A Streaming Approach to Neural Team Formation Training

Hossein Fani(✉), Reza Barzegar, Arman Dashti, and Mahdis Saeedi

University of Windsor, Windsor, ON, Canada
{hfani,barzegar,vaghehd,msaeedi}@uwindsor.ca

Abstract. Predicting *future* successful teams of experts who can effectively collaborate is challenging due to the experts' temporality of skill sets, levels of expertise, and collaboration ties, which is overlooked by prior work. Specifically, state-of-the-art neural-based methods learn vector representations of experts and skills in a *static* latent space, falling short of incorporating the possible drift and variability of experts' skills and collaboration ties in time. In this paper, we propose (1) a streaming-based training strategy for neural models to capture the evolution of experts' skills and collaboration ties over time and (2) to consume time information as an additional signal to the model for predicting future successful teams. We empirically benchmark our proposed method against state-of-the-art neural team formation methods and a strong temporal recommender system on datasets from varying domains with distinct distributions of skills and experts in teams. The results demonstrate neural models that utilize our proposed training strategy excel at efficacy in terms of classification and information retrieval metrics. The codebase is available at https://github.com/fani-lab/OpeNTF/tree/ecir24.

Keywords: Neural Team Formation · Training Strategy · OpeNTF

1 Introduction

Teamwork has shown to be crucial in today's interdisciplinary environment, like in academia [15,28,46], industry [2,6,18], law [17,42], freelancing [4], and the healthcare system [8,40]. Team formation problem aims to automate forming teams of experts whose combined skills, applied in coordinated ways, can solve difficult tasks such as science projects whose success can be measured by publications, or the next blockbuster '*thriller*' with a touch of '*sci-fi*' in the movie industry. Team formation can also be seen as social information retrieval (Social IR) where the *right* group of experts is required to solve the task at hand. Forming teams is challenging due to the large number of candidates from various cultural backgrounds and personality traits as well as unknown synergistic balance among them. More importantly, experts' interests, skills, and levels of expertise change due to society's demands, novel technologies, and working experience. For instance, with the growth of automation, more and more experts are

N. Goharian et al. (Eds.): ECIR 2024, LNCS 14608, pp. 325–340, 2024.
https://doi.org/10.1007/978-3-031-56027-9_20

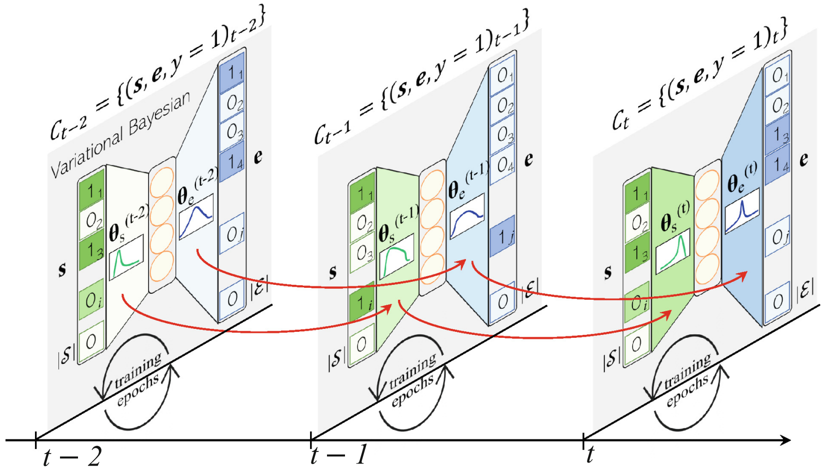

Fig. 1. Streaming training strategy for future team prediction. A neural model learns from the collaborations of experts $\mathcal{C}_t$ at time t to kick-start learning the collaborations at the next time interval $t + 1$. Best viewed in color.

acquiring skills related to computer science, as seen in social science, biology, and linguistics, among other sciences [14,31]. Therefore, a successful collaboration of experts *years ago* would not tailor a successful one in the *future*.

Despite a large body of computational methods to address the team formation problem for an overwhelming number of experts, the positive impacts of considering temporality are yet to be studied. Operations research (OR)-based methods, wherein multiple objective functions are optimized with respect to constraints such as planned budget and timeline via integer programming, forego the temporality of experts' skills and collaboration ties [3,11,39,45]. Graph-based team formation methods represent the expert network as a *static* graph and overlook the evolution of the expert network in time and the emergence of new collaborations [19,22,27,29]. State-of-the-art neural-based methods learn experts and skills vector representations in a *static* latent space, and hence, fall short of incorporating the possible drift of experts' skills in time and its impact on the prediction of future successful teams [9,10,33,34,36–38,41]. Little work considers time but as a *constraint* to model the projects' deadlines, the availability of experts, or uncertainty about the duration of the projects [3,39].

In this paper, we propose a streaming training strategy to encode temporal aspects in neural-based team formation methods. Specifically, given the stream of experts' collaborations in each time interval, a neural model learns the collaborations of experts at time interval t to kick-start learning the collaborations of the next time interval $t + 1; 1 \leq t \leq \mathrm{T}$, as shown in Fig. 1. Our proposed training strategy, when employed by neural models, allows experts to change their vector positions in latent space as their skills and collaboration ties evolve over time, and captures the change trajectories up until time interval T to accurately predict experts' vector positions in *future* time interval T+1. Contrary to non-temporal methods that assume the independent and identically distributed (i.i.d) instances of teams (bag of teams) [9,19,27,36,41], our approach incorporates temporality by streaming the teams within time intervals in its training step. In contrast to considering time as a constraint, we study the horizontal nature of time to learn the evolution of experts' skills and collaboration ties in time. We perform experiments on datasets that enjoy distinct distributions of skills and experts in teams, namely `dblp`[1] [26,27], `imdb`[2] [19,21], `uspt`[3] [24], `gith`[4] [25] to demonstrate the domain-free effectiveness of our proposed method. Comparing our work with the state-of-the-art, our results show that incorporating the temporal evolution of experts' skills and collaboration ties exhibits superior performance in predicting future successful teams of experts.

2 Related Works

Since Zakarian and Kusiak's work [47], there has been a surge of research in the team formation problem that can be differentiated based on their optimization method: (1) search-based, where the task of searching for the best team is executed over *all* the subgraphs of the expert network or via integer programming, and (2) learning-based, where a machine learning algorithm, a neural network in particular, is utilized to form teams of experts by learning the distributions of experts and skills in the context of successful teams in the past. Nonetheless, literature related to the team formation problem has ignored the impact of experts' temporal behavior, by and large, despite widespread successful incorporation of temporality in other domains such as temporal information retrieval, temporal knowledge graphs, and temporal recommender systems [12,30], to name a few. There is little work in team formation [3,11,39,45] that studied time but as a constraint such as the projects' deadlines in the optimization functions. In this section, we review some of the prominent works in the team formation literature.

2.1 Non-temporal Methods

Search-based Methods. The foremost method of team formation was conceived in operations research (OR), where multiple objectives must be optimized simultaneously via integer or real programming to find the *optimum* team, given constraints for human and non-human factors and scheduling preferences. Based on the engineering characteristics of a product and the importance of customer requirements, Zakarian *et al.* [47] used the integer linear programming approach to form multi-functional teams. They imposed integer constraints on experts, such as a cap on the number of projects each expert as a team member may take on. More recently, Neshati *et al.* [32] translate team formation to facility location analysis to form groups of experts to perform a multi-aspect task that

[1] https://aminer.org/citation.
[2] https://imdb.com/interfaces/.
[3] https://uspto.gov/ip-policy/economic-research/research-datasets.
[4] https://codelabs.developers.google.com/codelabs/bigquery-github.

requires a diverse set of skills. Therein, teams were defined as locations, a set of skills as facilities, and experts' membership in teams as customers' needs and the optimization happens for optimal locations of facilities while simultaneously satisfying customers' needs. Such works, however, were premised on the mutually independent selection of experts and overlooked the collaborative and social ties among experts.

Although Chen and Lin [7] were among the first to consider experts' ties for team formation, they were Lappas *et al.* [27] who employed social network analysis to fill the gap by incorporating social ties and interpersonal collaboration features. They represented the experts' social network with a graph where nodes are experts with their set of skills, and edges represent the previous collaboration between them. The optimum team hence can be found by a search on *all* possible subgraphs. They proposed two algorithms based on the diameter of the graph and the cost of the minimum spanning tree (MST) to find a subgraph in which experts collectively hold the required skills and can collaborate effectively with minimum communication cost. However, the diameter of a subgraph or the minimum spanning tree is *in*accurate estimators of the true communication costs in a team, and also sensitive to slight changes in the graph that yield a radical change in the solution. To overcome these issues, Kargar and An [19] proposed two novel communication cost functions that minimize the sum of distances function for teams with a leader and lack thereof. Later, Kargar *et al.* [22] further proposed to consider additional budget constraints (expert salary) on top of communication costs as in real-world scenarios. They proposed a *bi*-objective approximation algorithm to optimize communication cost and salary in tandem.

Methods of efficient keyword search on attributed graphs have also been employed for team formation [25,26]. For instance, given a set of query keywords as skills and the desired size of the subgraph as the team size, Khan *et al.* [25] aimed to find closely connected subgraphs with the specific number of nodes wherein nodes contain as many query keywords as possible. Since the total number of answers is exponential in the number of query keywords and the size of the group, they proposed a method to find the approximate top-k groups with polynomial delay. Nonetheless, OR or graph-based optimization models for the task of team formation are computationally intractable and have to be followed by polynomial heuristic solutions such as multichoice [1] for subgraph identification with shortest diameter [27], or simulated annealing [3], branch-and-cut, genetic algorithms [43], and balanced placement [13] for those based on integer programming (IP). Indeed, IP is NP-hard, and subgraph optimization can be reduced to the decision version of the Steiner-tree problem, which is also proved to be NP-hard [23].

Learning-based Methods. Recently, a paradigm shift to machine learning has been observed in team formation literature, opening doors to the analysis of massive collections of experts from different domains. Machine learning approaches efficiently learn relationships between experts and their skills in the context of successful (positive samples) and unsuccessful teams (negative samples) from all past instances [9,10,33,34,36–38,41]. Sapienza *et al.* [41] employed a deep neural autoencoder to form teams and to capture which teammates foster the growth of their peers. However, when training data suffers from the popularity bias, such as in the team formation problem where a few experts have participated in the majority of teams for a small subset of skills while many experts have participated sparingly, autoencoder neural networks are prone to overfitting [5]. Rad *et al.* [36] proposed a variational Bayesian neural architecture to employ uncertainty in learnable parameters and overcome popularity bias. However, they only utilized past successful teams to train their neural model. Dashti *et al.* [9] proposed to utilize negative sampling heuristics to incorporate both successful and *virtually* unsuccessful teams in their training, which resulted in more efficient and effective neural models during training and inference, respectively. Nonetheless, existing learning-based methods neglected the *temporal* nature of experts' skills and collaborative ties.

2.2 Time as Constraint

There has been little work that used time as a constraint to model experts' availabilities or predefined start and due dates of projects. Durfee *et al.* [11] take into account scheduling constraints or preferences in a two-step team formation process. First, teams are built in the matchmaking optimization stage using integer linear programming, taking into account the required skills as well as the ability to be more readily (re)scheduled with respect to the timing requirement. Next, in the scheduling optimization stage, time slots are allotted to the team for completing the task using integer *non*linear programming optimization in a way that minimizes the total delay of the starting times of all the members while satisfying sequential and concurrent ordering constraints. Rahmanniyay *et al.* [39] studied the impact of various factors like weather conditions that can change the duration of a project or delay the delivery of material to a manufacturing company. Yang *et al.* [45] apply integer programming to determine the optimum team of experts *available* at a certain point in time. Contrary to considering time as an optimization constraint, we propose to treat time as an *aspect* through which experts' skills and collaboration ties evolve.

3 Problem Definition

We aim to incorporate the evolution of experts' skills and collaborative ties over time in order to predict *future* teams of experts who collectively hold a set of required skills and can effectively cooperate toward a shared goal based on their gained experience through time. Let $\mathcal{S}$ and $\mathcal{E}$ be the sets of skills and experts, and $\mathcal{C}_t = \{(s, e, y)_t | s \subseteq \mathcal{S}, e \subseteq \mathcal{E}, y \in \{0, 1\}\}$ be the set of collaborations at time t where (s, e) is a team whose members are a subset of experts, e, that collectively hold the subset of skills, s, and has been either successful $y = 1$ or a failure $y = 0$, and t is a discrete entity showing the time intervals. Intuitively, $\mathcal{C}_t$ is a snapshot of all teams of experts over skills during the time interval t and $[\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}]$ streams the dynamic distribution of experts over skills within T consecutive time intervals in the context of teams. Examples of teams include research groups where researchers are the experts and fields of study are the skills, movies consisting of casts and crews such as actors and directors as the experts and the genres as the skills, patents consisting of inventors as the experts and categories (classes) as the skills, or software projects where software developers and programming languages are the experts and the skills, respectively. Figure 2 demonstrates the *non*-uniform and temporal distribution of movies over genres (skills) and casts and crews within time in `imdb` dataset. As seen, although the set of genres remains the same over 100 years, the number of movies that adopt each genre varies over time. Also, an actor (expert) adopts various genres (skills) during his career. In real world, a similar trend can also be observed in research (`dblp`), patents (`uspt`), and computer software (`gith`) domains.

Strangely, the basic question of *"what it means for a team to be successful"* has gone underexamined and has remained controversial in the literature. Finding experts who collectively cover the required skills for a team is *in*sufficient and error-prone for a successful team since skillful experts enjoy various cultural backgrounds and personality traits that result in an unknown synergistic balance among them. Recently, little learning-based work (Sect. 2.1) has defined success (failure) based on the tangible outcomes of a team, like the number of publications for a research group, or the number of issued patents for a team of inventors. In some domains, however, what constitutes success remains controversial. For example, in the movie industry, a movie's success can be measured based on its immediate reception by the people (box office) or critical reviews (ratings) within a long span of time. Nonetheless, a team's label of success y can be redefined without loss of generality in our proposed method. For instance, success can be redefined based on the number of citations for a research paper, critical acclaim for a movie, and commercialization for a patent. In the absence of unlabeled unsuccessful teams, state-of-the-art learning-based methods follow the closed-world assumption; they presume existing instances of teams in the training dataset as successful ($y = 1$) and subsets of experts who have *not* collaborated yet for the input skills as unsuccessful teams (*virtually* negative samples).

Given the stream of collaboration sets $[\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}]$ in the past, we aim to recommend a *new* team of experts e' for a given subset of skills s at a yet-to-be-seen time interval T+1 whose collaboration has a high chance of success, i.e., $(s, e', 1)_{\mathrm{T}+1}$, also referred to as an *optimum* team. More formally, we aim to estimate a mapping function f of parameters θ from the stream of collaboration sets and a subset of skills to a subset of experts whose collaboration in a team is almost surely successful for the one-step-ahead *future* time interval T+1; that is, $f([\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}], s; \theta) = e'$ such that $(s, e', y = 1)_{\mathrm{T}+1}$.

4 Proposed Method

The main contribution of this paper is not a novel machine learning model but a training strategy for such models to take into account the temporal nature of

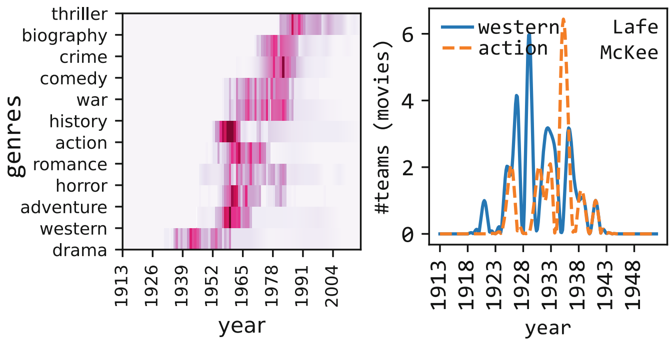

Fig. 2. Temporal distribution of movies over genres (left), and temporal inclination of an actor toward two genres (right). Best viewed in color.

the data in team formation. Let $[\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}]$ be the ordered list of all previous collaborations at each time interval t until T in which experts' collaborations and their skills in teams are evolving over time. We estimate f using a neural model that maximizes the average log probability of successful subsets of experts:

$$\frac{1}{|\mathcal{C}_{\mathrm{T}+1}|} \sum_{(s,e,y)\in\mathcal{C}_{\mathrm{T}+1}} \log \mathrm{p}(y|(s,e):\mathrm{T}+1) \tag{1}$$

where $\mathcal{C}_{\mathrm{T}+1}$ is the collection of yet-to-be-formed unseen (un)successful teams (s, e, y) in the *future* time interval T+1. Since $\mathcal{C}_{\mathrm{T}+1}$ is unseen, we optimized Eq. 1 through observed teams of (s, e, y) in the past:

$$\sum_{t=1}^{\mathrm{T}} \frac{1}{|\mathcal{C}_t|} \sum_{(s,e,y)\in\mathcal{C}_t} \log \mathrm{p}(y|(s,e):t) \tag{2}$$

The same team (s, e) may experience success and/or failure in different time intervals. Therefore, $p(y|(s, e) : t)$ depends on the time interval information. To maximize Eq. 2, we map each subset of skills s and each subset of expert e to a low-rank d-dimensional vector in the same latent space, denoted by v_s and v_e, whose positions up until time interval T depend on the preceding movements in the latent space since the first time interval via observation of $[\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}]$ while imposing the following assumptions: (i) skills and experts change their latent representations over time, (ii) subsets of experts who collaborated in teams over similar subsets of skills within $[\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}]$ remain close in latent space, (iii) subsets of experts and skills who are close in latent space at their final positions in the latent space are presumably the optimum teams whose successes are almost surely guaranteed in the *future* time interval T+1.

4.1 Streaming Learning

Previous work in team formation assumed teams are independent and identically distributed and followed the bag of teams approach during model training on a shuffled dataset [3,9,19,20,27,36,39,41]. Further, they evaluated their models on a randomly selected subset of teams as the test set, instead of predicting future successful teams. In this work, however, we train a neural model incrementally over an ordered collection of teams from $[\mathcal{C}_1..\mathcal{C}_t..\mathcal{C}_\mathrm{T}]$. As seen in Fig. 1, after random initialization of skills' and experts' embeddings, we start training the model on the teams from the first time interval $\mathcal{C}_1$ for several epochs, then we continue with training (fine-tuning) on the teams of second time interval $\mathcal{C}_2$ using the learned embeddings from the first time interval and so forth until we finish the training on the last training time interval $\mathcal{C}_\mathrm{T}$. We believe that using this approach helps the model observe how experts' skills and collaborative ties evolve through time, and hence the final embeddings are their optimum representations in the latent space to predict future successful teams.

At each time interval t, we estimate $p(y|(s,e):t)$ through pairwise cosine similarities of embeddings for the subset of experts e and the subset of skills s through all (un)successful teams at time interval t in $\mathcal{C}_t$. More specifically, we estimate $p(y=1|(s,e):t)$ by learning v_e and v_s that are close (high cosine similarity) in the latent space if the subset of experts e has successful collaborations in $\mathcal{C}_t$ with the subset of skills s during the time interval t and estimate $p(y=0|(s,e):t)$ by learning v_e and v_s that are distant (low cosine similarity) otherwise. Hence, $p(y|(s,e):t)$ can be formulated with the sigmoid function σ:

$$p(y|(s,e):t) = \sigma(v_e^\top \cdot v_s) \quad (3)$$

Like Dashti *et al.* [9], when no unsuccessful team is available in the training set, we follow the closed-world assumption to generate *virtually* unsuccessful teams (negative samples), that is, if no successful team for the subset of skills s is known for a randomly selected subset of experts e'' at time interval t, i.e., $(s,e'') \notin \mathcal{C}_t$, the team is considered to be unsuccessful $(s,e'',y=0)$. To this end, we employ an optimization function that discriminates successful and unsuccessful teams through negative sampling from a distribution over the subsets of experts:

$$\sum_{(s,e)\in\mathcal{C}_t \leftrightarrow (s,e,y=1)} [\log\sigma(v_e^\top \cdot v_s) + \sum_{(s,e'')\sim\mathbb{P}:(s,e'')\notin\mathcal{C}_t \leftrightarrow (s,e'',y=0)}^{k} \log\sigma(-v_{e''}^\top \cdot v_s)] \quad (4)$$

where $\mathbb{P}$ is the probability distribution from which we randomly draw k subsets of experts e'' as negative samples for a given subset of skills s. The input layer of the neural model is either (i) sparse occurrence vector representations for skills of size $|\mathcal{S}|$, (ii) pre-trained dense vector representations (`emb`) for the subsets of skills as suggested by Rad *et al.* [36], or (iii) temporal dense skill vector representations (`dt2v`) using temporal word embedding method by Hamilton *et al.* [16] to directly incorporate temporal evolution of skills into the underlying neural model in addition to our proposed streaming strategy. The output layer of the model is sparse occurrence vector representations for experts of size $|\mathcal{E}|$.

Table 1. Statistics of the raw and preprocessed datasets.

	dblp		uspt		imdb		gith	
	raw	filtered	raw	filtered	raw	filtered	raw	filtered
#teams	4,877,383	99,375	7,068,508	152,317	507,034	32,059	132,851	11,312
#unique experts	5,022,955	14,214	3,508,807	12,914	876,981	2,011	452,606	2,686
#unique skills	89,504	29,661	241,961	67,315	28	23	20	19
avg #expert per team	3.06	3.29	2.51	3.79	1.88	3.98	5.52	7.53
avg #skill per team	8.57	9.71	6.29	9.97	1.54	1.76	1.37	1.57
avg #team per expert	2.97	23.02	5.05	44.69	1.09	62.45	1.62	31.72
avg #skill per expert	16.73	96.72	19.49	102.53	1.59	10.85	2.03	5.18
#team w/ single expert	768,956	**0**	2,578,898	**0**	322,918	**0**	**0**	**0**
#team w/ single skill	5,569	56	939,955	8,110	315,503	15,180	69,131	6014

5 Experiments

In this section, we lay out the details of our experiments and findings toward answering the following research questions:

RQ1: Does moving embeddings of experts and skill through time in the latent space improve the performance of neural models for the prediction of *future* successful teams? To this end, we benchmark state-of-the-art variational Bayesian neural network [9] (`bnn-*`) that utilizes negative sampling heuristics with our proposed streaming scenario training approach (t`bnn-*`) and lack thereof.

RQ2: Does adding time explicitly to the input embeddings of skills boost neural models performance? We compare the performance of neural models with utilizing temporal skills in the input t`bnn_dt2v_emb` and lack thereof t`bnn_emb`.

RQ3: Is the impact of our proposed training strategy consistent across datasets from various domains with distinct statistical distributions? We benchmark our proposed training approach on `dblp`, `imdb`, `uspt`, and `gith` datasets.

5.1 Setup

Dataset. We evaluate our proposed method on four well-known benchmark datasets in team formation literature, namely `dblp` [26,27], `imdb` [19,21], `uspt` [24], `gith` [25]. In `dblp`, each instance is a publication in computer science consisting of authors, the fields of study (fos), and the year it was published including papers from 1979 to 2018. We map each publication to a team whose authors are the experts and fields of studies are the set of skills. In `imdb`, each instance is a movie consisting of its cast and crew such as director, producer, actors, genre and the year the movie was released, spanning from 1914 to 2020. We consider each movie as a team whose members are the cast and crew, and the movies' genres are the teams' skills. The choice of `imdb` in team formation literature is not to be confused with its use cases in recommender systems or review analysis research; herein, the goal is to form a team of casts and

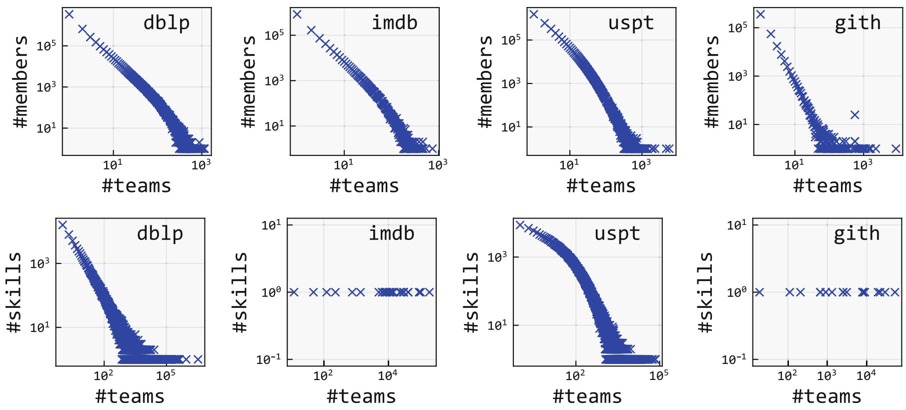

Fig. 3. Distribution of teams over members and skills for all datasets *before* preprocessing.

crews for a movie production as opposed to a movie recommendation [19,21]. In `uspt`, each instance is a patent invention in the United States Patents and Trademarks consisting of inventors (experts) and subcategories (skills) and the time the patent is issued, consisting of patents from 1976 to 2019. In `gith`, each instance is a GitHub repository consisting of the contributors of the repository (experts), the title and programming languages of the project (skills), and the time of the project's release, consisting of repositories from 2008 to 2022.

In all datasets, we can observe the long tail problem in the distributions of teams over experts. As shown in Fig. 3, many experts (researchers in `dblp`, cast and crew in `imdb`, inventors in `uspt`, and developers in `gith`) have participated in very few teams (papers in `dblp`, movies in `imdb`, inventions in `uspt`, and repositories in `gith`). For instance, 10^6 researchers have participated in 1 team only while few researchers have co-authored more than 10^3 papers in `dblp`. With respect to the set of skills, `dblp` and `uspt` are clearly following different distributions compared to `imdb` and `gith`. While `dblp` and `uspt` suffer further from the long-tailed distribution of skills in teams, `imdb` and `gith` follow a more fair distribution, as shown in Fig. 3. Specifically, `imdb` and `gith` have a limited variety of skills (genres and programming languages) which are, by and large, employed by many movies and repositories, respectively.

We filter out singleton and sparse teams with less than `3` members as well as experts who relatively participated in very few teams, as suggested by [9,36]. The latter also reduced the computational complexity of the neural models in their last layer where the size equals the number of experts. We filter out experts who participated in less than `75` teams for `dblp`, `imdb`, and `uspt`, and less than `10` teams for `gith`. From Fig. 4, we ensured that the preprocessing step made no major change to the statistical distributions of the datasets. Also, Table 1 reports additional point-wise statistics on the datasets.

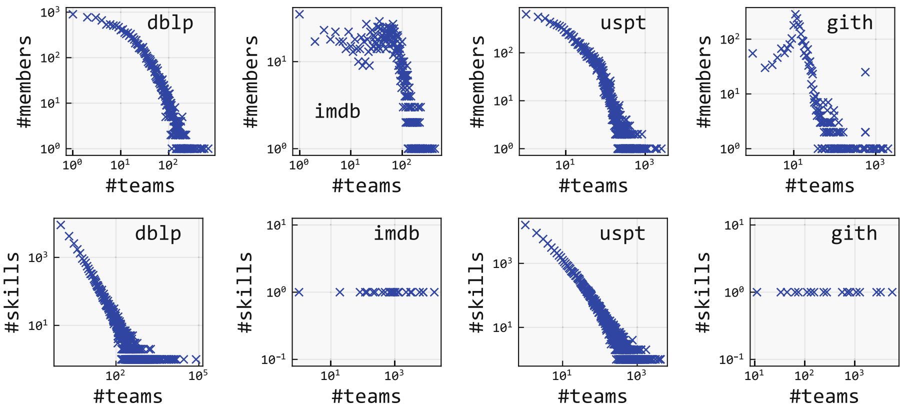

Fig. 4. Distribution of teams over members and skills for all datasets *after* preprocessing.

Baselines. We compare our temporal neural models using streaming training strategy (t`bnn-*`) with (1) non-temporal Bayesian (variational) neural network [9] (`bnn-*`) and (2) recurrent recommender networks [44] (`rrn`), where we recommend experts as items to input skills as users. In contrast to conventional recommender systems that assume users' profiles and items' attributes are static, `rrn` captures their temporal dynamics to predict future behavioral trajectories using a long short-term memory (lstm) autoregressive model and to excel at prediction accuracy. Both temporal and non-temporal Bayesian neural networks utilize the negative sampling objective function (Eq. 4) and include a single hidden layer of size `d = 128`, and `relu` and `sigmoid` are the activation functions for the hidden layer and the output layer, respectively. We used the smoothed unigram distribution in each training mini-batch [9] to generate the negative samples in Eq. 4. We train the model at each time interval t with a learning rate of `0.1` over `20` epochs with mini-batches of size `128` and use `Adam` as the optimizer. We used the same hyper-parameters for `rrn`.

Evaluation. To test the impact of the streaming training strategy and incorporation of time information to the input embeddings in the prediction of *future* successful teams, we take the last year of each dataset for the test set. To ensure the effectiveness of our approach, we perform `5`-fold cross-validation on the teams in each year for model training and validation. Given a team $(s, e)_{\mathrm{T}+1}$ from the test set, we compare the ranked list of predicted experts e' by the model of each fold with the observed subset of experts e and report the average performance of models trained on each fold by information retrieval metrics including normalized discounted cumulative gain (`ndcg`), and mean average precision (`map`) at top-{2,5,10} as well as classification metrics including precision (`pr`) and recall (`rec`) at top-{2,5,10} and area under the receiver operating characteristic curve (`aucroc`).

Table 2. Average performance of 5-fold neural models on the test set.

dblp	%pr2	%pr5	%pr10	%rec2	%rec5	%rec10	%ndcg2	%ndcg5	%ndcg10	%map2	%map5	%map10	%aucroc
bnn [36]	0.0570	0.0663	0.0710	0.0351	0.0993	0.2118	0.0538	0.0806	0.1330	0.0242	0.0411	0.0558	63.52
bnn_emb [35]	0.1124	0.1290	0.1251	0.0668	0.1909	0.3699	0.1083	0.1555	0.2397	0.0474	0.0792	0.1033	66.81
rrn [44]	0.0570	0.0391	0.0472	0.0380	0.0630	0.1552	0.0478	0.0523	0.0959	0.0217	0.0281	0.0446	50.73
*t*bnn	0.1189	0.1413	0.1664	0.0706	0.2090	0.4984	0.1126	0.1689	0.3031	0.0484	0.0845	0.1223	73.08
*t*bnn_emb	0.2996	0.2938	0.2811	0.1816	0.4433	0.8431	0.3048	0.3860	0.5721	0.1411	0.2095	0.2635	74.83
*t*bnn_dt2v_emb	**0.4299**	**0.3973**	**0.3612**	**0.2601**	**0.5963**	**1.0801**	**0.4284**	**0.5221**	**0.7465**	**0.1947**	**0.2864**	**0.3520**	**77.01**
gith													
bnn [36]	0.2128	0.5106	0.4255	0.1418	0.8511	1.3050	0.1646	0.5699	0.7848	0.0709	0.2600	0.3148	51.16
bnn_emb [35]	0.4255	0.5106	0.6383	0.2837	0.8511	1.9574	0.3292	0.5923	1.1358	0.1418	0.2813	0.4389	51.82
rrn [44]	0.0000	0.8511	0.8511	0.0000	1.4184	2.8369	0.0000	0.8163	1.4606	0.0000	0.3191	0.6265	52.22
*t*bnn	0.8511	**1.5319**	1.4043	0.5319	**2.4610**	4.4965	0.7548	1.7381	2.6829	0.3369	0.8215	1.1674	63.46
*t*bnn_emb	0.8511	1.1064	1.0638	0.5674	1.7518	1.3262	0.9474	1.4848	2.2007	0.4965	0.8138	1.0099	**66.87**
*t*bnn_dt2v_emb	**1.9149**	1.1915	**1.4468**	**1.2411**	1.9504	**4.5532**	**1.8667**	**1.8703**	**3.0303**	**0.9043**	**1.1099**	**1.4293**	66.56
uspt													
bnn [36]	0.0657	0.0769	0.0910	0.0353	0.0976	0.2212	0.0655	0.0883	0.1481	0.0266	0.0433	0.0592	64.54
bnn_emb [35]	0.3663	0.4123	0.3748	0.1608	0.4509	0.8141	0.3652	0.4531	0.6094	0.1212	0.2027	0.2583	69.85
rrn [44]	0.0239	0.0383	0.0654	0.0140	0.0500	0.1370	0.0221	0.0408	0.0868	0.0096	0.0186	0.0340	51.60
*t*bnn	0.1843	0.1841	0.2029	0.0933	0.2321	0.5158	0.1794	0.2152	0.3481	0.0681	0.1056	0.1429	75.44
*t*bnn_emb	0.8272	0.7539	0.7042	0.3970	0.9021	1.6933	0.8457	0.9057	1.2657	0.3104	0.4533	0.5679	83.59
*t*bnn_dt2v_emb	**1.2268**	**1.0583**	**0.9324**	**0.6037**	**1.2928**	**2.2518**	**1.2322**	**1.2960**	**1.7348**	**0.4626**	**0.6659**	**0.8118**	**85.34**
gith													
bnn [36]	3.0693	2.8515	2.6931	1.2164	2.8846	5.1174	3.1365	3.2893	4.2340	1.0104	1.5706	2.1633	56.18
bnn_emb [35]	**7.3267**	**4.7129**	3.3861	**3.5441**	**5.1580**	**6.1885**	**6.4753**	**5.8418**	**6.2665**	**2.3424**	**3.0822**	**3.3837**	62.65
rrn [44]	0.0000	0.1980	0.0990	0.0000	0.0619	0.0619	0.0000	0.1679	0.1090	0.0000	0.0206	0.0206	52.26
*t*bnn	3.8614	2.8515	2.3564	1.8801	3.1525	4.5754	4.3319	3.9721	4.5031	1.8025	2.3978	2.8768	56.65
*t*bnn_emb	4.9505	3.5248	3.1287	1.9434	3.0770	4.3718	5.0849	4.4715	4.9844	1.6957	2.1431	2.5949	62.20
*t*bnn_dt2v_emb	5.7426	4.5941	**3.8020**	2.1874	3.8474	4.7855	5.6081	5.3287	5.6670	1.7131	2.4258	2.7858	**64.89**

5.2 Results

Foremost, we acknowledge that baselines achieve low values of evaluation metrics for practical application of team formation, which is primarily due to the simplicity of the neural model architectures and the small number of training epochs; metric values are reported in % for ease of readability and comparison. Our main goal is *not* to report state-of-the-art results for a novel model *but* to showcase the synergistic effects of our proposed training strategy for such models.

In response to **RQ1**, i.e., whether the streaming training strategy improves the predictive power of state-of-the-art neural models, from Table 2, comparing bnn and bnn_emb with *t*bnn and *t*bnn_emb respectively, we can observe that streaming training strategy increases neural models' relative performance between 10% to 20% on dblp and uspt in terms of the classification metrics (aucroc). On imdb and gith, it also improves the performance of neural models in terms of the information retrieval metrics. More specifically, on imdb, comparing bnn_emb with *t*bnn_emb, we can observe a relative gain of near 200% on some metrics (ndcg2, ndcg5, map2, and map5). Moreover, our training strategy increases neural models' relative performance on most of the information retrieval metrics between 100% and 200% on dblp and uspt. On gith, however, we can observe that using the streaming training strategy decreases the models' performance when using pretrained dense vector representation for input skills even though *t*bnn outperforms bnn in most of the information retrieval metrics.

In response to **RQ2**, i.e., whether adding time explicitly to the input of the neural model improves its performance while utilizing the streaming training strategy, from Table 2, comparing *t*bnn_emb with *t*bnn_dt2v_emb, we see that the models that utilize temporal skills in the input gain relative performance of between 30% to 50% in terms of *all* information retrieval metrics on dblp, uspt, and gith and up to 100% on imdb. Comparing neural models with sparse skill input representation (*t*bnn) with the ones that utilize temporal skill embeddings *t*bnn_dt2v_emb, we observe a substantial gain in relative performance in terms of information retrieval metrics between 100% and 500% on dblp and uspt. On gith, we still observe an increase in models' performance, but the gain in performance is not as substantial. Finally, on imdb, *t*bnn outperforms *t*bnn_dt2v_emb on some of the metrics such as pr5 and rec5 and perform on a par on pr10 and rec10. Nonetheless, we observe a relative gain of up to 100% on other information retrieval metrics on imdb. In summary, the temporal dense vector representation of skills always leads to a performance improvement in terms of classification and information retrieval metrics.

Regarding **RQ3**, i.e., whether the impact of our proposed streaming training strategy is consistent across different datasets with distinct distributions of skills and experts, from Table 2, we can see that the degree of the increase in performance of neural models depends on the distributions of experts and skills (Figs. 3 and 4) in teams and the evolution of experts and skills over time (Fig. 2). More concretely, for datasets with a long-tailed distribution of skills in teams (dblp and uspt), utilizing our proposed streaming strategy will help neural models in the prediction of *future* successful teams, which is contrary to datasets with a limited set of skills that are employed almost uniformly by teams (imdb and gith) (Fig. 4). Finally, from Table 2, we can see that the results of our proposed training strategy and incorporation of temporal skills are always superior compared to the temporal recommender system baseline [44] (rrn) on all four datasets for all the metrics.

6 Concluding Remarks

In this paper, we proposed a streaming training strategy for neural models to learn the evolution of experts' skills and collaborative ties to predict future successful teams. We further examined the impact of adding temporal information to the input of neural models. Our experiments on four datasets with distinct distributions of teams over skills and experts in time show that (1) our proposed streaming training strategy improves the predictive power of neural models, (2) neural models that leverage temporal information in the input obtain better performance compared to the lack thereof in most cases, and (3) neural models utilizing our proposed training strategy outperform the temporal recommender system baseline. Possible future directions of our work include spatio-temporal study of team formation where both temporal dynamics of experts and skills in teams as well as their geo-locations are considered to recommend location-based future teams with minimum communication costs.

References

1. Arkin, E., Hassin, R.: Minimum diameter covering problems. Networks **36**, 147–155 (2000)
2. Askari, G., Asghri, N., Gordji, M.E., Asgari, H., Filipe, J.A., Azar, A.: The impact of teamwork on an organization's performance: a cooperative game's approach. Mathematics **8**(10), 1–15 (2020)
3. Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. Cybern. Syst. Int. J. **38**(2), 155–185 (2007)
4. Bernabé, R.B., Navia, I.A., García-Peñalvo, F.J.: Faat: freelance as a team. In: Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM 2015, pp. 687–694, New York, NY, USA. Association for Computing Machinery (2015)
5. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1613–1622, Lille, France, 07–09 Jul 2015. PMLR (2015)
6. Bursic, K.M.: Strategies and benefits of the successful use of teams in manufacturing organizations. IEEE Trans. Eng. Manage. **39**(3), 277–289 (1992)
7. Chen, S.-J., Li, L.: Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. IEEE Trans. Eng. Manag. **51**(2), 111–124 (2004)
8. Craig, M., McKeown, D.: How to build effective teams in healthcare. Nurs. Times **111**(14), 16–18 (2015)
9. Dashti, A., Samet, S., Fani, H.: Effective neural team formation via negative samples. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM 2022, pp. 3908–3912, New York, NY, USA. Association for Computing Machinery (2022)
10. Dashti, A., Saxena, K., Patel, D., Fani, H.: OpeNTF: a benchmark library for neural team formation. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022, pp. 3913–3917. ACM (2022)
11. Durfee, E.H., Boerkoel, J.C., Sleight, J.: Using hybrid scheduling for the semi-autonomous formation of expert teams. Future Gener. Comput. Syst. **31**, 200–212 (2014). Special Section: Advances in Computer Supported Collaboration: Systems and Technologies
12. Fani, H., Bagheri, E., Du, W.: Temporal latent space modeling for community prediction. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 745–759. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_49
13. Fitzpatrick, E., Askin, R.G.: Forming effective worker teams with multi-functional skill requirements. Comput. Ind. Eng. **48**(3), 593–608 (2005)
14. Gu, Y., et al.: Domain-specific language model pretraining for biomedical natural language processing. ACM Trans. Comput. Healthcare **3**(1) 2021
15. Hall, K., et al.: The science of team science: a review of the empirical evidence and research gaps on collaboration in science. Am. Psychol. **73**, 532–548 (2018)

16. Hamilton, W.L., Leskovec, J., Jurafsky, D.: Diachronic word embeddings reveal statistical laws of semantic change. In: ACL 2016 (2016)
17. Jia, H., Liden, R.C.: Making a difference in the teamwork: linking team prosocial motivation to team processes and effectiveness. Acad. Manag. J. **58**, 1102–1127 (2014)
18. A., Kairgalievna, Zayed, N.M.: The effect of teamwork on employee productivity (2021)
19. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 985–994 (2011)
20. Kargar, M., An, A.: Efficient top-k keyword search in graphs with polynomial delay. In: 2012 IEEE 28th International Conference on Data Engineering, pp. 1269–1272 (2012)
21. Kargar, M., Golab, L., Srivastava, D., Szlichta, J., Zihayat, M.: Effective keyword search over weighted graphs. IEEE Trans. Knowl. Data Eng. **34**(2), 601–616 (2022)
22. Kargar, M., Zihayat, M., An, A.: Finding affordable and collaborative teams from a network of experts. In: Proceedings of the 2013 SIAM International Conference on Data Mining, pp. 587–595. SIAM (2013)
23. Karp, R.M.: Reducibility among Combinatorial Problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds) Complexity of Computer Computations. The IBM Research Symposia Series, pp. 85–103. Springer, Boston (1972). https://doi.org/10.1007/978-1-4684-2001-2_9
24. Keane, P., Ghaffar, F., Malone, D.: Using machine learning to predict links and improve steiner tree solutions to team formation problems - a cross company study. Appl. Netw. Sci. **5**(1), 57 (2020)
25. Khan, A., Golab, L., Kargar, M., Szlichta, J., Zihayat, M.: Compact group discovery in attributed graphs and social networks. Inf. Process. Manag. **57**(2), 102054 (2020)
26. Kou, Y., et al.: Efficient team formation in social networks based on constrained pattern graph. In: 36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, 20–24 April 2020, pp. 889–900. IEEE (2020)
27. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: SIGKDD 2009, pp. 467–476. ACM (2009)
28. Leahey, E.: From sole investigator to team scientist: trends in the practice and study of research collaboration. Ann. Rev. Sociol. **42**(1), 81–100 (2016)
29. Li, C.-T., Shan, M.-K., Lin, S.-D.: On team formation with expertise query in collaborative social networks. Knowl. Inf. Syst. **42**(2), 441–463 (2015)
30. Liao, S., Liang, S., Meng, Z., Zhang, Q.: Learning dynamic embeddings for temporal knowledge graphs. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM 2021, pp. 535–543, New York, NY, USA. Association for Computing Machinery (2021)
31. McCormick, T.H., Lee, H., Cesare, N., Shojaie, A., Spiro, E.S.: Using twitter for demographic and social science research: tools for data collection and processing. Sociol. Methods Res. **46**(3), 390–421 (2017)
32. Neshati, M., Beigy, H., Hiemstra, D.: Expert group formation using facility location analysis. Inf. Process. Manag. **50**(2), 361–383 (2014)
33. Rad, R.H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: Retrieving skill-based teams from collaboration networks. In: SIGIR 2021: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, 11–15 July 2021, pp. 2015–2019. ACM (2021)

34. Rad, R.H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: Subgraph representation learning for team mining. In: WebSci 2022: 14th ACM Web Science Conference 2022, Barcelona, Spain, 26–29 June 2022, pp. 148–153. ACM (2022)
35. Rad, R.H., Fani, H., Bagheri, E., Kargar, M., Srivastava, D., Szlichta, J.: A variational neural architecture for skill-based team formation. ACM Trans. Inf. Syst. (2023). Just Accepted
36. Rad, R.H., Fani, H., Kargar, M., Szlichta, J., Bagheri, E.: Learning to form skill-based teams of experts. In: CIKM 2020, pp. 2049–2052. ACM (2020)
37. Rad, R.H., Mitha, A., Fani, H., Kargar, M., Szlichta, J., Bagheri, E.: PyTFL: a python-based neural team formation toolkit. In: Demartini, G., Zuccon, G., Culpepper, J.S., Huang, Z., Tong, H. (eds.) CIKM 2021: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, 1–5 November 2021, pp. 4716–4720. ACM (2021)
38. Rad, R.H., Seyedsalehi, S., Kargar, M., Zihayat, M., Bagheri, E.: A neural approach to forming coherent teams in collaboration networks. In: Proceedings of the 25th International Conference on Extending Database Technology, EDBT 2022, Edinburgh, UK, March 29 - April 1, 2022, pp. 2:440–2:444. OpenProceedings.org (2022)
39. Rahmanniyay, F., Yu, A.J., Seif, J.: A multi-objective multi-stage stochastic model for project team formation under uncertainty in time requirements. Comput. Ind. Eng. **132**, 153–165 (2019)
40. Rosen, M.A., et al.: Teamwork in healthcare: key discoveries enabling safer, high-quality care. Am. Psychol. **73**(4), 433–450 (2018). Cited by: 297. All Open Access, Green Open Access
41. Sapienza, A., Goyal, P., Ferrara, E.: Deep neural networks for optimal team composition. Front. Big Data **2**, 14 (2019)
42. Sherer, P.D.: Leveraging human assets in law firms: human capital structures and organizational capabilities. ILR Rev. **48**(4), 671–691 (1995)
43. Wi, H., Seungjin, O., Mun, J., Jung, M.: A team formation model based on knowledge and collaboration. Expert Syst. Appl. **36**(5), 9121–9134 (2009)
44. Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. WSDM 2017, pp. 495–503, New York, NY, USA. Association for Computing Machinery (2017)
45. Yang, D.-N., Chen, Y.-L., Lee, W.-C., Chen, M.-S.: On social-temporal group query with acquaintance constraint. Proc. VLDB Endow. **4**(6), 397–408 (2011)
46. Younglove-Webb, J., Gray, B., Abdalla, C.W., Thurow, A.P.: The dynamics of multidisciplinary research teams in academia. Rev. High. Educ. **22**(4), 425–440 (1999)
47. ARMEN Zzkarian and Andrew Kusiak: Forming teams: an analytical approach. IIE Trans. **31**(1), 85–97 (1999)