graph structure based on the Weisfeiler-Lehman graph isomorphism test [11].

gine [21]: It extends gin to include global graph features based on a subgraph of neighbors located between k_1-hop and k_2-hop of a node where $k_1 < k_2$, and is designed to utilize link features.

gat [46]: It aggregates neighbors with *learnable* multi-head attention scores based on the importance of neighbors when aggregating and combining vectors during message passing.

gatv2 [3]: As opposed to gat, which computes pairwise attention between a node and its neighbors using shared parameters, it uses a different set of parameters for a node and each of its neighbors that are updated through layers (hops).

han [51]: It is a hierarchical gat that captures the importance of different node types and their predefined meta-paths to neighbours.

All graph neural networks included convolutional layers of size **64**, followed by a relu activation, after a search over different sizes of {16, 32, 64, 128, 256} across dblp and imdb datasets. For message passing, all neighbors were considered with a negative sampling at a ratio of 5:1. We employed 1-hop sampling with 20 nodes, 2-hop sampling with 20 and 10 nodes in the first and second hops, i.e., [20,10], and 3-hop sampling with [20,10,5] and [30,20,10], representing different node allocations across hops. From Figure 3, we observed generally favorable results with **2**-hop sampling and **[20,10]** across datasets, though not uniformly across all models. All graph neural networks were trained for 5 epochs in batch sizes of 125 with a learning rate of 0.001 using Adam.

For a minimum baseline in performance, we further considered two *naive* vector representations of skills, as in early works [39, 40]:

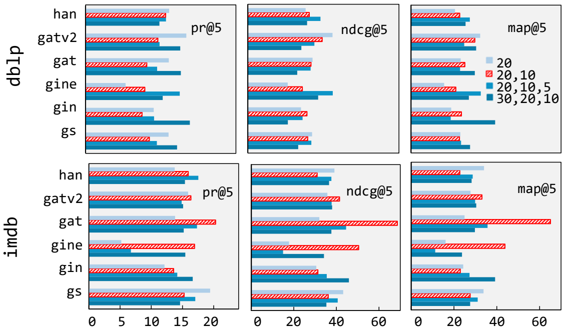

Figure 3: Impact of neighborhood sampling.

d2v: It is a document-based embeddings of skills, wherein each team is considered as a document whose words are the skills and uses doc2vec [33] with a context window of 10.

mhot: It is the $|\mathcal{S}|-$dimensional sparse occurrence (multi-hot) vector of skills to represent a subset of skills $\mathbf{s}$.

For the transfer-based, we cross-compare vector representations of skills under variational Bayesian (t-bnn-*) and non-variational (t-fnn-*) neural classifiers, following their settings in [38, 39].

5.3 Evaluation Strategy and Metrics

We used the entire dataset to create the expert collaboration graph. For the teams in the test set, which comprise 15% of the randomly shuffled dataset, we removed the team-expert links $(n_t, n_e) \in \mathcal{L}$ to mask the expert members as *unseen* during training. Given a team $t_{\mathbf{s},\mathbf{e},y=1}$ from the test set, a trained model predicts the membership probability of all experts for the team by team-expert link predictions. We selected a subset of experts $\hat{\mathbf{e}}$ with the top$-k$ highest probabilities as the recommended team of size $k \in \{5, 10\}$ and compared it with the correct subset of experts $\mathbf{e}$ and reported the average performance of models in terms of classification metrics including precision (pr) and recall (rec), and ranking metrics including normalized discounted cumulative gain (ndcg) and mean average precision (map), which penalize a model if a correct expert of the test team is at the lower part of the top-k recommendations.

6 Results

RQ1: End-to-end vs. transfer-base approach: From Table 2, we observe that the end-to-end approach (e2e-*) consistently and substantially outperforms *all* transfer-based methods using either t-fnn-* or t-bnn-* classifiers across all employed graph neural networks, datasets, and metrics. The superiority of the end-to-end approach lies in leveraging both multi-hop relational and structural information in the expert collaboration and supervised information about the optimal subset of experts within successful teams.

RQ2: Cross-domain performance consistency. From Table 2, we can observe that the performance improvement by graph neural networks is consistent for both datasets in our end-to-end approach. However, in the transfer-based approach, the performance varies in variational and non-variational models with *no* consistent trend based on their underlying graph neural network for different datasets. Yet, attentive models, t-*-han, t-*-gat, or t-*-gatv2, are generally stand out as the top-performers.

RQ3: Cross-graph structural performance consistency. To study the impact of graph structure, we considered *i*) skill-team-expert-**location** where a team is further connected to its location,

[2] torch_geometric.nn.conv.HeteroConv.html

Table 2: Efficacy of end-to-end vs. transfer-based approaches in dblp and imdb using skill-team-expert graph structure.

		dblp								imdb							
		%pr		%rec		%ndcg		%map		%pr		%rec		%ndcg		%map	
		@5	@10	@5	@10	@5	@10	@5	@10	@5	@10	@5	@10	@5	@10	@5	@10
e2e	gs	14.37	09.24	54.00	67.79	33.21	38.39	25.18	28.10	15.39	09.16	49.73	57.44	33.70	36.78	27.00	29.03
	gin	4.47	09.07	4.68	66.95	31.50	36.12	22.74	25.36	14.08	08.91	45.34	56.49	28.49	32.79	21.76	24.30
	gine	07.66	07.65	21.56	44.17	16.50	24.74	12.67	16.18	9.33	2.21	3.14	6.25	8.80	3.86	1.13	4.07
	gat	14.33	09.46	53.81	68.86	8.11	43.78	1.44	34.65	14.66	09.17	47.44	57.47	33.45	37.40	27.17	29.68
	gatv2	14.03	9.48	52.73	9.01	37.87	3.98	31.43	4.86	14.49	09.09	46.72	57.18	33.81	37.88	27.86	30.39
	han	**28.97**	**15.17**	**83.19**	**86.38**	**70.42**	**71.74**	**64.28**	**65.27**	**23.37**	**15.42**	**57.82**	**74.15**	**45.58**	**52.38**	**37.42**	**42.21**
t-fnn	mhot	00.70	00.65	01.07	01.96	00.96	01.37	00.53	00.66	00.75	00.70	00.86	01.59	00.84	01.19	00.41	00.52
	d2v	00.83	00.72	01.27	02.21	01.18	01.62	00.72	00.88	00.84	00.76	00.93	01.72	00.95	01.32	00.45	00.57
	m2v	00.82	00.70	01.24	02.13	01.13	01.55	00.66	00.82	00.84	00.77	00.95	01.76	00.95	01.33	00.46	00.57
	lant	00.90	00.83	01.35	02.48	01.17	01.69	00.67	00.84	00.74	00.68	00.81	01.52	00.85	01.18	00.40	00.50
	gs	00.87	00.74	01.30	02.24	01.26	01.63	00.60	00.76	00.83	00.73	00.91	01.64	00.95	01.27	00.44	00.55
	gin	00.95	00.78	01.43	02.35	01.37	01.57	00.59	00.76	00.76	00.68	00.84	01.56	00.87	01.19	00.40	00.51
	gine	01.02	00.83	01.52	02.49	01.35	01.80	00.76	00.92	00.73	00.64	00.82	01.45	00.85	01.14	00.41	00.50
	gat	01.19	00.92	01.78	02.75	01.62	02.08	00.92	01.09	00.92	00.82	01.00	01.83	01.06	01.43	00.50	00.62
	gatv2	01.01	00.83	01.51	02.51	01.41	01.88	00.80	00.96	00.87	00.80	00.96	01.80	00.98	01.37	00.47	00.59
	han	00.93	00.82	01.42	02.49	01.33	01.82	00.80	00.97	00.77	00.72	00.86	01.63	00.85	01.21	00.39	00.50
t-bnn	mhot	00.53	00.49	00.82	01.51	00.73	01.05	00.43	00.54	00.83	00.74	00.95	01.71	00.92	01.28	00.45	00.57
	d2v	00.68	00.61	01.02	01.86	00.95	01.33	00.56	00.69	00.87	00.81	00.99	01.84	00.98	01.38	00.47	00.60
	m2v	00.31	00.32	00.48	00.98	00.42	00.65	00.23	00.30	00.76	00.71	00.83	01.58	00.85	01.20	00.41	00.52
	lant	00.66	00.61	01.01	01.87	00.88	01.28	00.49	00.61	01.01	00.96	01.14	02.18	01.18	01.66	00.58	00.73
	gs	00.71	00.66	01.08	02.00	00.94	01.36	00.52	00.65	00.99	00.90	01.12	02.03	01.12	01.55	00.54	00.67
	gin	00.60	00.52	00.91	01.59	00.84	01.15	00.49	00.60	00.90	00.83	01.04	01.92	01.07	01.47	00.52	00.65
	gine	00.75	00.63	01.12	01.90	01.08	01.44	00.66	00.79	01.01	00.93	01.15	02.14	01.16	01.62	00.58	00.71
	gat	00.66	00.62	00.99	01.86	00.87	01.28	00.49	00.62	00.96	00.86	01.07	01.94	01.10	01.50	00.53	00.66
	gatv2	00.64	00.54	00.96	01.64	00.87	01.19	00.49	00.58	00.93	00.80	01.05	01.84	01.05	01.41	00.51	00.62
	han	00.80	00.71	01.20	02.15	01.09	01.53	00.61	00.76	01.06	00.99	01.19	02.25	01.21	01.70	00.59	00.74

Table 3: Efficacy of approaches using location in dblp.

		%pr		%rec		%ndcg		%map	
		@5	@10	@5	@10	@5	@10	@5	@10
e2e	gs	4.63	09.15	2.91	64.97	35.37	39.90	28.27	30.88
	gin	11.80	08.78	42.44	62.46	26.28	33.65	19.51	23.54
	gine	07.66	07.65	21.56	44.17	16.50	24.74	12.67	16.17
	gat	13.69	**09.39**	50.24	**66.50**	5.57	1.74	9.10	2.62
	gatv2	**15.46**	9.22	**56.15**	5.38	**42.58**	**46.13**	**36.62**	**38.75**
	han	06.77	07.93	13.90	29.40	13.85	20.50	10.15	13.39
t-fnn	mhot	00.70	00.65	01.07	01.96	00.96	01.37	00.53	00.66
	d2v	00.83	00.72	01.27	02.21	01.18	01.62	00.72	00.88
	m2v	00.87	00.66	01.24	02.17	01.13	01.57	00.66	00.84
	lant	00.88	00.77	01.33	02.31	01.20	01.65	00.67	00.82
	gs	00.54	00.54	00.82	01.65	00.71	01.10	00.38	00.50
	gin	00.94	00.80	01.43	02.44	01.28	01.75	00.73	00.88
	gine	00.83	00.73	01.24	02.19	01.09	01.53	00.60	00.75
	gat	00.77	00.70	01.18	02.12	01.04	01.47	00.57	00.71
	gatv2	00.96	00.79	01.45	02.40	01.25	01.69	00.69	00.83
	han	00.92	00.79	01.38	02.36	01.26	01.72	00.72	00.87
t-bnn	mhot	00.53	00.49	00.82	01.51	00.73	01.05	00.43	00.54
	d2v	00.68	00.61	01.02	01.86	00.95	01.33	00.56	00.69
	m2v	00.51	00.55	00.70	01.30	00.58	00.65	00.27	00.40
	lant	00.65	00.60	00.99	01.84	00.87	01.26	00.48	00.60
	gs	00.85	00.73	01.28	02.20	01.14	01.56	00.64	00.78
	gin	00.46	00.42	00.70	01.27	00.62	00.89	00.34	00.42
	gine	00.51	00.47	00.76	01.40	00.68	00.97	00.38	00.46
	gat	00.59	00.53	00.90	01.61	00.81	01.14	00.46	00.57
	gatv2	00.52	00.52	00.77	01.56	00.67	01.04	00.37	00.48
	han	00.62	00.58	00.94	01.76	00.81	01.19	00.44	00.56

and *ii*) skill-expert bipartite graph, consisting of links between skills and expert members of a team but with *no* node type for team. Following the literature [38], we set the location of a team in dblp as the publication venue. In imdb, however, the available location information is the producing country of a movie, which is US in almost all movies, and was therefore omitted from our experiments. From Tables 2 and 3, we observe that the performance of e2e-* models on the expert collaboration graph with location and lack thereof is closely aligned, with marginal differences, except for han, which is no longer the best due to its meta-path-based attention mechanism that overlooked the location links. In contrast, for transfer-based models, we observe little to *no* improvement, which is expected as the vector representations of skills are *in*sufficient to convey such additional information to feedforward neural classifiers. With respect to the skill-expert bipartite graph, of which we refer the reader to the codebase for the results due to the page limit, we observe that *neither* end-to-end *nor* transfer-based models perform on par with their performance in other graph structures due to the *highly dense* bipartite graph structure, wherein all skills and experts are connected to each other without the contextual separation by team nodes, which highlights the importance of graph structure in effective message passing in graph neural networks.

RQ4: Performance leaderboard for graph neural networks. From Table 2 and for the end-to-end approach, it is evident that han outperforms *all* methods across datasets and metrics, with gat and gatv2 being the closest runners-up in dblp. The consistent high performance of han, gat, and gatv2 highlights the dominance of attention-based graph neural networks. Conversely, despite the gin model's theoretically proven efficacy in capturing graph structure [53], its empirical performance typically matches or lags behind the attentive models for its sensitivity to hyperparameters. With respect to the transfer-based approach, not unexpectedly, we observe a consistent efficacy of graph neural networks compared to naive embeddings of skills (mhot). Attentive models (gat, gatv2) generally perform the best, similarly when employed in the end-to-end approach. However, the performance ranking of graph neural networks is *in*consistent across neural classifiers; it is more stable for non-variational models (t-fnn-*) than for variational ones (t-bnn-*) due to the greater sensitivity of variational architectures.

7 Concluding Remarks and Future Work

In this paper, we proposed an end-to-end graph neural network-based approach to improve the efficacy of team recommendations. Our experiments on two large-scale datasets with distinct distributions of teams over skills show state-of-the-art performance, regardless of the underlying graph neural network and the structure of the expert collaboration graph. For future work, we study cold-start scenarios for new experts and emerging skills with limited historical data, as well as team *refinement*, a subproblem in team recommendation, where experts in an existing team are replaced to maintain or enhance performance in dynamic environments.

References

[1] Meltem Aksoy, Seda Yanik, and Mehmet Fatih Amasyali. 2023. Reviewer Assignment Problem: A Systematic Review of the Literature. *J. Artif. Intell. Res.* 76 (2023), 761–827. https://doi.org/10.1613/JAIR.1.14318

[2] Negar Arabzadeh, Sajad Ebrahimi, Sara Salamat, Mahdi Bashari, and Ebrahim Bagheri. 2024. Reviewerly: Modeling the Reviewer Assignment Task as an Information Retrieval Problem. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, Edoardo Serra and Francesca Spezzano (Eds.). ACM, 5554–5555. https://doi.org/10.1145/3627673.3679081

[3] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks?. In *ICLR*. https://openreview.net/forum?id=F72ximsx7C1

[4] Manoel B. Campêlo, Tatiane Fernandes Figueiredo, and Ana Silva. 2020. The sociotechnical teams formation problem: a mathematical optimization approach. *Ann. Oper. Res.* 286, 1 (2020), 201–216. https://doi.org/10.1007/S10479-018-2759-5

[5] Gonzalo Candel, Víctor Sánchez-Anguix, Juan M. Alberola, Vicente Julián, and Vicent J. Botti. 2023. An Integer Linear Programming Model for Team Formation in the Classroom with Constraints. In *Hybrid Artificial Intelligent Systems - 18th International Conference, HAIS 2023, Salamanca, Spain, September 5-7, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 14001)*, Pablo García Bringas, Hilde Pérez García, Francisco Javier Martínez de Pisón, Francisco Martínez-Álvarez, Alicia Troncoso Lora, Álvaro Herrero, José Luís Calvo-Rolle, Héctor Quintián, and Emilio Corchado (Eds.). Springer, 397–408. https://doi.org/10.1007/978-3-031-40725-3_34

[6] Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. 2024. Macro Graph Neural Networks for Online Billion-Scale Recommender Systems. In *WWW*. 3598–3608. https://doi.org/10.1145/3589334.3645517

[7] Arman Dashti, Saeed Samet, and Hossein Fani. 2022. Effective Neural Team Formation via Negative Samples. In *CIKM*. 3908–3912. https://doi.org/10.1145/3511808.3557590

[8] Arman Dashti, Karan Saxena, Dhwani Patel, and Hossein Fani. 2022. OpeNTF: A Benchmark Library for Neural Team Formation. In *CIKM*. 3913–3917. https://doi.org/10.1145/3511808.3557526

[9] DBLP. . https://www.aminer.org/citation

[10] Yuxiao Dong, Nitesh V. Chawla, and A. Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *SIGKDD*. 135–144.

[11] Brendan L Douglas. 2011. The weisfeiler-lehman method and graph isomorphism testing. *arXiv preprint arXiv:1101.5211* (2011).

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

[13] José G. M. Esgario, Iago E. da Silva, and Renato A. Krohling. 2019. Application of Genetic Algorithms to the Multiple Team Formation Problem. *CoRR* abs/1903.03523 (2019). http://arxiv.org/abs/1903.03523

[14] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *WWW*. 417–426. https://doi.org/10.1145/3308558.3313488

[15] Hossein Fani, Reza Barzegar, Arman Dashti, and Mahdis Saeedi. 2024. A Streaming Approach to Neural Team Formation Training. In *ECIR*, Vol. 14608. Springer, 325–340. https://doi.org/10.1007/978-3-031-56027-9_20

[16] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Jure Leskovec. 2021. GNNAutoScale: Scalable and Expressive Graph Neural Networks via Historical Embeddings. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 3294–3304. http://proceedings.mlr.press/v139/fey21a.html

[17] Matthias Fey, Jinu Sunil, Akihiro Nitta, Rishi Puri, Manan Shah, Blaz Stojanovic, Ramona Bendias, Alexandria Barghi, Vid Kocijan, Zecheng Zhang, Xinwei He, Jan Eric Lenssen, and Jure Leskovec. 2025. PyG 2.0: Scalable Learning on Real World Graphs. *CoRR* abs/2507.16991 (2025). https://doi.org/10.48550/ARXIV.2507.16991 arXiv:2507.16991

[18] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. 2023. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *Trans. Recomm. Syst.* 1, 1 (2023), 1–51. https://doi.org/10.1145/3568022

[19] Kiarash Golzadeh, Lukasz Golab, and Jarek Szlichta. 2025. Explaining Expert Search and Team Formation Systems with ExES. In *ICDE*. IEEE, 224–237. https://doi.org/10.1109/ICDE65448.2025.00024

[20] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems*. 1024–1034.

[21] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *ICLR 2020,*. https://openreview.net/forum?id=HJlWWJSFDH

[22] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2025. GRAG: Graph Retrieval-Augmented Generation. In *NAACL*. 4145–4157.

https://aclanthology.org/2025.findings-naacl.232/

[23] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware Coupled Graph Neural Network for Social Recommendation. In *AAAI*. 4115–4122. https://doi.org/10.1609/AAAI.V35I5.16533

[24] IMDb. . https://developer.imdb.com/non-commercial-datasets/

[25] Sunny Joseph Kalayathankal, John T Abraham, and Joseph Varghese Kureethara. 2019. A Fuzzy Approach To Project Team Selection. *International Journal of Scientific Technology Research* 8 (2019).

[26] Mehdi Kargar and Aijun An. 2011. Discovering top-k teams of experts with/without a leader in social networks. In *CIKM*. ACM, 985–994.

[27] Mehdi Kargar, Lukasz Golab, Divesh Srivastava, Jaroslaw Szlichta, and Morteza Zihayat. 2022. Effective Keyword Search Over Weighted Graphs. *IEEE Trans. Knowl. Data Eng.* 34, 2 (2022), 601–616. https://doi.org/10.1109/TKDE.2020.2985376

[28] Sagar Kaw, Ziad Kobti, and Kalyani Selvarajah. 2023. Transfer Learning with Graph Attention Networks for Team Recommendation. In *International Joint Conference on Neural Networks, IJCNN*. 1–8. https://doi.org/10.1109/IJCNN54540.2023.10191717

[29] Bosung Kim, Hyewon Choi, Haeun Yu, and Youngjoong Ko. 2021. Query Reformulation for Descriptive Queries of Jargon Words Using a Knowledge Graph based on a Dictionary. In *CIKM*. 854–862. https://doi.org/10.1145/3459637.3482382

[30] Yue Kou, Derong Shen, Quinn Snell, Dong Li, Tiezheng Nie, Ge Yu, and Shuai Ma. 2020. Efficient Team Formation in Social Networks based on Constrained Pattern Graph. In *ICDE*. 889–900. https://doi.org/10.1109/ICDE48307.2020.00082

[31] Christin Katharina Kreutz and Ralf Schenkel. 2023. RevASIDE: Evaluation of Assignments of Suitable Reviewer Sets for Publications from Fixed Candidate Pools. *J. Data Intell.* 4, 1&2 (2023), 101–133. https://doi.org/10.26421/JDI4.1-2-1

[32] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[33] Quoc V. Le and Tomás Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*, Vol. 32. 1188–1196. http://proceedings.mlr.press/v32/le14.html

[34] Zijian Li, Qingyan Guo, Jiawei Shao, Lei Song, Jiang Bian, Jun Zhang, and Rui Wang. 2025. Graph Neural Network Enhanced Retrieval for Question Answering of Large Language Models. In *NAACL*. 6612–6633. https://aclanthology.org/2025.naacl-long.337/

[35] Poppy L McLeod, Quinn W Cunningham, Deborah DiazGranados, Gabi Dodoiu, Seth Kaplan, Joann Keyton, Nicole Larson, Chelsea LeNoble, Stephan U Marsch, Thomas A O'Neill, et al. 2021. Hacking teamwork in health care: Addressing adverse effects of ad hoc team composition in critical care medicine. *Health care management review* 46, 4 (2021), 341–348.

[36] Larry K Michaelsen, Arletta Bauman Knight, and L Dee Fink. 2023. *Team-based learning: A transformative use of small groups in college teaching*. Taylor & Francis.

[37] Hoang Nguyen, Radin Hamidi Rad, Feras N. Al-Obeidat, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, Jaroslaw Szlichta, and Fattane Zarrinkalam. 2023. Learning heterogeneous subgraph representations for team discovery. *Inf. Retr. J.* 26, 1 (2023), 8. https://doi.org/10.1007/S10791-023-09421-6

[38] Radin Hamidi Rad, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2021. Retrieving Skill-Based Teams from Collaboration Networks. In *SIGIR*. ACM, 2015–2019. https://doi.org/10.1145/3404835.3463105

[39] Radin Hamidi Rad, Hossein Fani, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2024. A Variational Neural Architecture for Skill-based Team Formation. *ACM Trans. Inf. Syst.* 42, 1 (2024), 7:1–7:28. https://doi.org/10.1145/3589762

[40] Radin Hamidi Rad, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2020. Learning to Form Skill-based Teams of Experts. In *CIKM*. 2049–2052. https://doi.org/10.1145/3340531.3412140

[41] Radin Hamidi Rad, Aabid Mitha, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2021. PyTFL: A Python-based Neural Team Formation Toolkit. In *CIKM*. 4716–4720. https://doi.org/10.1145/3459637.3481992

[42] Pisol Ruenin, Morakot Choetkiertikul, Akara Supratak, and Suppawong Tuarob. 2024. TeReKG: A temporal collaborative knowledge graph framework for software team recommendation. *Knowl. Based Syst.* 289 (2024), 111492. https://doi.org/10.1016/J.KNOSYS.2024.111492

[43] Kalyani Selvarajah, Pooya Moradian Zadeh, Ziad Kobti, Mehdi Kargar, Mohd Tazim Ishraque, and Kathryn Pfaff. 2018. Team Formation in Community-Based Palliative Care. In *2018 Innovations in Intelligent Systems and Applications, INISTA 2018, Thessaloniki, Greece, July 3-5, 2018*. IEEE, 1–7. https://doi.org/10.1109/INISTA.2018.8466272

[44] Mauro Sozio and Aristides Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *SIGKDD*. ACM, 939–948. https://doi.org/10.1145/1835804.1835923

[45] Damjan Strnad and Nikola Guid. 2010. A fuzzy-genetic decision support system for project team formation. *Appl. Soft Comput.* 10, 4 (2010), 1178–1187. https://doi.org/10.1016/j.asoc.2009.08.032

[46] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[47] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm. 2019. Deep Graph Infomax. In *ICLR*. https://openreview.net/forum?id=rklz9iAcKQ

[48] Lin Wang, Yifeng Zeng, Bilian Chen, Yinghui Pan, and Langcai Cao. 2020. Team Recommendation Using Order-Based Fuzzy Integral and NSGA-II in StarCraft. *IEEE Access* 8 (2020), 59559–59570. https://doi.org/10.1109/ACCESS.2020.2982647

[49] Menghan Wang, Yujie Lin, Guli Lin, Keping Yang, and Xiao-Ming Wu. 2020. M2GRL: A Multi-task Multi-view Graph Representation Learning Framework for Web-scale Recommender Systems. In *SIGKDD*. 2349–2358. https://doi.org/10.1145/3394486.3403284

[50] Ruijie Wang, Yutong Zhang, Jinyang Li, Shengzhong Liu, Dachun Sun, Tianchen Wang, Tianshi Wang, Yizhuo Chen, Denizhan Kara, and Tarek F. Abdelzaher. 2024. MetaHKG: Meta Hyperbolic Learning for Few-shot Temporal Reasoning. In *SIGIR*. 59–69. https://doi.org/10.1145/3626772.3657711

[51] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*. ACM, 2022–2032.

[52] Hyeongon Wi, Seungjin Oh, Jungtae Mun, and Mooyoung Jung. 2009. A team formation model based on knowledge and collaboration. *Expert Syst. Appl.* 36, 5 (2009), 9121–9134. https://doi.org/10.1016/j.eswa.2008.12.031

[53] Keyulu Xu, W. Hu, Jure Leskovec, and S. Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*. https://openreview.net/forum?id=ryGs6iA5Km

[54] Armen Zakarian and Andrew Kusiak. 1999. Forming teams: An analytical approach. *IIE Transactions* 31 (01 1999), 85–97. https://doi.org/10.1023/A:1007580823003

[55] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*. 29476–29490. https://proceedings.neurips.cc/paper/2021/hash/f6a673f09493afcd8b129a0bcf1cd5bc-Abstract.html