OpeNTF2: Fairness-aware Graph Neural Team Formation

Hamed Loghmani
University of Windsor, Canada
ghasrlo@uwindsor.ca

Md Jamil Ahmed
University of Windsor, Canada
ahmed491@uwindsor.ca

Kap Thang
University of Windsor, Canada
thangk@uwindsor.ca

Gabriel Rueda
University of Windsor, Canada
ruedag@uwindsor.ca

Hossein Fani
University of Toronto, Canada
hossein.fani@utoronto.ca

Abstract

We contribute `OpeNTF2`, a unified framework for neural team formation, which achieves state-of-the-art efficacy while improving efficiency in recommending teams of experts who are *almost surely* successful at completing complex tasks. `OpeNTF2` extends its initial release in four prime directions: it (1) integrates fairness-aware debiasing rerankers to mitigate popularity and gender disparities; (2) models team formation as link prediction on expert–skill collaboration graphs to capture multi-hop cross-team relational dependencies via graph neural networks; (3) formulates team formation as a sequence prediction task to leverage encoder-decoder recurrent models and transformers; and (4) introduces a temporal training strategy to account for the evolution of experts' skills and collaboration ties. While prior libraries support reproducible pipelines, none incorporate fairness, sequential and graph-based modeling, and temporal dynamics within a single framework. The codebase, along with the installation and execution instructions as well as case studies of extension components, can be obtained under `cc-by-nc-sa-4.0` license at: `https://github.com/fani-lab/OpeNTF/tree/v0.2.0.0`.

CCS Concepts

• **Information systems** → **Recommender systems**; • **Human-centered computing** → **Social recommendation**; • **Computing methodologies** → **Neural networks**.

Keywords

Neural Team Recommendation; Social Information Retrieval;

ACM Reference Format:

Hamed Loghmani, Md Jamil Ahmed, Kap Thang, Gabriel Rueda, and Hossein Fani. 2026. OpeNTF2: Fairness-aware Graph Neural Team Formation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, Australia.* ACM, New York, NY, USA, 9 pages.

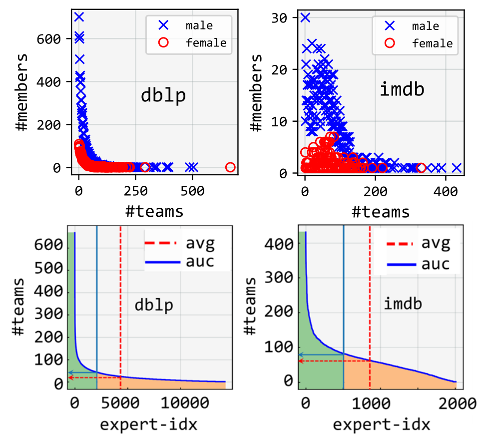

Figure 1: Gender disparity (top) and long-tail distribution of nonpopular experts (bottom) in `dblp` and `imdb` [35, 36].

1 Introduction

Team formation aims to recommend experts whose combined skills, applied in coordinated teamwork, successfully solve difficult tasks, and can be seen as social information retrieval (Social IR) where the right group of experts are searched for and hired to solve the task at hand. Examples are forming a team of cast and crew to produce a *blockbuster* movie in the genres of '*action*' and '*comedy*,' or a team of researchers in the fields of '*biomedical*' and '*machine learning*' whose findings are yet to be *published* in a reputable journal. Forming teams is challenging for the large pool of candidate experts from diverse backgrounds and personality traits, along with the *un*known synergistic balance among them; *not* all teams with the best experts are successful [22, 60]. To address the complex nature of team formation, machine learning has been employed, among which neural models build up the following in scalability and recommendation efficacy [6, 38, 39, 42, 49]. To support neural team formation research with a reproducible and open-source platform, Rad et al. released `PyTFL` [42], a python-based library that uses a variational multi-label neural classifier to predict the optimal subset of experts for a team. `PyTFL`, however, struggles with large-scale datasets and lacks modularity for ease of customization and extension to new models and training datasets. To fill the gap, we previously open-sourced `OpeNTF` [7], a modular and scalable benchmark framework, which includes reference neural models along with negative sampling heuristics [6] and three large-scale datasets from varying domains. Nonetheless, the focus of these and existing libraries has been the maximization of the models'

Table 1: Comparison of existing libraries for neural team formation.

	neural models	datasets	training strategies	fairness-aware	notions of fairness
`PyTFL` [42]	variational	`dblp`	shuffled	×	×
`Adila` [35]	×	×	×	greedy deterministic [16]	demographic parity [11]
`OpeNTF` [7]	feedforward, variational	`dblp`, `imdb`, `uspt`	shuffled, negative sampling	×	×
`OpeNTF2`	feedforward, variational +**gnn (transfer)** +**gnn (end-to-end)** +**seq-to-seq** +**transformers**	`dblp`, `imdb`, `uspt` +**`github`**	shuffled, negative sampling +**temporal (streaming)**	greedy deterministic [16] +**probabilistic reranking** [59]	demographic parity [11] +**equal opportunity** [21]

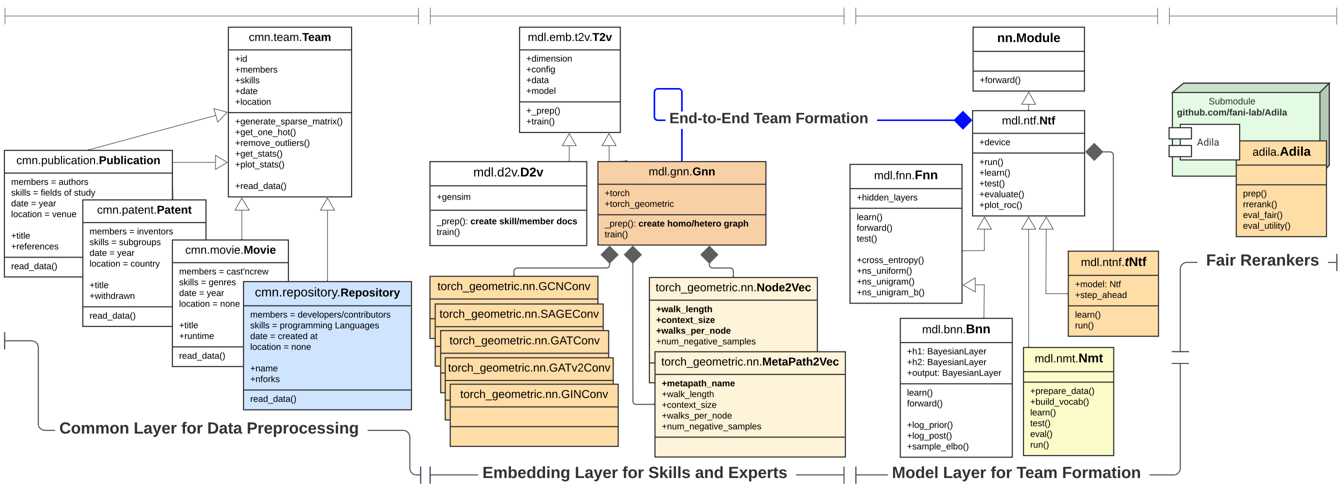

Figure 2: `OpeNTF2` partial class diagram. Coloured (gray) components are extensions to our previous release [7].

efficacy, largely ignoring fairness, especially when training datasets withhold biases. From Figure 1 (top), datasets in team formation are segmented toward male experts, and females are heavily underrepresented. From Figure 1 (bottom), such datasets also suffer from popularity bias, where the majority of *non*popular experts have scarcely participated in teams, whereas few popular experts dominate. Therefore, popular or male experts would receive more attention and are more frequently recommended by neural models, leading to systematic discrimination against already disadvantaged nonpopular or female experts. To the best of our knowledge, there is no fairness-aware library in neural team formation except that of `Adila` [35], which has adapted deterministic greedy reranking algorithms [16] to mitigate popularity bias. However, `Adila` is *not* an end-to-end pipeline; it only operates on the recommendation outputs of other libraries. Moreover, existing libraries frame the team formation problem as a *multi-label* Boolean classification based on the mutually independent selection of experts, overlooking both team-level joint dependencies and multi-hop cross-team relational structure. They also learn vector representations of experts and skills in a *static* latent space, failing to capture their temporal dynamics due to societal demands, novel technologies, and working experiences.

Contributions. As summarized in Table 1 and from Figure 2, we bring forth `OpeNTF2` that extends the prior version and contrasts with existing libraries in four major directions:

(1) Fair Team Formation [16, 35, 59]. To counter unfair biases in the recommended experts for teams by neural models, we adopt probabilistic [59] as well as deterministic [16] debiasing reranking methods based on two well-known alternative notions of fairness: *i*) demographic parity [11] and *ii*) equal opportunity [21], which enables further post-processing refinements to the list of recommended experts to ensure the desired fair outcome in terms of popularity or gender.

(2) Graph Neural Team Formation [1, 29, 39]. To capture multi-hop relational and structural information encoded in the expert collaboration network, we integrate graph neural networks in two ways: *i*) prior pretraining and transfer of dense vector representations of skills to accommodate a large set of skills and reduce the complexity of the neural classifiers' *input* layer, following existing work [29, 38, 39]; and *ii*) *end-to-end* team formation via link prediction between required skills and optimal experts for a team, performed directly on the expert collaboration graph, following the efficacy of end-to-end graph neural networks in various recommendation and information retrieval tasks [4, 14, 23, 24, 30, 34, 52, 53, 64].

(3) Translative Team Formation [49]. To accommodate the large number of experts for neural classifiers in the *output* layer, we reformulate the team formation problem into a translation task from a sequence of required skills to a sequence of optimum experts and incorporate transformers and encoder-decoder recurrent models with attention for team formation, following their efficacy not only in natural language tasks, but also in recommendation systems [8, 27, 47].

(4) Temporal Training Strategy [12]. Inspired by curriculum learning [54] and temporal latent space inference [62, 63], we develop a temporal training strategy that neural models can seamlessly utilize during training to learn temporal aspects of experts' skill sets, levels of expertise, and collaboration ties through the succession of time intervals and to boost the efficacy of inference [12].

Enhancements. To further enrich the testbed, we add new evaluation metrics, including `skill coverage` [40, 51] to capture the coverage of required skills in the recommended team, and a new large-scale `github` dataset, where repositories correspond to teams, programmers to experts, and programming languages to skills. Contrary to existing datasets, `github` has a limited variety of skills, opening new challenges for the generalizability of team formation models in varying domains with distinct distributions. To enhance the architecture, we transition to component-based design in `OpeNTF2`, essential for integrating a variety of models and training strategies in a unified framework; components are loosely coupled and reusable, dependencies are installed on demand at the component level for execution efficiency and prevent package conflicts, and configurations are managed by `hydra` [56], enabling systematic and extensive ablation studies. `OpeNTF2` further incorporates continuous integration and testing along with containerized execution for consistent and reproducible behavior across environments[1].

2 Neural Team Formation

Given a set of skills $\mathcal{S}$ and a set of experts $\mathcal{E}$, a team of experts $\mathbf{e} \subseteq \mathcal{E}$ that collectively cover the skill set $\mathbf{s} \subseteq \mathcal{S}$ is shown by $(\mathbf{s}, \mathbf{e})$ along with its success status $y \in \{0, 1\}$, and $\mathcal{T} = \{(\mathbf{s}, \mathbf{e})_y : y \in \{0, 1\}\}$ indexes all teams. Examples of teams include `dblp` research papers, where authors serve as the experts and fields of study (keywords) act as the associated skills. Similarly, `imdb` movies can be seen as teams comprising cast and crew members as the experts, while genres and subgenres serve as the skills. The choice of movies in team formation literature is *not* to be confused with its use cases in item recommender systems or review analysis; herein, the goal is to form a team of cast and crew for a *movie production*, rather than to recommend movies.

Given a subset of required skills $\mathbf{s}$ for an *unseen* team, the team formation problem aims to recommend an optimal subset of experts $\mathbf{e}$ such that their collaboration is *almost surely* successful, that is $(\mathbf{s}, \mathbf{e})_{y=1}$, while avoiding subsets of experts $\mathbf{e}'$ that would result in $(\mathbf{s}, \mathbf{e}')_{y=0}$. To this end, neural team formation models learn a mapping function $f_{\boldsymbol{\theta}} : \mathcal{S} \to \mathcal{E}$, parameterized by $\boldsymbol{\theta}$, from $\mathcal{T}$ using a neural network by maximizing the posterior probability of $\boldsymbol{\theta}$ such that $f_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{e}$, that is,

$$\texttt{argmax}_{\boldsymbol{\theta}}\ p(\boldsymbol{\theta}|\mathcal{T}) \propto p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{(\mathbf{s},\mathbf{e})_{y=1} \in \mathcal{T}} p(\mathbf{e}|\mathbf{s}, \boldsymbol{\theta}) \quad (1)$$

where $p(\mathcal{T}|\boldsymbol{\theta})$ is the likelihood and $p(\boldsymbol{\theta})$ is the prior joint probability of parameters, which is unknown but can be estimated as Gaussian distributions by a variational Bayesian neural network (`Bnn`) [39–41]. Alternatively, we can assume a *uniform* probability distribution over all possible real-values

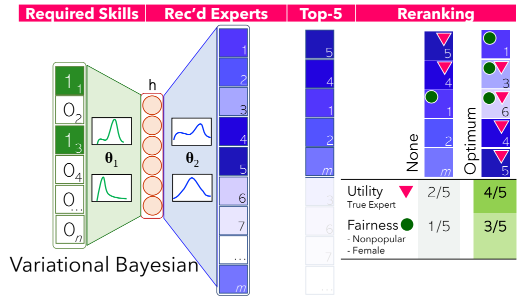

Figure 3: Post-hoc debiasing reranking in `OpeNTF2`. Although illustrated for a variational feedforward model, the reranking is model-agnostic and applicable to graph-based, translative, and temporal models [35, 36].

of $\boldsymbol{\theta}$ and only estimate the likelihood $p(\mathcal{T}|\boldsymbol{\theta})$ via maximum likelihood optimization by a *non*-variational feedforward neural network (`Fnn`), discarding prior uncertainty $p(\boldsymbol{\theta})$ at the cost of overconfident point estimates of $\boldsymbol{\theta}$ [18].

3 `OpeNTF2` Extensions

Herein, we lay out the details of `OpeNTF2`'s key extensions, along with a brief overview of benchmark results.

3.1 Fair Team Formation

From Figure 3, a trained neural model, whether feedforward [6, 41], graph-based [29, 39], translative [49], or temporal [12], recommends *all* available experts but with different probabilities after a `sigmoid` layer, and the final recommended team includes experts with top-k highest probabilities. This enables further post-hoc *re*ranking of the list of recommended experts to ensure the desired fair and unbiased distribution of experts. Here, we provide a brief and high-level overview of the principal components; for theoretical depth and details, we refer to our fairness-related papers [35, 36], also linked in the codebase. For sample runs following best practices, see our colab script at the codebase.[2]

3.1.1 ***Debiasing Rerankings***. `OpeNTF2` integrated debiasing rerankers including three deterministic [16] and the state-of-the-art probabilistic (`fa-ir`) [59] ones. For instance, in `fa-ir`, the top-k ranked list of a model's prediction is assumed to follow k independent Bernoulli trials of win/lose at achieving the desired fair team. Given p as the desired proportion of disadvantaged experts (nonpopular or female) in a fair team and a significance level α, at each position i, we check whether top-$i \in \{1, \cdots, k\}$ ranked list statistically significantly follows a Bernoulli distribution with winning probability p. Otherwise, based on a *notion* of fairness, we select a disadvantaged expert from the list after i and insert her at the ith position.

3.1.2 ***Notions of Fairness***. To design and evaluate fairness-aware models, fairness has been formalized based on well-known *notions* of justice and equity at a group level, like

[1] https://github.com/fani-lab/OpeNTF/tree/main/.github/workflows

[2] https://github.com/fani-lab/OpeNTF/blob/main/ipynb/fair.ipynb

female vs. male or nonpopular vs. popular experts. `OpeNTF2` has integrated two well-known notions:

Demographic Parity [11], which requires the top-k recommended experts to reflect p as the desired proportion of disadvantaged experts, *irrespective* of an expert having the required skills. For instance, given $p = 0.5$ as the desired proportion of females, a debiasing reranker should preserve the `1:1` balance in the top-k ranked list of recommended experts between female and male experts. This notion of fairness forego qualifications and has limitations [11, 21];

Equal Opportunity [21], which enforces the desired proportion of disadvantaged experts only among the *qualified* experts. For example, given '*machine learning*' as the required skill and $p = 0.5$, a debiasing reranker should select *only* among experts who have '*machine learning*' in their skills while preserving the `1:1` balance between female and male experts in the top-k ranked experts.

3.1.3 ***Fairness Metrics***. `OpeNTF2` integrated fairness evaluation metrics, including `skew` and normalized discounted cumulative KL-divergence (`ndkl`) [16]. While `skew` calculates the logarithmic ratio of the proportion of disadvantaged experts in top-k list of experts to the desired proportion, `ndkl` measures the divergence of the distributions using Kullback–Leibler [33]. In both metrics, the lower, the better, with being `0` in the ideal equal distributions. In tandem with accuracy metrics such as `precision`, `recall`, `ndcg`, and `map` at top-k, which measure the efficacy of the model at recommending *correct* experts for the teams in a test set, `OpeNTF2` allows exploring the trade-offs between notions of fairness and the model's accuracy for a team formation model.

3.1.4 ***Gender vs. Popularity Labels***. `OpeNTF2` relies on experts' labels of popularity or gender to measure the biases before and after running the debiasing rerankers. Since gender is self-identified, gender labels are either available, like in `uspt` dataset, or have been inferred, like in `imdb`, based on the roles identified as actors or actresses. For the popularity status, an expert has been *objectively* labelled based on the number of teams the expert has participated in, referred to as *sociometric* popularity, which is widely adopted in recommender system literature [32]. As shown earlier in Figure 1 (bottom), we adopt two alternatives in `OpeNTF2`: *i*) `avg`: an expert is popular if the expert participated in more than the average number of teams per expert over the entire dataset, and nonpopular otherwise, or *ii*) `auc`: an expert is popular if she belongs to the *short head* in the 2-d curve of the distribution of experts in teams, and nonpopular otherwise. We split the curve into a short head and a long tail based on equal area under the curve. We acknowledge that an expert's participation in many teams, e.g., movies in `imdb`, does not necessarily reflect popularity from the people's perspectives (i.e., being liked), but repetition of the expert in many training samples of teams from the neural model's perspective does and leads to the model's popularity bias.

3.1.5 ***Case Studies of Fairness***. `OpeNTF2`'s case studies of debiasing algorithms on `imdb` to mitigate popularity bias based on the two notions of fairness have partially shown in Table 2. As seen, deterministic rerankers (`det-*`) [16] reduce the divergence with the desired distribution and mitigate popularity bias, as shown by lower `ndkl`, but at the cost of negative impacts on models' accuracy. Similar trends can be observed in `dblp` dataset [35]. However, the probabilistic reranker (`fa-ir`) [59] could mitigate the bias while maintaining the models' accuracy. For the complete comparative results, we refer the reader to our fairness-related papers [35, 36].

Table 2: Results of debiasing rerankers on the Bayesian (variational) neural classifier (`Bnn`), with skill embeddings (`-emb`) and lack thereof, for mitigating popularity bias on `imdb`.

			%ndkl10↓		%map10↑	%ncdg10↑
	neural model	reranker	before	after	Δ	Δ
demographic parity	Bnn [40]	det-cons [16]	67.53	**16.59**	-0.36	-0.82
		det-greedy [16]		**16.60**	-0.36	-0.82
		det-relaxed [16]		**16.35**	-0.36	-0.82
		fa-ir [59]		**17.27**	**0.00**	**0.00**
	Bnn-emb [40]	det-cons [16]	74.67	**15.71**	-0.48	-1.03
		det-greedy [16]		**15.72**	-0.48	-1.03
		det-relaxed [16]		**15.43**	-0.48	-1.03
		fa-ir [59]		**17.53**	**0.00**	**0.00**
equal opportunity	Bnn [40]	det-cons [16]	61.74	**19.85**	-0.35	-0.81
		det-greedy [16]		**20.11**	-0.35	-0.81
		det-relaxed [16]		**19.70**	-0.35	-0.81
		fa-ir [59]		**16.61**	**0.00**	**0.00**
	Bnn-emb [40]	det-cons [16]	68.57	**18.94**	-0.48	-1.03
		det-greedy [16]		**19.21**	-0.48	-1.03
		det-relaxed [16]		**18.77**	-0.48	-1.03
		fa-ir [59]		**16.88**	**0.00**	**0.00**

3.2 Graph Neural Team Formation

The use of experts' collaboration graphs was mainstream in traditional team formation approaches rooted in subgraph optimization [17, 28, 46]. However, recent neural models have increasingly leveraged them for pretraining and *transfer* learning of dense vector representations of skills, enabling scalability to a large set of skills and reducing the complexity of neural classifiers in their input layers [29, 38, 39], as illustrated in Figure 4 (left). Such approaches entail two key shortcomings: *i*) the skill embeddings are learned disjointly in a self-supervised manner during a pretraining phase, oblivious to the team recommendation learning process, and *ii*) they suffer from the curse of sparsity in the output layer of the neural classifiers. To fill the gaps, from Figure 4 (right), we reformulate the team recommendation problem into *end-to-end* link predictions between nodes of required skills and optimal experts for a team node directly within the expert collaboration graph. The end-to-end formulation directly taps into multi-hop relational and structural information in the graph for complex cross-team links among experts and their associated skills, as shown in Figure 5.

3.2.1 ***Expert Collaboration Graph***. As shown in Figure 4, given a successful team $(\mathbf{s}, \mathbf{e})_{y=1} \in \mathcal{T}$, we map it into a connected *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 team node but not to each other. After incorporating all successful teams of a training set in a heterogeneous graph $\mathrm{G} = \langle \mathcal{N}, \mathcal{L} \rangle$, whose nodes are skills $n_{\mathcal{S}}$, successful teams $n_{\mathcal{T}}$ and experts $n_{\mathcal{E}}$, and links

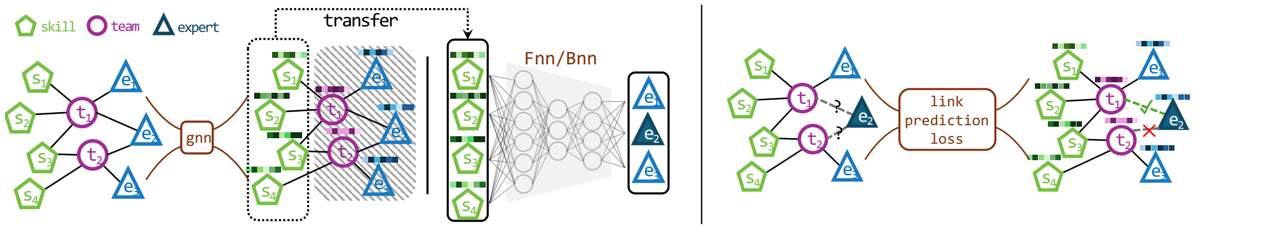

Figure 4: Transfer-based graph neural network [1] (left) versus end-to-end approach [2] (right).

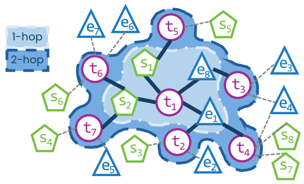

Figure 5: Multi-hop relational and structural information in the expert collaboration graph.

are $\mathcal{L} = \{n_{\mathcal{S}} \times n_{\mathcal{T}}\} \cup \{n_{\mathcal{T}} \times n_{\mathcal{E}}\}$, we then train a graph neural network g, parameterized by $\boldsymbol{\phi}$, $g_{\boldsymbol{\phi}} : \mathcal{N} \to \mathbb{R}^d$, via link prediction loss to learn dense vectors of skills and experts using positive (existing) and negative (non-existing) link sampling strategies. To study the impact of graph structure, `OpeNTF2` currently has a variety of graph structures, including `skill-member` bipartite, `skill-team-member` tripartite, and `skill-team-member-`**`location`** when the geolocation information for the teams is available, like in `dblp`.

3.2.2 ***Transfer vs. End-to-End Learning***. Via `PyG`[3] [13], we now have integrated a diverse set of graph neural networks for g_ϕ, including random-walk-, metapath-, message-passing-, and attention-based models, such as node2vec (`n2v`) [19], metapath2vec (`m2v`) [10], graph convolutional network (`gcn`) [58], graphsage (`gs`) [20], graph attention network (`gat`) [3], and graph isomorphic network (`gin`) [55]. In the transfer-based setting, as shown in Figure 4 (left), `OpeNTF2` operates in the embedding mode, where a graph neural network is trained to learn skill embeddings while disregarding embeddings of other node types. These skill embeddings are then fed into an underlying multi-label classifier, such as a non-variational feedforward neural network (`Fnn`) or a variational Bayesian neural network (`Bnn`). During inference, given a subset of skills $\mathbf{s}$ required to form an unseen test team, the corresponding skill embeddings are summed and fed into the neural classifier per Equation 1, i.e., $f_{\boldsymbol{\theta}}(\mathbf{s}) \simeq f_{\boldsymbol{\theta}}(v_{\mathbf{s}} = \sum_{s_i \in \mathbf{s}} g_\phi(n_{s_i}))$, and the experts with top-k highest probabilities are selected as the recommended team of size k. In contrast, in the end-to-end scenario, a graph neural network is trained to directly predict `member-team` links, using them as supervision signals. During inference, as shown in Figure 4 (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 $\mathcal{E}$ and the team's node t as cosine similarity between their embeddings. Formally, $\forall e_j \in \mathcal{E}; f_{\boldsymbol{\theta}}(\mathbf{s}) \simeq f(\mathbf{s} : \mathrm{G}, g_{\boldsymbol{\phi}}) = g_{\boldsymbol{\phi}}(n_{e_j})^\top g_{\boldsymbol{\phi}}(n_t)$ in $\mathrm{G}\langle\mathcal{N}, \mathcal{L}\rangle$.

[3] https://pyg.org/

Table 3: Efficacy of end-to-end (`e2e`) vs. transfer (`t-*`) approaches in `dblp` and `imdb` using `skill-team-expert` graph.

		dblp				imdb			
		%ndcg5	%ndcg10	%map5	%map10	%ndcg5	%ndcg10	%map5	%map10
e2e	gs	33.21	38.39	25.18	28.10	**33.70**	36.78	27.00	29.03
	gin	31.50	36.12	22.74	25.36	28.49	32.79	21.76	24.30
	gat	**38.11**	**43.78**	**31.44**	**34.65**	33.45	**37.40**	**27.17**	**29.68**
t-Fnn	gs	01.26	01.63	00.60	00.76	00.95	01.27	00.44	00.55
	gin	01.37	01.57	00.59	00.76	00.87	01.19	00.40	00.51
	gat	01.62	02.08	00.92	01.09	01.06	01.43	00.50	00.62
t-Bnn	gs	00.94	01.36	00.52	00.65	01.12	01.55	00.54	00.67
	gin	00.84	01.15	00.49	00.60	01.07	01.47	00.52	00.65
	gat	00.87	01.28	00.49	00.62	01.10	01.50	00.53	00.66

As seen, the end-to-end approach has a single set of parameters $\boldsymbol{\phi}$, eliminating $\boldsymbol{\theta}$ for the multi-label classifier, and hence, excels in computational efficiency and resource usage. For sample runs in *transfer* or *end-to-end* settings, see our colab script at the codebase.[4]

3.2.3 ***Case Studies of Graph-based Learning***. From Table 3, we observe that the end-to-end approach (`e2e`) consistently and substantially outperforms *all* transfer-based methods using either non-variational (`t-Fnn`) or variational Bayesian (`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 graph and supervised information about the optimal subset of experts within teams. For the full comparative results, we refer to our paper currently under review [2].

3.3 Translative Team Formation

From Figure 6, the team formation problem can be viewed as a translation task, mapping a dynamic-length input sequence of required skills onto a dynamic-length output sequence of experts while leveraging the autoregression and global attention mechanisms, which capture dependencies beyond independent expert probabilities in multi-label classification. In `OpeNTF2`, we integrated `OpenNMT-py`[5] [31] to utilize modern transformers and encoder-decoder recurrent models with attention mechanisms [5, 25, 26, 37, 48]. Formally, we estimate $f_{\boldsymbol{\theta}}(\mathbf{s})$ in Equation 1 on a parallel dataset whose pairs of sequences are the ordered list of required skills $\mathbf{s} = [s_1, \ldots, s_n]$ into an optimum ordered list of experts $\mathbf{e} = [e_1, \ldots, e_m]$ for each successful team $(\mathbf{s}, \mathbf{e})_{y=1} \in \mathcal{T}$. An encoder maps the sequence of skills $[s_1, \ldots, s_n]$ onto $\mathbf{h}_n$ and a decoder generates the sequence of experts $[e_1, \ldots, e_m]$ from the $\mathbf{h}_n$, one expert at a time, decomposing the conditional probability $p(\mathbf{e}|\mathbf{s}, \boldsymbol{\theta})$ as $\prod_{k=1}^{m} p(e_k|e_{<k}, \mathbf{s}, \boldsymbol{\theta})$ and seeking the maximum

[4] https://github.com/fani-lab/OpeNTF/blob/main/ipynb/gnn.ipynb
[5] https://github.com/OpenNMT/OpenNMT-py

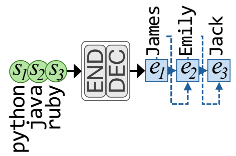

Figure 6: Translative team formation [49].

Table 4: Efficacy of translative vs. multi-label models in dblp.

	k	transformer [50]	convs2s [15]	rnn-att [61]	Bnn	Bnn-emb
%ndcg	2	**10.3611**	2.4770	<u>3.5822</u>	0.0538	0.1083
	5	**10.4597**	2.4276	<u>3.5184</u>	0.0806	0.1555
	10	**10.4824**	2.4487	<u>3.5391</u>	0.1330	0.2397
%map	2	**5.9463**	1.3554	<u>1.9412</u>	0.0242	0.0474
	5	**9.2909**	2.0008	<u>2.8791</u>	0.0411	0.0792
	10	**9.3210**	2.0127	<u>2.8930</u>	0.0558	0.1033

probability among subsets of experts as an optimum team for $\mathbf{s}$, i.e., $f_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{e}$. The probability of generating an expert at the decoder can be conditioned not only on $\mathbf{h}_n$ but also on all $\mathbf{h}_{<n}$ at the encoder, enabling the decoder to *attend* to all skills in the input sequence selectively [37]. To reduce the computational complexity at the encoder and the decoder, a model may have *no* recurrent connections, like in transformers [50], enabling parallel calculation of $\mathbf{h}_{<n}$ at the encoder and $\mathbf{h}_{\geq n}$ at the decoder, an architecture that yielded promising performance on machine translation and led to extensive research on sequence modelling [9, 44, 57]. For sample runs in encoder-decoder settings, see our colab script at the codebase.[6]

3.3.1 ***Case Studies of Translative Learning.*** From Table 4, translative models yield better performance vs. multi-label neural classifiers (Bnn-∗) across metrics in dblp dataset, which can be attributed to conditioning of expert recommendation on the previously recommended experts in the output sequence, addressing the inherent sparse activations in the output layer of neural classifiers. For the complete comparative results, we refer the reader to our paper [49].

3.4 Temporal Training Strategy

As time elapses, skill proficiency or interests of experts evolve due to the shift in societal needs, emerging technologies, and working experiences. Therefore, time as an aspect should be considered in models to incorporate the possible drift and variability of experts' skills and collaboration ties. However, prior team formation methods learn experts and skills vector representation in a *static* latent space [7, 39, 43, 45], overlooking the temporality of experts' skills in time and its impact on the recommendation of future teams.

Inspired by curriculum learning [54] and temporal latent space inference [62], we developed a temporal training strategy [12] to consume time information as an aspect of experts' skills and collaboration ties in teams, as opposed to an extra numeric input feature, while being model-agnostic and with *no* modification to the models' architecture[7]. In contrast to

[6] https://github.com/fani-lab/OpeNTF/blob/main/ipynb/nmt.ipynb
[7] The temporal training strategy is compatible with variational (Bnn) and non-variational neural classifiers (Fnn). It does not apply to seq-to-seq models and transformers. Support for temporal graph neural networks is in the future release.

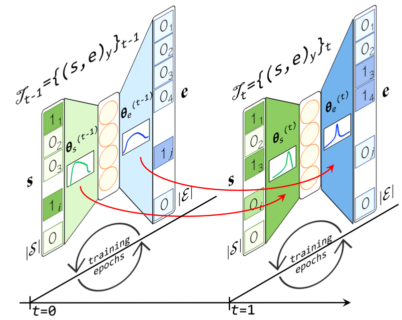

Figure 7: Temporal training strategy [12].

Table 5: Performance improvement of temporal training.

	neural model	%pr2	%pr5	%pr10	%ndcg2	%ndcg5	%ndcg10	%aucroc
dblp	Bnn	0.0570	0.0663	0.0710	0.0538	0.0806	0.1330	63.52
	Bnn-emb [40]	0.1124	0.1290	0.1251	0.1083	0.1555	0.2397	66.81
	*t*Bnn	<u>0.1189</u>	<u>0.1413</u>	<u>0.1664</u>	<u>0.1126</u>	<u>0.1689</u>	<u>0.3031</u>	<u>73.08</u>
	*t*Bnn-emb	**0.2996**	**0.2938**	**0.2811**	**0.3048**	**0.3860**	**0.5721**	**74.83**
imdb	Bnn	0.2128	0.5106	0.4255	0.1646	0.5699	0.7848	51.16
	Bnn-emb [40]	0.4255	0.5106	0.6383	0.3292	0.5923	1.1358	51.82
	*t*Bnn	**0.8511**	**1.5319**	**1.4043**	<u>0.7548</u>	**1.7381**	**2.6829**	<u>63.46</u>
	*t*Bnn-emb	**0.8511**	<u>1.1064</u>	<u>1.0638</u>	**0.9474**	<u>1.4848</u>	<u>2.2007</u>	**66.87**

the independent and identically distributed (i.i.d) assumption (*bag of teams*) during model training on a shuffled dataset, we order the teams based on time intervals (e.g., yearly) and train a neural model incrementally on streamed subsets of teams. From Figure 7, we randomly initialize the model's parameters at t=0 and train it on the subset of teams in the first time interval $\mathcal{T}_{t=1}$ for a number of epochs. We then continue training on the subsequent time intervals, using the learned parameters from the previous time interval. This process is repeated until we complete the training on the subset of teams in the last time interval $\mathcal{T}_{t=\mathrm{T}}$. Temporal training moves skill and expert embeddings in their optimal positions in the latent space for recommending experts for a team at the future time interval t=T+1. For sample temporal training of models, see our colab script at the codebase.[8]

3.4.1 ***Case Studies of Temporal Learning.*** Table 5, which shows partial results for Bayesian models on dblp and imdb datasets, along with our comprehensive empirical results [12] demonstrate that neural models that utilize the temporal training strategy (*t*Bnn and *t*Bnn-emb) excel at the efficacy of recommended experts for teams.

3.5 Quick Start

Below is an example command with hydra config overrides to adjust the default settings of each component as needed:[9]

```
python -u ./src/main.py "cmd=[prep, train, test, eval, fair]"
  "models.instances=[mdl.bnn.Bnn,                     #Bayesian classifier
                     mdl.nmt.Nmt,                     #translative model
                     mdl.emb.gnn.Gnn,                 #graph-based (e2e)
                     mdl.tntf.tNtf_mdl.bnn.Bnn]"      #temporal training of Bnn

  "data.domain=cmn.publication.Publication"
  "data.source=../data/dblp/toy.dblp.v12.json"
  "data.output=../output/dblp/toy.dblp.v12.json"
  "data.filter.min_team_size=10"
  "data.embedding.class_method=mdl.emb.gnn.Gnn_gs" #GraphSAGE (e2e, transfer)

  "train.nfolds=3"
  "train.train_test_ratio=0.85"
```

[8] https://github.com/fani-lab/OpeNTF/blob/main/ipynb/tntf.ipynb
[9] https://github.com/fani-lab/OpeNTF/blob/main/ipynb/quickstart.ipynb

```
"train.save_per_epoch=3"
"train.step_ahead=1"

"test.per_epoch=True"

"eval.metrics.topk=\'2,5,10\'"
"eval.metrics.trec=[ndcg_cut_topk, map_cut_topk]"
"eval.metrics.other=[skill_coverage_topk]"
"eval.metrics.fair=[ndkl, skew]"
"eval.per_instance=True"                          #for paired sig. tests

"fair.fgender=../data/dblp/toy.dblp.v12.json.females.csv"
"fair.algorithm=[fa-ir, det_greedy]"
"fair.notion=[eo, dp]"
"fair.attribute=[gender, popularity]"
"fair.is_popular_alg=[avg]"
```

4 Concluding Remarks and Future Work

We presented OpeNTF2 with key extensions to its initial release including *i*) fairness-aware rerankers to mitigate popularity and gender disparities; *ii*) graph neural network for transferring embeddings of skills as well as end-to-end team recommendation via link prediction; *iii*) encoder-decoder recurrent models and transformers for team recommendation framed as a sequence prediction (translation) task; *iv*) and training strategy to incorporate temporal aspects of experts' skills and collaboration ties within time. OpeNTF2's future focus is on fairness, incorporating in-process and pre-process mitigation methods, as well as temporal graph neural networks.

References

[1] Md Jamil Ahmed, Mahdis Saeedi, and Hossein Fani. 2024. Vector Representation Learning of Skills for Collaborative Team Recommendation: A Comparative Study. In *Web Information Systems Engineering - WISE 2024 - 25th International Conference, Doha, Qatar, December 2-5, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 15437)*, Mahmoud Barhamgi, Hua Wang, and Xin Wang (Eds.). Springer, 193–207. https://doi.org/10.1007/978-981-96-0567-5_15

[2] Md Jamil Ahmed, Mahdis Saeedi, and Hossein Fani. abs/2026. Graph Neural Team Recommendation: An Integrated Approach. *CoRR* (abs/2026). https://hosseinfani.github.io/res/papers/Graph_Neural_Team_Recommendation_An_Integrated_Approach.pdf Under review / preprint.

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

[4] 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

[5] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1724–1734. https://doi.org/10.3115/V1/D14-1179

[6] Arman Dashti, Saeed Samet, and Hossein Fani. 2022. Effective Neural Team Formation via Negative Samples. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 3908–3912. https://doi.org/10.1145/3511808.3557590

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

[8] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeongmin Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*, Humberto Jesús Corona Pampín, Martha A. Larson, Martijn C. Willemsen, Joseph A. Konstan, Julian J. McAuley, Jean Garcia-Gathright, Bouke Huurnink, and Even Oldridge (Eds.). ACM, 143–153. https://doi.org/10.1145/3460231.3474255

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[10] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 135–144. https://doi.org/10.1145/3097983.3098036

[11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, Shafi Goldwasser (Ed.). ACM, 214–226. https://doi.org/10.1145/2090236.2090255

[12] Hossein Fani, Reza Barzegar, Arman Dashti, and Mahdis Saeedi. 2024. A Streaming Approach to Neural Team Formation Training. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 14608)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 325–340. https://doi.org/10.1007/978-3-031-56027-9_20

[13] 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

[14] 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

[15] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1243–1252. http://proceedings.mlr.press/v70/gehring17a.html

[16] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *KDD*. ACM, 2221–2231.

[17] 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

[18] Alex Graves. 2011. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger (Eds.). 2348–2356. https://proceedings.neurips.cc/paper/2011/hash/7eb3c8be3d411e8ebfab08eba5f49632-Abstract.html

[19] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*,

Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 855–864. https://doi.org/10.1145/2939672.2939754

[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] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 3315–3323. https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

[22] Lu Hong and Scott E Page. 2024. Individual selection criteria for optimal team composition. *Theory and Decision* 96, 4 (2024), 607–626.

[23] 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/

[24] 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

[25] Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 1–10. https://doi.org/10.3115/V1/P15-1001

[26] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1700–1709. https://aclanthology.org/D13-1176/

[27] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 197–206. https://doi.org/10.1109/ICDM.2018.00035

[28] Mehdi Kargar and Aijun An. 2011. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). ACM, 985–994.

[29] 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 2023, Gold Coast, Australia, June 18-23, 2023*. IEEE, 1–8. https://doi.org/10.1109/IJCNN54540.2023.10191717

[30] 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

[31] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, Mohit Bansal and Heng Ji (Eds.). Association for Computational Linguistics, 67–72. https://doi.org/10.18653/V1/P17-4012

[32] Anastasiia Klimashevskaia, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. 2024. A survey on popularity bias in recommender systems. *User Model. User Adapt. Interact.* 34, 5 (2024), 1777–1834. https://doi.org/10.1007/S11257-024-09406-0

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

[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] Hamed Loghmani and Hossein Fani. 2023. Bootless Application of Greedy Re-ranking Algorithms in Fair Neural Team Formation. In *Advances in Bias and Fairness in Information Retrieval - 4th International Workshop, BIAS 2023, Dublin, Ireland, April 2, 2023, Revised Selected Papers (Communications in Computer and Information Science, Vol. 1840)*, Ludovico Boratto, Stefano Faralli, Mirko Marras, and Giovanni Stilo (Eds.). Springer, 108–118. https://doi.org/10.1007/978-3-031-37249-0_9

[36] Hamed Loghmani, Mahdis Saeedi, Gabriel Rueda, Edwin Paul, and Hossein Fani. 2026. A Probabilistic Greedy Attempt to be Fair in Neural Team Recommendation. *CoRR* abs/2026 (2026). https://hosseinfani.github.io/res/papers/2026__COIN__A__Probabilistic__Greedy__Attempt__to__be__Fair__in__Neural__Team__Recommendation.pdf Under review / preprint.

[37] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton (Eds.). The Association for Computational Linguistics, 1412–1421. https://doi.org/10.18653/V1/D15-1166

[38] 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

[39] 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

[40] 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

[41] Radin Hamidi Rad, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2020. Learning to Form Skill-based Teams of Experts. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2049–2052. https://doi.org/10.1145/3340531.3412140

[42] 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*. ACM, 4716–4720. https://doi.org/10.1145/3459637.3481992

[43] Radin Hamidi Rad, Shirin Seyedsalehi, Mehdi Kargar, Morteza Zihayat, and Ebrahim Bagheri. 2022. A Neural Approach to Forming Coherent Teams in Collaboration Networks. In *EDBT*. OpenProceedings.org, 2:440–2:444. https://doi.org/10.48786/edbt.2022.37

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. https://jmlr.org/papers/v21/20-074.html

[45] Anna Sapienza, Palash Goyal, and Emilio Ferrara. 2019. Deep Neural Networks for Optimal Team Composition. *Frontiers Big Data* 2 (2019), 14. https://doi.org/10.3389/fdata.2019.00014

[46] 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

[47] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1441–1450. https://doi.org/10.1145/3357384.3357895

[48] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 3104–3112. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html

[49] Kap Thang, Hawre Hosseini, and Hossein Fani. 2025. Translative Neural Team Recommendation: From Multilabel Classification to Sequence Prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne (Eds.). ACM, 2800–2806. https://doi.org/10.1145/3726302.3730259

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[51] Karan Vombatkere, Theodoros Lappas, and Evimaria Terzi. 2025. A QUBO Framework for Team Formation. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2025, Porto, Portugal, September 15-19, 2025, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 16014)*, Rita P. Ribeiro, Bernhard Pfahringer, Nathalie Japkowicz, Pedro Larrañaga, Alípio M. Jorge, Carlos Soares, Pedro H. Abreu, and João Gama (Eds.). Springer, 492–510. https://doi.org/10.1007/978-3-032-05981-9_29

[52] 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

[53] 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

[54] Xin Wang, Yudong Chen, and Wenwu Zhu. 2022. A Survey on Curriculum Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 9 (2022), 4555–4576. https://doi.org/10.1109/TPAMI.2021.3069908

[55] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=ryGs6iA5Km

[56] Omry Yadan. 2019. Hydra - A framework for elegantly configuring complex applications. Github. https://github.com/facebookresearch/hydra

[57] Andrew Yates, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1154–1156. https://doi.org/10.1145/3437963.3441667

[58] Rex Ying, Ruining He, K. Chen, P. Eksombatchai, W. L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *SIGKDD*. 974–983. https://doi.org/10.1145/3219819.3219890

[59] Meike Zehlike, Tom Sühr, Ricardo Baeza-Yates, Francesco Bonchi, Carlos Castillo, and Sara Hajian. 2022. Fair Top-k Ranking with multiple protected groups. *Inf. Process. Manag.* 59, 1 (2022), 102707. https://doi.org/10.1016/J.IPM.2021.102707

[60] Yi Zhao, Yuzhuo Wang, Heng Zhang, Donghun Kim, Chao Lu, Yongjun Zhu, and Chengzhi Zhang. 2024. Do more heads imply better performance? An empirical study of team thought leaders' impact on scientific team performance. *Inf. Process. Manag.* 61, 4 (2024), 103757. https://doi.org/10.1016/J.IPM.2024.103757

[61] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. *Trans. Assoc. Comput. Linguistics* 4 (2016), 371–383. https://doi.org/10.1162/TACL_A_00105

[62] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. 2016. Scalable Temporal Latent Space Inference for Link Prediction in Dynamic Social Networks. *IEEE Trans. Knowl. Data Eng.* 28, 10 (2016), 2765–2777. https://doi.org/10.1109/TKDE.2016.2591009

[63] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. 2017. Scalable Temporal Latent Space Inference for Link Prediction in Dynamic Social Networks (Extended Abstract). In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*. IEEE Computer Society, 57–58. https://doi.org/10.1109/ICDE.2017.35

[64] 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