Adaptive Loss-based Curricula for Neural Team Recommendation

ABSTRACT

Neural team recommendation models have brought state-of-the-art efficacy while enhancing efficiency at recommending collaborative teams of experts who, more likely than not, can solve complex tasks. Yet, they suffer from popularity bias and overfit to a few dominant popular experts and, hence, result in discrimination and reduced visibility for already disadvantaged *non*-popular experts. Such models are trained on randomly shuffled datasets with the disproportionate distribution of a few popular experts over many teams and a sparse long-tailed distribution of non-popular ones, overlooking the difficulty of recommending *hard* non-popular vs. *easy* popular experts. To bridge the gap, we propose three curriculum-based learning strategies to empower neural team recommenders sifting through easy popular and hard non-popular experts and to mitigate popularity bias and improve upon the existing neural models. We propose (1) a parametric curriculum that assigns a learnable parameter to each expert enabling the model to learn an expert's levels of difficulty (or conversely, levels of popularity) during training, (2) a parameter-free (non-parametric) curriculum that presumes the worst-case difficulty for each expert based on the model's loss, and (3) a static curriculum to provide a minimum base for comparison amongst curriculum-based learning strategies and lack thereof. Our experiments on two benchmark datasets with distinct distributions of teams over skills showed that our parameter-free curriculum improved the performance of non-variational models across different domains, outperforming its parametric counterpart, and the static curriculum was the poorest. Moreover, among neural models, variational models obtain little to no gain from our proposed curricula, urging further research on more effective curricula for them. The code to reproduce our experiments is publically available at `https://anonymous.4open.science/r/OpeNTF-CL`.

1 INTRODUCTION

As modern tasks' complexities surpass the capacity of individual experts, collaborative teams of experts are employed to interact interdependently and adaptively toward accomplishing a common goal [45]. Recommending teams of experts whose success is *almost surely* guaranteed has been a surge of research interest in many disciplines for years, including psychology [25], the science of team science (scits) [68], management [3], medicine [56], and industrial and mechanical engineering [43]. A rich body of various computational methods, from operations research [4, 13, 32, 57, 62, 66, 69], social network analysis [33, 55], and more recently, machine learning [10, 18, 47, 48] have been proposed to replace the tedious, error-prone, and suboptimal manual search by a human selector, who has hidden personal and societal biases [51], falls short for an overwhelming number of experts, and fails to consider a multitude of criteria to optimize simultaneously [6]. Specifically, neural models have been proposed to learn the distributions of experts and their skill sets in the context of successful teams in the past to draw future successful teams. Due to the iterative and online learning procedure, and availability of training datasets, such models brought state-of-the-art efficacy while enhancing efficiency [10, 11, 44, 47, 49].

Table 1: Summary of our findings. Black vs. gray show strong vs. conservative positive (✓) or negative (✕) answers.

			our proposed curricula		
		neural model	static (sc)	parametric (pc)	non-parametric (npc)
dataset	dblp	non-variational	✕	✕	✓
		variational	✓ (gray)	✓ (gray)	✓ (gray)
	imdb	non-variational	✕	✕	✓
		variational	✕ (gray)	✕	✕

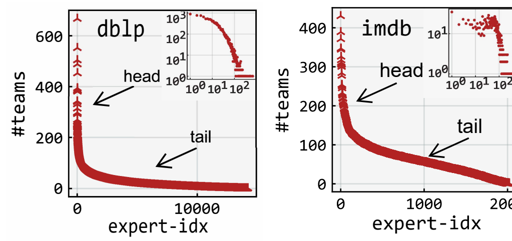

Figure 1: Long-tail distribution of experts in dblp and imdb.

Neural team recommendation models, by and large, frame the team recommendation problem as a *multilabel* Boolean classification task, learning to recommend a subset of experts whose collaborations for a predefined set of required skills yield success. Each expert is mapped to a label and would be recommended if their class's prediction probability is close to one. Such models, however, suffer from popularity bias; that is, they overfit popular experts when recommending teams and, hence, result in discrimination and reduced visibility for *non*-popular ones. Not unexpectedly, the model's popularity bias is due to the biased training datasets wherein few dominant experts (minority popular) have been disproportionately distributed over many teams compared to the sparse long-tailed distribution of many experts (majority non-popular), as shown in Figure 1 for two well-known benchmark datasets in the team recommendation literature. Moreover, existing neural models are trained based on the standard learning strategy, i.e., randomly shuffled training samples of teams, overlooking the different levels of difficulties in observing and learning from non-popular experts versus popular ones in such biased training datasets, which discounts their recommendation quality.

In this paper, we propose to leverage curriculum-based learning strategies that provide an order between experts from the *easy* popular experts to the *hard* non-popular ones to overcome the neural models' popularity bias. We propose two dynamic self-paced curricula that learn experts' difficulty level (easy popular vs. difficult non-popular) in response to a neural model's loss *during* learning procedure. In contrast to dynamic curricula, we also set up a static curriculum wherein experts are ordered statically *prior* to learning and the order remains constant during training, oblivious to the model's loss. Curriculum-based learning strategies have been successfully employed in a wide range of machine learning tasks [14, 17, 23, 24, 26, 28, 39, 46, 65, 71], yet, to our knowledge, their application in team recommendation has not been explored before our study.

Our experiments on two large-scale benchmark datasets with varied distributions of teams over skills in a host of variational and non-variational models, as summarized in Table 1, demonstrated the consistent synergistic effect of our non-parametric curriculum for non-variational models across different domains, outperforming its parametric counterpart, and the static curriculum is the poorest. For variational models, however, the gain from our proposed curricula was marginal and only in one domain (`dblp`), urging further research on curricula for them. In summary, our contributions are:

(1) We propose dynamic curricula to empower neural team recommenders sifting through *easy* popular and *hard* non-popular experts to mitigate popularity bias, which is a novel and inventive step in team recommendation literature.
(2) We present a parametric curriculum, which assigns a learnable parameter to each expert to asses their difficulty levels during training, and a parameter-free curriculum that presumes the worst-case difficulty for each expert based on model's feedback, along with a static curriculum to provide a minimum base for comparison amongst curriculum-based learning strategies and lack thereof.
(3) We demonstrate the efficacy of our parametric-free curriculum in boosting the efficacy and efficiency of non-variation neural models across two large-scale benchmark datasets from various domains with distinct distributions of skills in teams.

2 RELATED WORKS

The related works to this paper are primarily centred around (1) team recommendation and (2) curriculum learning.

2.1 Team Recommendation

Team recommendation methods can be distinguished based on the way optimizations are performed: (1) search-based, where the search for an almost surely successful team (optimum team) is carried over *all* subsets of experts using operation research techniques (e.g., integer programming) [2, 7, 9, 58, 67] or over *all* subgraphs of the expert network using graph-based methods [19, 33, 41, 55], (2) reinforcement-based [16, 40, 70], where team recommendation processes are emulated for team sports or online multiplayer games through trial and error in a multi-agent environment using neural-based policy estimators where autonomous experts learn to negotiate and form teams to perform a task that an expert cannot complete alone, and 3) learning-based [18, 27, 35, 47, 48, 52], wherein all past successful team compositions are considered as training samples to learns the relationships between experts and their skills within the context of teams using neural models. Search-based and reinforcement-based methods are, however, computationally intractable dealing with mid to large-scale sets of experts in real-world applications. In contrast, learning-based methods have shown efficiency while enhancing efficacy for iterative and online learning procedures and, hence, took the stage and became canonical.

Proposed learning-based approaches are mainly neural-based including non-variational feedforward [48], variational Bayesian network [11, 27, 48], and graph neural network [35, 47, 52]. Initially, Rad et al. [48] defined team recommendation as a multilabel classification task and, as a naive baseline for a minimum level of comparison, developed a simple feedforward network with one hidden layer to map the required subset of skills in the input layer onto a subset of experts in the output layer using the standard cross-entropy loss. Rad et al. [27, 48] then proposed a variational Bayesian network to mitigate the popularity bias through uncertainty in neural model weights in the form of Gaussian distributions. In this line, Dashti et al [10] further proposed negative sampling heuristics assuming groups of experts who have little or no collaborative experience for the required subset of skills have a low chance for a successful collaboration and can be considered as *virtually un*successful teams. Given that popular experts were dominant in the training datasets, Dashti et al. presume that groups of popular experts are more likely to be selected as negative samples of teams. Successfully as they are, the primary focus of Dashti et al. and Rad et al. was the maximization of the efficacy by tailoring the recommended experts for a team to the required skills only, overlooking to substantiate whether the higher efficacy comes with mitigation of popularity bias.

Sapienza et al.[52] were the first to use a graph neural network in the form of an autoencoder for team recommendation in online multiplayer games. Later, Rad et al.[47] proposed to transfer dense vector representations of skills for the input of variational Bayesian neural network from a heterogeneous graph whose nodes are teams, experts, skills, and locations and edges connect experts who have collaborated in a team residing in a location using Dong et al.'s `metapath2vec` [12] and obtained the state-of-the-art performance. More recently, Kaw et al. [35] employed deep graph infomax [61], a graph convolution network with attention layer as an encoder, to learn more effective vector representations of skills in less training epochs owing to the convolutional architecture and contrastive learning procedure.

Nonetheless and despite few efforts [10, 27, 48], existing neural team recommendation models still withhold extreme popularity bias due to their learning strategy that overlooks the different levels of difficulties in recommending non-popular experts vs. popular ones in a biased dataset. In this paper, we aim to bridge the gap via curriculum learning strategies.

2.2 Curriculum Learning

A curriculum for a machine learning model can be static and predefined, or dynamically adjusted during model learning. Predefined static curricula have been effectively employed in natural language processing and computer vision. For instance, in machine translation, Platanios et al. [46] defined a static curriculum based on the length of a sentence, assuming longer sentences are more difficult, whereas Kocmi et al. [36] considered a sentence with more rare words are more difficult. In computer vision, Guo et al. [24] defined static curriculum based on the distribution of objects in an image; the more objects in the image, the more potential for misclassification, and hence, the more difficult for the model. Simple and straightforward, static curricula, however, overlook the feedback from the model during learning. A training sample that appears to be difficult for a human observer might be easy for a model or vice versa, as shown in the image classification task [29]. Moreover, the complexity of training samples evolves from the model's perspective as it undergoes learning iterations and accumulates knowledge, and what was initially difficult becomes easier for the model after.

Unlike static curricula, dynamic curricula reorganize the sequence of training data based on their varying levels of difficulty in response to the model's losses *during* the learning process. Han et al. [28] and others [65, 71] proposed using an external pretrained model as a *teacher* to evaluate the difficulty of training samples for a *student* model based on its loss value during learning. Teacher-student approaches, however, suffer from a lack of self-sufficiency due to their reliance on external, usually large teacher models. In contrast, Kumar et al. [39] introduced a *self-paced* curriculum, where the model autonomously selects difficult or easy samples based on its own loss as feedback. To achieve a task-agnostic *universal* dynamic curriculum, Saxena et al. [53] proposed learnable parameters for the difficulty level of each individual data point. These parameters dynamically adjust the importance of samples during training via gradient descent, facilitating a *differentiable* curriculum. Parametric curricula, however, have overheads like more parameters to learn overall and more learning epochs, which would come at the cost of overfitting the model's original parameters. To fill the gap, Castells et al. [5] proposed a parameter-free (non-parametric) dynamic curriculum that can be integrated with various tasks and loss functions without altering the training process.

Despite extensive research on curriculum learning [42, 54, 64], *no* work has addressed neural models' popularity bias using curriculum learning, in general, except that of the very recent work by Jeon et al. [31] to address cold-start bundle recommendation. Moreover, *no* work has applied curriculum learning to team recommendation. To the best of our knowledge, we are the first to bridge curriculum-based learning strategies to address popularity bias in the context of team recommendation.

3 TASK FORMULATION

Given a set of skills $\mathcal{S} = \{i\}$ and a set of experts $\mathcal{E} = \{j\}$, a successful team of experts $\mathbf{e} \subseteq \mathcal{E}; \mathbf{e} \neq \emptyset$ that collectively cover the skill set $\mathbf{s} \subseteq \mathcal{S}; \mathbf{s} \neq \emptyset$ is shown by $(\mathbf{s}, \mathbf{e})$, and $\mathcal{T} = \{(\mathbf{s}, \mathbf{e})\}$ is the collection of all successful teams. The team recommendation problem aims at recommending an optimal subset of experts $\mathbf{e}$ such that their collaboration in the predicted team $(\mathbf{s}, \mathbf{e})$ is successful, that is $(\mathbf{s}, \mathbf{e}) \in \mathcal{T}$, while avoiding a subset of experts $\mathbf{e}'$ resulting in $(\mathbf{s}, \mathbf{e}') \notin \mathcal{T}$. More concretely, the team recommendation problem is to find a mapping function f of parameters $\boldsymbol{\theta}$ from the powerset of skills to the powerset of experts such that $f_{\boldsymbol{\theta}} : \mathcal{P}(\mathcal{S}) \rightarrow \mathcal{P}(\mathcal{E}), f_{\boldsymbol{\theta}}(\mathbf{s}) = \mathbf{e}$. Neural team recommender estimates $f_{\boldsymbol{\theta}}(\mathbf{s})$ using a multilayer neural network that learns, from $\mathcal{T}$, to map a vector representation of subset of skills $\mathbf{s}$, referred to as $v_{\mathbf{s}}$, to a vector representation of subset of experts $\mathbf{e}$, referred to as $v_{\mathbf{e}}$, by maximizing the posterior probability of $\boldsymbol{\theta}$ in $f_{\boldsymbol{\theta}}$ over $\mathcal{T}$, that is, $\texttt{argmax}_{\boldsymbol{\theta}}\ p(\boldsymbol{\theta}|\mathcal{T})$.

For the vector representation of subset of skills $v_{\mathbf{s}}$, neural team recommenders adopt either (1) the *occurrence* vector representation for $\mathbf{s}$, which is a Boolean vector of size $|\mathcal{S}|$, i.e., $v_{\mathbf{s}} \in \{0, 1\}^{|\mathcal{S}|}$ where $v_{\mathbf{s}}[i] = 1$ if $i \in \mathbf{s}$, and 0 otherwise, or (2) a dense low d-dimensional vector representation of $\mathbf{s}$, $d << |\mathcal{S}|$, pretrained by a graph neural network method [35, 47, 50]. In the output layer for vector representation of the subset of experts $v_{\mathbf{e}}$, neural team recommenders frame the problem as a multilabel Boolean classification task and used occurrence vector representation for $\mathbf{s}$, that is, $v_{\mathbf{e}} \in \{0, 1\}^{|\mathcal{E}|}$ where $v_{\mathbf{e}}[j] = 1$ if $j \in \mathbf{e}$, and 0 otherwise, as seen in Figure 2. Using a neural model of one hidden layer $\mathbf{h}$ of size d, without loss of generality to multiple hidden layers, with the input layer $v_{\mathbf{s}}$ and output layer $v_{\mathbf{e}}$, a neural team recommender can be formalized as,

$$\mathbf{h} = \pi(\boldsymbol{\theta}_1 v_{\mathbf{s}} + \mathbf{b}_1) \tag{1}$$

$$logits \rightarrow \mathbf{z} = \boldsymbol{\theta}_2 \mathbf{h} + \mathbf{b}_2 \tag{2}$$

$$v_{\mathbf{e}} = \sigma(\mathbf{z}) \tag{3}$$

where π is a nonlinear activation function, $\boldsymbol{\theta} = \boldsymbol{\theta}_{.} \cup \mathbf{b}_{.}$ are learnable parameters for the mapping function f, and σ is the `sigmoid` function to generate the predictions for each class/expert. During training, given a team $(\mathbf{s}, \mathbf{e})$, neural models tune the parameters $\boldsymbol{\theta}$ by maximizing the posterior probability of $\boldsymbol{\theta}$ in $f_{\boldsymbol{\theta}}$ over $\mathcal{T}$. By Bayes theorem:

$$\texttt{argmax}_{\boldsymbol{\theta}}\ p(\boldsymbol{\theta}|\mathcal{T}) \propto p(\mathcal{T}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{(\mathbf{s},\mathbf{e}) \in \mathcal{T}} p(\mathbf{e}|\mathbf{s}, \boldsymbol{\theta}) \tag{4}$$

where $p(\mathcal{T})$ is independent of optimizing the parameters $\boldsymbol{\theta}$, $p(\mathcal{T}|\boldsymbol{\theta})$ is the likelihood:

$$p(\mathbf{e}|\mathbf{s}, \boldsymbol{\theta}) = \prod_{j \in e} \sigma(\mathbf{z}[j]) \propto \sum_{j \in e} \texttt{log}\ \sigma(\mathbf{z}[j]) \tag{5}$$

where $p(\mathcal{T}|\boldsymbol{\theta})$ is the likelihood and $p(\boldsymbol{\theta})$ is the prior joint probability of weights, which is unknown. The *true* prior probability of weights $p(\boldsymbol{\theta})$ cannot be calculated analytically or efficiently sampled [22], and as such, we can assume *uniform* probability distribution over all possible real-values of $\boldsymbol{\theta}$ and proceed with maximum likelihood estimation $p(\mathcal{T}|\boldsymbol{\theta})$ [10], or estimate $p(\boldsymbol{\theta})$ by Gaussian distribution and calculate the maximum a posteriori via a variational Bayesian neural architecture [27, 35, 47, 48]. Whether maximum likelihood estimation or maximum a posteriori optimization is used to estimate f, the long tail problem in the distributions of teams over experts renders the neural model to overfit popular experts. To tackle the issue, we propose curriculum-based learning strategies for neural models without modifications to their architectures.

4 PROPOSED CURRICULA

As opposed to the random shuffling of the training sample of teams, we propose learning curricula that are cognizant of gradual moves from easy popular experts to difficult non-popular ones based on the neural model's learning loss. We propose one static and two dynamic self-paced curricula to improve the performance of neural team recommenders. While our static curriculum serves as a baseline, we opt for dynamic curricula since team recommendation finds applications in different domains, from the entertainment industry (e.g., movie production) to academia (e.g., research teams), each of which has its own specificities, making it impractical to adopt static predefined curricula for all varied domains. Further, the choice of a self-paced curriculum over student-teacher-based is motivated for its self-sufficiency with no dependency on an external large pretrained model as a teacher.

4.1 Static Curriculum

Prior to the model's learning process, we define an order among the training instances of teams based on the difficulty levels of their experts. We presume a neural model easily predicts popular experts due to their frequent participation in many teams during training but struggles to predict non-popular experts, who participate sparingly. Therefore, the difficulty level of a team can be defined based

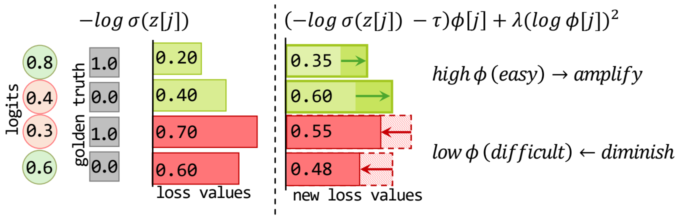

Figure 2: Standard (left) vs. dynamic curriculum (right) loss.

on the proportion of its popular experts, i.e.,

$$d((\mathbf{s},\mathbf{e}))=1-\frac{|\{j: j\in\mathbf{e}, j \text{ is popular}\}|}{|\mathbf{e}|} \tag{6}$$

where $d:\mathcal{T}\rightarrow\mathbb{R}^{[0,1]}$ is the difficulty measurer function, which calculates a fixed difficulty level per team, from `0.0` for the easiest teams whose all experts are popular to `1.0` for the most difficult teams whose all experts are non-popular. Eq. 6, however, requires defining the *popularity status* of experts, which could be controversial. To avoid varied interpretations, we follow recommender system literature [1, 15] where the popularity status of an expert can be objectively measured based on the number of teams the expert has participated in. An expert is popular if the expert participated in more than the average number of teams per expert over the entire dataset and non-popular otherwise. It is worth noting that, although an expert's participation in many teams, e.g., movies in `imdb` or research papers in `dblp`, does not necessarily indicate popularity from the people's perspectives, repetition of the expert in many training samples of teams from the neural model's perspective does.

During learning epochs, training samples of teams are selected in batches based on the proportions of easy vs. difficult teams. Initially, we manually set the proportion to favour more easy teams with many popular experts, like `90%` easy teams vs. `10%` difficult teams. As the training progresses and the model evolves, we gradually reverse the proportions to favour more difficult teams with many non-popular experts, like `10%` easy teams vs. `90%` difficult teams.

4.2 Parametric Curriculum

We further propose a parametric self-paced curriculum to dynamically estimate the difficulty (conversely, popularity) level of experts in a self-supervised manner during learning. We define a *new* set of learnable parameters for the set of experts $\mathcal{E}$ as a vector of parameters $\boldsymbol{\phi}\in\mathbb{R}^{|\mathcal{E}|}$ to learn the levels of difficulties for experts based on the neural model's loss, in parallel to learning the model's parameter $\boldsymbol{\theta}$. Initially, all of the $\boldsymbol{\phi}$ values are low, i.e., $\boldsymbol{\phi}[j]\simeq 0; \forall j\in\mathcal{E}$, assuming all experts are at the most difficult level (conversely, assuming all experts are non-popular). After the first learning batch, $\boldsymbol{\phi}$ would be updated; a *true* expert j that has been recommended correctly with a high probability value, equivalently low loss value, would be considered as an easy expert and $\boldsymbol{\phi}[j]$ would be increased toward 1. However, if the model produces a low probability value for the true expert, that expert would be difficult, and hence, $\boldsymbol{\phi}[j]$ remains low. We update the model's learning parameters $\boldsymbol{\theta}$ based on not only the recommendation losses of true experts but also the $\boldsymbol{\phi}$ values such that the model knows the easy experts sooner, and gradually moves toward the more difficult ones. From Figure 2, for a true yet difficult expert j whose $\boldsymbol{\phi}[j]$ is low, we diminish its recommendation loss for the model to update its learning parameters $\boldsymbol{\theta}$ so as to decrease the j's loss in the next learning iteration, and meanwhile, increasing $\boldsymbol{\phi}[j]$ (trying to learn expert j and make it as an easy expert). For an easy true expert j, i.e., $\boldsymbol{\phi}[j]\simeq 1$, however, we amplify its loss to avoid overfitting to the model's parameters $\boldsymbol{\theta}$.

To implement our proposed parametric loss-based curriculum, we follow the common forward pass during training for a given team $(\mathbf{s},\mathbf{e})\in\mathcal{T}$, i.e., we input the vector representation of the skill subset $\mathbf{s}$, i.e., $v_\mathbf{s}$ through the neural model in eq.1 and compute the experts' probability of being in the team in the output $v_\mathbf{e}$ after applying the `sigmoid` followed by the original loss (eq.5). However, before backpropagation of the gradients, we further learn the difficulty levels of experts based on the losses generated for the current sample team as model's feedback not only for the model's parameters $\boldsymbol{\theta}$ but also for $\boldsymbol{\phi}$. We modify eq. 5 to scale up (down) the gradients of losses for difficult experts having low (high) $\boldsymbol{\phi}$:

$$l((\mathbf{s},\mathbf{e}),\boldsymbol{\phi})=\sum_{j\in\mathbf{e}}-\log\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]}) \tag{7}$$

If we treat all the experts equally easy by setting the $\boldsymbol{\phi}=[1,1,...,1]$, eq.7 recovers the original loss. The gradient of the loss with respect to $v_\mathbf{e}$ to optimize the model parameters $\boldsymbol{\theta}$ will be:

$$\frac{\partial l}{\partial\mathbf{z}[j]}=\frac{-1}{\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]})}\cdot\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]})\cdot(1-\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]}))\cdot\frac{1}{\boldsymbol{\phi}[j]}=\frac{1}{\boldsymbol{\phi}[j]}\cdot(\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]})-1) \tag{8}$$

During training, we also need to update the difficulty levels of experts $\boldsymbol{\phi}$. The gradient of the loss with respect to $\boldsymbol{\phi}$:

$$\begin{aligned}\frac{\partial l}{\partial\boldsymbol{\phi}[j]}&=\frac{-1}{\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]})}\cdot\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]})\cdot(1-\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]}))\cdot\frac{-\mathbf{z}[j]}{\boldsymbol{\phi}[j]^2}\\&=\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]^2}\cdot(1-\sigma(\frac{\mathbf{z}[j]}{\boldsymbol{\phi}[j]}))\end{aligned} \tag{9}$$

The parametric curriculum, however, introduces additional complexity to the training process, requiring more learning epochs to accommodate the extra parameters $\boldsymbol{\phi}$. The size of $\boldsymbol{\phi}$ is directly proportional to the number of experts in the training set, hence, the parametric curriculum may fall short of training efficiency and inference efficacy trade-off when compared to the static curriculum or the randomly shuffled training strategy when dealing with a large pool of experts. In Section 5, **RQ4**, we revisit this challenge when comparing each of our proposed curricula.

4.3 Non-parametric Curriculum

Alternatively, we propose a non-parametric curriculum, which requires no set of parameters $\boldsymbol{\phi}$ for learning the experts' difficulty levels. Similar to the parametric curriculum, our non-parametric curriculum scales up (down) the loss:

$$l((\mathbf{s},\mathbf{e}),\boldsymbol{\phi}[j])=(\overbrace{-\log\sigma(\mathbf{z}[j])}^{\text{original loss in eq. 5}}-\tau)\boldsymbol{\phi}[j]+\lambda(\log\boldsymbol{\phi}[j])^2 \tag{10}$$

where τ is a threshold that separates easy experts from difficult ones based on their respective loss, $\lambda>0$ is a regularization hyperparameter which controls the severity of new loss' effect on the normal loss, and $\boldsymbol{\phi}$ is the learnable parameters for levels of difficulties for

Table 2: Statistics of the raw and preprocessed datasets.

	dblp		imdb	
	raw	filtered	raw	filtered
#teams	4,877,383	99,375	507,034	32,059
#unique experts	5,022,955	14,214	876,981	2,011
#unique skills	89,504	**29,661**	28	**23**
avg #expert per team	3.06	3.29	1.88	3.98
avg #skill per team	8.57	9.71	1.54	1.76
avg #team per expert	2.97	**23.02**	1.09	**62.45**
avg #skill per expert	16.73	96.72	1.59	10.85
#team w/ single expert	768,956	**0**	322,918	**0**
#team w/ single skill	5,569	56	315,503	15,180

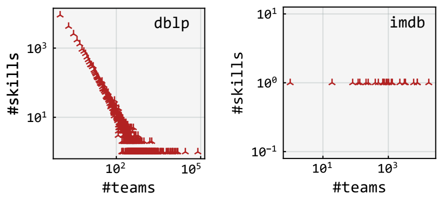

Figure 3: Long-tail vs. uniform distribution of skills in dblp and imdb, respectively.

expert set $\mathcal{E}$. However, as opposed to the parametric curriculum, we set the difficulty level of an expert to its worst case for a loss value, cancelling the learning the difficulty level parameters $\boldsymbol{\phi}$ for experts', reducing the computational complexity of the training. Therefore, the curriculum-based loss becomes:

$$l((\mathbf{s},\mathbf{e}),\boldsymbol{\phi}[j]) = l((\mathbf{s},\mathbf{e}),\boldsymbol{\phi}^*[j]) \quad (11)$$

where $\boldsymbol{\phi}^*$ would only depend on the value of the original loss and has a closed-form solution [5]:

$$\begin{aligned}\boldsymbol{\phi}^*_{l((\mathbf{s},\mathbf{e}),\boldsymbol{\phi}[j])} &= \mathtt{argmin}_{\boldsymbol{\phi}[j]} l((\mathbf{s},\mathbf{e}),\boldsymbol{\phi}[j]) \\ &= e^{-\omega(\frac{1}{2}\max(\frac{-2}{e},\frac{-\log\,\sigma(\mathbf{z}[j])-\tau}{\lambda}))}; j \in \mathbf{e}\end{aligned} \quad (12)$$

where ω is the `product logarithm` (Lambert omega) function [8]. During a learning iteration, we calculate eq.12 as the difficulty level of expert $j \in \mathbf{e}$ for the input $(\mathbf{s},\mathbf{e})$ from the original loss. We then propagate the gradients of eq.10 to the model's parameters $\boldsymbol{\theta}$ assuming the result of eq.12 as a constant.

5 EXPERIMENT

In this section, we seek to answer the following research questions:
RQ1: Can curriculum-based learning strategies improve the *efficacy* of neural team recommenders via mitigating popularity bias?
RQ2: Does curriculum-based learning strategy improve the *efficiency* of neural team recommenders while enhancing efficacy?
RQ3: Is the impact of curricula consistent across datasets from various domains with distinct statistical distributions?
RQ4: Among our proposed curricula, which curriculum has been effective the most (least)?

5.1 Datasets

We evaluate our proposed curricula on two well-known large-scale benchmark datasets in team recommendation literature, including dblp [37, 41, 59], consisting of computer science publications and imdb [30, 33, 34] consisting of movies. In dblp, we see each publication as a successful team whose authors are the experts and fields of studies are the set of skills. In imdb, we consider each movie as a successful team for it has been produced, experts are the cast and crew, and the movies' genres are the teams' skills. Regarding imdb, contrary to movie recommender systems or movie review analysis, wherein the goal is to recommend a movie, herein, we aim to recommend a team of casts and crews to *produce* a movie. We filtered out singleton teams as well as experts who have participated in very few teams. Earlier from Figure 1, we observe the distributions of teams over experts have a very long tail of non-popular experts in both datasets; many experts (researchers in dblp and cast and crew in imdb) have participated in very few teams (papers in dblp and movies in imdb). With respect to the set of skills, from Figure 3, while dblp suffers further from the long-tailed distribution of skills in teams, imdb follows a more fair distribution; imdb have a limited number of skills (genres) all of which are generally employed by many movies. Also, Table 2 reports additional point-wise statistics on the raw and filtered datasets.

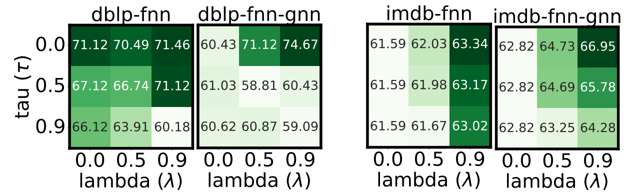

Figure 4: Grid search on non-parametric curriculum hyperparameters on dblp (left) and imdb (right) based on aucroc.

Popularity Label. As for our static curriculum, we consider an expert popular if she has participated in more teams than the average number of teams per expert. From Table 2, experts who participate in more than `23.02` teams in the dblp and `62.45` teams in imdb are considered popular, while the rest are assumed to be non-popular.

5.2 Baselines and Hyperparameters

We compare the impact of our proposed curricula on improving the performance of two neural model alternatives, especially when the training datasets are heavily biased toward popular experts: (1) non-variational feedforward neural network (`fnn-*`) [10] and (2) variational Bayesian neural network [27, 48] (`bnn-*`). Both models include a single hidden layer of size d=128, `leaky relu` is the activation function for the hidden layer, and `Adam` is the optimizer. For the input layer, we used sparse occurrence vector representations of skills of size $|\mathcal{S}|$ as well as gnn-based pretrained dense vector representations of size $d = \mathtt{100}$ [12] (`*-gnn-*`). We trained baselines with and without our proposed curricula ({`*-sc`, `*-pc`, `*-npc`} vs. `*-std`). We set $\tau = \mathtt{0}$ and $\lambda = \mathtt{0.9}$ for the non-parametric curriculum after a grid search for best settings across datasets and neural baselines in terms of `aucroc`, as seen in Figure 4. The regularization hyperparameter λ, which controls the degree of impact from curriculum strategy on the model's original loss, becomes more effective at higher values as it increases the penalty on the model's loss, thus mitigating the risk of overfitting to easy popular experts. Conversely, τ, the threshold that separates easy experts from difficult ones based on their respective losses, yields the best results in lower values, posing stricter criteria for classifying an expert as easy. Finally, to have a minimum level of comparison, we also add a model that randomly assigns experts to a team (`random`). In total, we compare `16 + 1` baselines.

Table 3: Comparative results of 5-fold neural models with curricula and lack thereof on the test sets for `dblp` and `imdb`. ‡ refer to statistically significant improvements (`*-pc` and `*-npc` vs. `*-std`) when *p*-value < 0.01. Bold and underlined numbers are column-wise highest and second-highest *per neural architecture*, respectively.

dblp	%precision		%recall		%ndcg		%map		
	@2	@10	@2	@10	@2	@10	@2	@10	%aucroc
random	0.0100	0.0200	0.0100	0.0600	0.0200	0.0400	0.0100	0.0200	49.9200
fnn-std [10]	0.2994	0.1799	0.1742	0.5300	0.2988	0.4024	0.1300	0.2049	68.3604
fnn-sc	0.0188	0.0164	0.0121	0.0509	0.0182	0.0338	0.0086	0.0154	50.1144
fnn-pc	0.2407	0.1751	0.1403	0.5134	0.2568	0.3884	0.1157	0.1981	61.4063
fnn-npc	**0.5031‡**	**0.3633‡**	**0.2929‡**	**1.0643‡**	**0.5062‡**	**0.7770‡**	**0.2220‡**	**0.3798‡**	**71.4635**
fnn-gnn-std	0.3975	0.2252	0.2318	0.6450	0.3967	0.5400	0.1745	0.2895	74.7247
fnn-gnn-sc	0.0188	0.0247	0.0110	0.0739	0.0197	0.0473	0.0089	0.0203	50.0975
fnn-gnn-pc	0.4721‡	0.2007	0.2712	0.5822	0.4795	0.5182	0.2094	0.2840	74.6674
fnn-gnn-npc	**0.5182‡**	**0.3792‡**	**0.2991‡**	**1.0912‡**	**0.5269‡**	**0.8005‡**	**0.2277‡**	**0.3898‡**	**75.9575**
bnn-std [27, 48]	0.0285	0.0272	0.0179	0.0856	0.0289	0.0586	0.0138	0.0288	50.0227
bnn-sc	0.0356	0.0266	0.0225	0.0837	0.0357	0.0595	0.0171	0.0299	49.8803
bnn-pc	0.0268	**0.0300**	0.0166	**0.0954**	0.0272	**0.0630**	0.0133	**0.0306**	**50.0703**
bnn-npc	**0.0369**	0.0223	**0.0233**	0.0698	**0.0388**	0.0534	**0.0188**	0.0278	50.0220
bnn-gnn-std [47]	0.0210	0.0263	0.0120	0.0811	0.0215	0.0509	0.0096	0.0217	50.0061
bnn-gnn-sc	**0.0356**	0.0241	**0.0234**	0.0776	**0.0366**	0.0576	**0.0184**	**0.0306**	49.9698
bnn-gnn-pc	0.0293	0.0247	0.0180	0.0773	0.0284	0.0526	0.0126	0.0254	**50.0863**
bnn-gnn-npc	0.0310	**0.0309**	0.0195	**0.0960**	0.0308	**0.0624**	0.0144	0.0278	49.9071

imdb	%precision		%recall		%ndcg		%map		
	@2	@10	@2	@10	@2	@10	@2	@10	%aucroc
random	0.1700	0.1800	0.0800	0.4500	0.1700	0.3100	0.0600	0.1200	49.8800
fnn-std [10]	0.9526	0.7750	0.4090	1.7625	0.9684	1.3405	0.3249	0.5839	58.9675
fnn-sc	0.2118	0.2093	0.1027	0.5234	0.2165	0.3692	0.0794	0.1534	50.2131
fnn-pc	0.9240	0.6878	0.3927	1.5619	0.9205	1.2018	0.3096	0.5329	54.8621
fnn-npc	**0.9915‡**	**0.8840‡**	**0.4415‡**	**2.0059‡**	**1.0056‡**	**1.5104‡**	**0.3595 ‡**	**0.6820‡**	**63.0213**
fnn-gnn-std	0.7319	0.6582	0.3376	1.5096	0.7449	1.1390	0.2645	0.5111	62.3177
fnn-gnn-sc	0.2326	0.2151	0.1154	0.5291	0.2429	0.3875	0.0915	0.1659	50.3146
fnn-gnn-pc	0.6645	0.6214	0.2991	1.3973	0.6645	1.0456	0.2277	0.4565	64.7053
fnn-gnn-npc	**0.8150‡**	**0.7408‡**	**0.3555‡**	**1.6556‡**	**0.8162‡**	**1.2426‡**	**0.2686‡**	**0.5317‡**	**66.7753**
bnn-std [27, 48]	**0.2544**	**0.2481**	**0.1284**	0.6062	**0.2626**	**0.4375**	**0.1016**	**0.1859**	49.7615
bnn-sc	0.1890	0.2450	0.0993	**0.6145**	0.1857	0.4070	0.0721	0.1630	50.0441
bnn-pc	0.1791	0.2149	0.0899	0.5177	0.1785	0.3553	0.0680	0.1431	**50.1223**
bnn-npc	0.2076	0.1983	0.0986	0.4871	0.2088	0.3488	0.0747	0.1462	49.9923
bnn-gnn-std [47]	**0.1973**	0.2030	**0.1005**	0.5029	**0.2008**	0.3557	**0.0772**	0.1495	**50.0626**
bnn-gnn-sc	0.1890	**0.2471**	0.0924	**0.6003**	0.1866	**0.4062**	0.0682	**0.1595**	50.0547
bnn-gnn-pc	0.1661	0.2424	0.0809	0.5944	0.1614	0.3922	0.0587	0.1508	50.0527
bnn-gnn-npc	0.1869	0.1900	0.0988	0.4896	0.1869	0.3393	0.0733	0.1424	49.9541

5.3 Evaluation Strategy and Metrics

Models' Efficacy. We randomly select 15% of teams for the test set and perform 5-fold cross-validation on the remaining teams for model training using the proposed curricula over `10` epochs that results in one trained model per each fold. Given a team $(\mathbf{s}, \mathbf{e})$ from the test set, a trained model predicts the membership probability of *all* experts in the team at the output layer. We select a subset of experts $\mathbf{e}'$ with the top-k highest probabilities as the recommended team of size k and compare it with the observed subset of experts $\mathbf{e}$ and report the average performance of models on all folds in terms of classification metrics including `precision` and `recall` at $k \in \{2, 10\}$. In contrast to top-k that enforces a fixed size for the recommended teams, we can select a probability threshold above which an expert becomes a member of the recommended team $\mathbf{e}'$. We report the area under the receiver operating characteristic (`aucroc`) as an overall performance indicator for the range of increasing probability thresholds from `0.0` to `1.0`.

Classification metrics fail to consider the ranks of the correct experts among the top-k. To evaluate a neural model more rigorously, we additionally use ranking metrics from information retrieval [60], including normalized discounted cumulative gain (`ndcg`), and mean average precision (`map`) at top-$k \in \{2, 10\}$. Ranking metrics penalize a neural model if a correct expert of the test team $\mathbf{e}$ is positioned at the lower part of the top-k recommended experts $\mathbf{e}'$.

Models' Efficiency. We evaluate neural models at *each* learning epoch on the test set to show whether our proposed curricula improve the efficiency of neural models with fewer learning epochs while achieving higher inference efficacy.

Models' Popularity Bias. We used ndkl [20, 21, 63], which measures the *divergence* of the actual distribution of non-popular experts in the top-k ranked list of recommendations (i.e., the proportions of non-popular vs. popular experts) from the *desired* distribution using Kullback–Leibler [38], and the lower divergence the better (↓), with being 0 in the *ideal* equal distributions. We report ndkl at top-k = 10 and compared the results of neural models that utilize curricula and lack thereof at an increasing range of desired distribution (proportion) of non-popular experts {0.1, 0.2, ..., 0.9}.

6 RESULTS

Foremost, we acknowledge that baseline methods achieve relatively low metric values for practical applications of team recommendation, primarily due to the simplicity of the model architectures and the small number of learning epochs for our limited access to computational resources; metric values are reported in % for ease of readability and comparison. Herein, our main goal is to study the synergistic effects of our proposed curricula in scaling up neural models' performance via mitigating popularity bias. With larger multi-layer architectures, better results would be expected.

RQ1: Efficacy Improvement via Proposed Curricula: From Table 3, we can observe that the answer depends on the underlying model architecture and training dataset. As seen, our *non*-parametric curriculum demonstrates a consistent statistically significant performance improvement in comparison with the lack thereof for the *non*-variational models (fnn-npc and fnn-gnn-npc) in both dblp and imdb datasets across all metrics. However, our parametric curriculum has shown little to no improvement for them (fnn-*-pc), and the static curriculum is the poorest (fnn-*-sc) across datasets. Indeed, the standard learning strategy based on the randomly shuffled dataset is the runner-up (fnn-*-std).

Unlike non-variational models (fnn-*), we cannot observe a consistent trend for performance improvements of variational models (bnn-*) by our proposed curricula across datasets. In dblp, while curriculum learning strategies generally yield improvements for variational models, yet no specific curriculum dominates. For instance, with occurrence vector representation of skills, the non-parametric curriculum (bnn-npc) is the best at team size of $k = 2$, but at $k = 10$, the parametric curriculum becomes the best. However, with gnn-based skill vectors, the static curriculum (bnn-gnn-sc) is the winner at $k = 2$ while non-parametric curriculum (bnn-gnn-npc) pulls ahead at $k = 10$. In imdb, we see the worst trend where no curriculum has been successful except for variational gnn-based models at $k = 10$ where the static curriculum could improve the performance.

While answering **RQ1**, we could reproduce Rad et al. [47]'s work about the performance gain of the gnn-based dense vectors of skills for non-variational models (fnn-gnn-std) compared to the sparse occurrence vectors (fnn-std) in dblp dataset, where the number of skills is large ($|\mathcal{S}| = 29,661$). Notably, Table 3 shows that the gnn-based models also gain more from our proposed curricula, maintaining the upward improvement trend. However, we were unable to reproduce Rad et al. [47]'s work in imdb dataset; that is, gnn-based vector representation of skills performs poorly compared to occurrence vectors. The reason relates to the relatively low number of skills in imdb ($|\mathcal{S}| = 23$), where the occurrence vector representation is already low-dimensional and the sparsity is negligible. Indeed, gnn-based vectors of size $d = 100$ are of higher dimension and discount the models' performance in imdb. From Table 3, gnn-based models also gain less from our proposed curricula, further cementing this reason.

To show whether the performance improvement of the non-parametric curriculum is indeed due to mitigating popularity bias, from Figure 5 (left), we observe that non-parametric curriculum consistently yields less divergence for non-variational models (fnn-*-npc), hence, better distribution for the desired ratio equal or above 50% non-popular experts, where we expect a balance or even more non-popular experts vs. popular ones in the recommended teams across dblp and imdb. For instance, at the extreme desired ratio of 0.9 where we expect to observe 90% non-popular experts, not unexpectedly, all baselines fall short of reaching the desired distribution, yet our non-parametric curriculum obtains closer distribution (lower ndkl). In imdb, while static curriculum (*-sc) yields the best result across almost all increasing ratios of ndkl, this has come at the cost of a substantial drop in the fnn's efficacy, as seen in Table 3, yielding its poor performance overall. The parametric curriculum generally performs the same or worse than the standard learning strategy, falling short of mitigating popularity bias either. From Figure 5 (right), the variational models generally perform similarly with and without a curriculum. The only marginal improvement is by the non-parametric curriculum for gnn-based variational model in dblp dataset (dblp-bnn-gnn), yielding less divergence but only for the desired ratios below 0.3.

Overall, our answer to **RQ1** is affirmative as long as the underlying model structure is non-variational. However, for variation models, we are reserved in giving a positive response, calling for further research on developing curricula for variational models.

RQ2: Efficiency Improvement via Proposed Curricula: From Figure 6 (left), we can observe that our non-parametric curriculum in fnn-*-npc outperforms standard and other curriculum-based learning strategies in fewer number of training epochs for non-variational models in imdb in terms of ndcg@10, and the standard learning strategy (*-std) is the runner-up. We cannot observe a similar trend for parametric curricula, which further explains that the parametric curriculum, in fact, needs more epochs to learn not only the model's parameters (θ) but also the parameters for experts' difficulty levels ($\boldsymbol{\phi}$). From the figure, the static curriculum strategy (*-sc), as already evidenced in Table 3, shows the poorest performance. From Figure 6 (right) and with respect to the training efficiency of variational models, we clearly observe no gain. In sum, our answer to **RQ2** is affirmative but for the non-parametric curriculum for non-variational models.

RQ3: Consistency of Curricula across Datasets: From Tables 3, we see that non-parametric curriculum's synergy to the performance of feedforward neural models (fnn-*-npc) is agnostic to the distributions of experts and skills in teams. More concretely, for dblp with a long-tailed distribution of skills in teams and imdb with a limited set of skills that are employed almost uniformly by teams, we can see that the results of the non-parametric curriculum are always superior for all the metrics compared to the standard learning strategy in non-variational models (fnn-*). Variational models' benefit from the proposed curricula, however, depends on the distribution of skills over teams. Contrary to dblp where

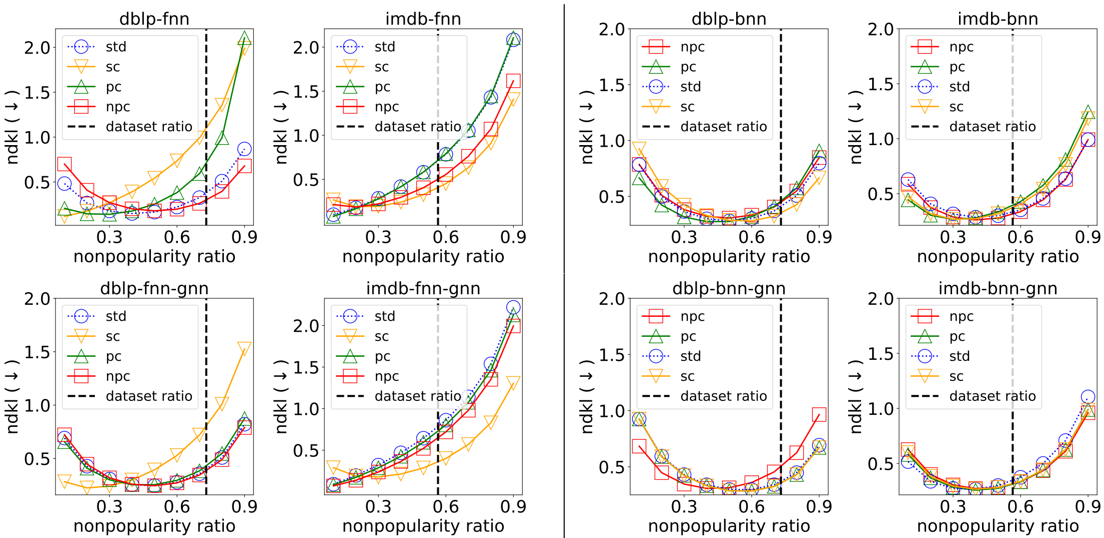

Figure 5: The comparative results of non-variational models `fnn-*` (left) and variational ones `bnn-*` (right) in mitigating popularity bias in terms of `ndkl@10` across increasing ratios of non-popular experts in recommended teams. The lower `ndkl` indicates less divergence, and hence, the better.

curriculum-based learning strategies marginally improved the results of the variational Bayesian model (`bnn-*-pc` and `bnn-*-npc` vs. `bnn-*-std`), we can observe little to no improvement in `imdb`. In conclusion, the answer to **RQ3** is positive for non-parametric curriculum utilized by non-variational models. However, for variational models, the answer is negative for our proposed curricula.

RQ4: Comparing Curriculum-based Strategies: As seen in Table 3, the non-parametric curriculum has consistently performed better compared to the parametric one. As previous studies showed, while parametric curricula have been successfully applied in challenging classification tasks, they add running overheads; they demand more learning epochs for learning the parameters associated with the difficulty (popularity) levels of the experts, which would come at the cost of overfitting for the model's parameters [5]. Additionally, as such parameters are assigned on a per-expert basis, the parametric curriculum yields more model complexity for a larger pool of experts, raising scalability concerns in real-world applications. However, our non-parametric curriculum eliminates the need for learning extra learnable parameters by taking conservative steps and choosing the worst difficulty level for an expert for a given loss value. Our static curriculum is the least effective, as it overlooks the feedback from the model during learning and cannot adjust its predefined difficulty measurer.

In terms of neural model architecture, Table 3 further shows that our proposed curricula have been less effective for variational Bayesian models (`bnn-*`) compared to non-variational neural models (`fnn-*`), which can be attributed to the probabilistic parameters of the variational models that confound learning or identifying the difficulty (conversely, popularity) of an expert during learning iterations for the same input team. An interesting avenue of research is to develop variational parametric curricula to estimate the levels of experts' difficulty (popularity) based on a probability distribution and study the impact when utilized by variational models. Overall, our findings show that our proposed non-parametric curriculum

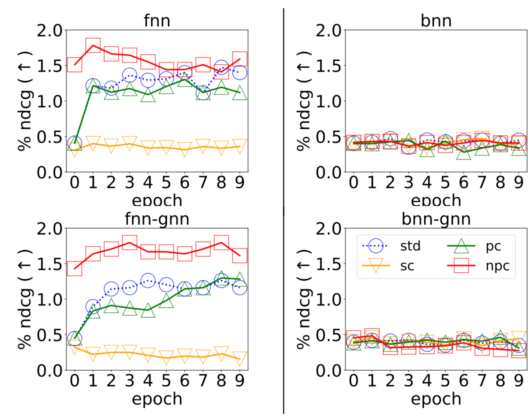

Figure 6: Training efficiency vs. inference efficacy for non-variational (left) and variational (right) in `imdb`.

benefits non-variational neural models in terms of efficacy and efficiency by addressing the popularity bias.

7 CONCLUDING REMARKS

In this paper, we proposed one static and two dynamic loss-based curricula to improve the efficacy and efficiency of neural team recommendation models in the presence of popularity bias. Our experiment, when performed on two large-scale datasets with distinct distributions of teams over skills and experts, shows that our proposed dynamic non-parametric curriculum improves the performance of non-variational neural models for team recommendation via mitigating popularity bias, surpassing other baselines across datasets. Variational models, which learn probabilistic weights, however, render the application of static and dynamic curricula moot and ineffective. For future work, we seek to design curricula for variational Bayesian neural architectures. We also aim to expand our testbed to other domains, including `uspt` collection of patents with similar distribution to `dblp`, and `github` collection of software repos with similar distribution to `imdb`.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward C. Malthouse. 2021. User-centered Evaluation of Popularity Bias in Recommender Systems. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2021, Utrecht, The Netherlands, June, 21-25, 2021*, Judith Masthoff, Eelco Herder, Nava Tintarev, and Marko Tkalcic (Eds.). ACM, 119–129. https://doi.org/10.1145/3450613.3456821

[2] Turkay Dereli Adil Baykasoglu and Sena Das. 2007. PROJECT TEAM SELECTION USING FUZZY OPTIMIZATION APPROACH. *Cybernetics and Systems* 38, 2 (2007), 155–185. https://doi.org/10.1080/01969720601139041 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01969720601139041

[3] Bernard K Baiden and Andrew DF Price. 2011. The effect of integration on project delivery team effectiveness. *International Journal of Project Management* 29, 2 (2011), 129–136.

[4] Manoel B. Campêlo, Tatiane Fernandes Figueiredo, and Ana Silva. 2020. The sociotechnical teams formation problem: a mathematical optimization approach. *Ann. Oper. Res.* 286, 1 (2020), 201–216. https://doi.org/10.1007/s10479-018-2759-5

[5] Thibault Castells, Philippe Weinzaepfel, and Jérôme Revaud. 2020. SuperLoss: A Generic Loss for Robust Curriculum Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/2cfa8f9e50e0f510ede9d12338a5f564-Abstract.html

[6] Shi-Jie (Gary) Chen and Li Lin. 2004. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Trans. Engineering Management* 51, 2 (2004), 111–124. https://doi.org/10.1109/TEM.2004.826011

[7] Shi-Jie Chen and Li Lin. 2004. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Transactions on Engineering Management* 51, 2 (2004), 111–124. https://doi.org/10.1109/TEM.2004.826011

[8] Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. 1996. On the Lambert W function. *Advances in Computational mathematics* 5 (1996), 329–359.

[9] Michelle Craig, Diane Horton, and François Pitt. 2010. Forming Reasonably Optimal Groups: (FROG). In *Proceedings of the 2010 ACM International Conference on Supporting Group Work* (Sanibel Island, Florida, USA) *(GROUP '10)*. Association for Computing Machinery, New York, NY, USA, 141–150. https://doi.org/10.1145/1880071.1880094

[10] 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*. ACM, 3908–3912. https://doi.org/10.1145/3511808.3557590

[11] Arman Dashti, Karan Saxena, Dhwani Patel, and Hossein Fani. 2022. OpeNTF: A Benchmark Library for Neural Team Formation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. ACM, 3913–3917. https://doi.org/10.1145/3511808.3557526

[12] 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

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

[14] Rasheed el-Bouri, David W. Eyre, Peter J. Watkinson, Tingting Zhu, and David A. Clifton. 2020. Student-Teacher Curriculum Learning via Reinforcement Learning: Predicting Hospital Inpatient Admission Location. In *ICML 2020, 13-18 July 2020 (Proceedings of Machine Learning Research, Vol. 119)*. 2848–2857.

[15] Mehdi Elahi, Danial Khosh Kholgh, Mohammad Sina Kiarostami, Sorush Saghari, Shiva Parsa Rad, and Marko Tkalcic. 2021. Investigating the impact of recommender systems on user-based and item-based popularity bias. *Inf. Process. Manag.* 58, 5 (2021), 102655. https://doi.org/10.1016/J.IPM.2021.102655

[16] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. 2023. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. http://papers.nips.cc/paper_files/paper/2023/hash/764c18ad230f9e7bf6a77ffc2312c55e-Abstract-Datasets_and_Benchmarks.html

[17] Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. 2018. Learning to Teach. In *ICLR 2018,*.

[18] Hossein Fani, Reza Barzegar, Arman Dashti, and Mahdis Saeedi. [n. d.]. 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.*

[19] Matthew E. Gaston, John Simmons, and Marie desJardins. 2004. Adapting Network Structure for Efficient Team Formation. In *AAAI*, Vol. FS-04-02. AAAI Press, 1–8. https://www.aaai.org/Library/Symposia/Fall/2004/fs04-02-001.php

[20] Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022. Explainable Fairness in Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 681–691. https://doi.org/10.1145/3477495.3531973

[21] 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.

[22] 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*. 2348–2356. https://proceedings.neurips.cc/paper/2011/hash/7eb3c8be3d411e8ebfab08eba5f49632-Abstract.html

[23] Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. 2017. Automated Curriculum Learning for Neural Networks. 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, 1311–1320. http://proceedings.mlr.press/v70/graves17a.html

[24] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. 2018. CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X (Lecture Notes in Computer Science, Vol. 11214)*. Springer, 139–154. https://doi.org/10.1007/978-3-030-01249-6_9

[25] Rajni Gyanchandani, Bhavika Nathani, and Deepak Jaroliya. 2019. Factors Affecting Team Performance in IT Sector: An Exploratory Analysis. *J-GIBS* 11, 1 (2019).

[26] Guy Hacohen and Daphna Weinshall. 2019. On The Power of Curriculum Learning in Training Deep Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2535–2544. http://proceedings.mlr.press/v97/hacohen19a.html

[27] Radin Hamidi Rad, Hossein Fani, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2023. A Variational Neural Architecture for Skill-Based Team Formation. *ACM Trans. Inf. Syst.* 42, 1, Article 7 (aug 2023), 28 pages. https://doi.org/10.1145/3589762

[28] Junwei Han, Yang Yang, Dingwen Zhang, Dong Huang, Dong Xu, and Fernando De la Torre. 2021. Weakly-Supervised Learning of Category-Specific 3D Object Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 4 (2021), 1423–1437. https://doi.org/10.1109/TPAMI.2019.2949562

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[30] imdb. [n. d.]. IMDb Non-Commercial Datasets - https://developer.imdb.com/non-commercial-datasets/. https://developer.imdb.com/non-commercial-datasets/

[31] Hyunsik Jeon, Jong-eun Lee, Jeongin Yun, and U Kang. 2024. Cold-start Bundle Recommendation via Popularity-based Coalescence and Curriculum Heating. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*. ACM, 3277–3286. https://doi.org/10.1145/3589334.3645377

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

[33] 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. https://doi.org/10.1145/2063576.2063718

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

[35] 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

[36] Tom Kocmi and Ondrej Bojar. 2017. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, Ruslan Mitkov and Galia Angelova (Eds.). INCOMA Ltd.,

379–386. https://doi.org/10.26615/978-954-452-049-6_050
[37] Yue Kou, Derong Shen, Quinn Snell, Dong Li, Tiezheng Nie, Ge Yu, and Shuai Ma. 2020. Efficient Team Formation in Social Networks based on Constrained Pattern Graph. In *ICDE*. IEEE, 889–900. https://doi.org/10.1109/ICDE48307.2020.00082
[38] Solomon Kullback and Richard A Leibler. 1951. On Information and Sufficiency. *Annals of Mathematical Statistics* (1951).
[39] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*. Curran Associates, Inc., 1189–1197. https://proceedings.neurips.cc/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html
[40] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google Research Football: A Novel Reinforcement Learning Environment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 4501–4510. https://doi.org/10.1609/AAAI.V34I04.5878
[41] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki (Eds.). ACM, 467–476. https://doi.org/10.1145/1557019.1557074
[42] Haoyang Li, Xin Wang, and Wenwu Zhu. 2023. Curriculum Graph Machine Learning: A Survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 6674–6682. https://doi.org/10.24963/IJCAI.2023/748
[43] Nathan J McNeese, Mustafa Demir, Nancy J Cooke, and Manrong She. 2021. Team situation awareness and conflict: A study of human–machine teaming. *Journal of Cognitive Engineering and Decision Making* 15, 2-3 (2021), 83–96.
[44] Hoang Nguyen, Radin Hamidi Rad, Fattane Zarrinkalam, and Ebrahim Bagheri. 2023. DyHNet: Learning dynamic heterogeneous network representations. *Inf. Sci.* 646 (2023), 119371. https://doi.org/10.1016/j.ins.2023.119371
[45] Carol R Paris, Eduardo Salas, and Janis A Cannon-Bowers. 2000. Teamwork in multi-person systems: a review and analysis. *Ergonomics* 43, 8 (2000), 1052–1075.
[46] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based Curriculum Learning for Neural Machine Translation. 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, 1162–1172. https://doi.org/10.18653/v1/n19-1119
[47] Radin Hamidi Rad, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2021. Retrieving Skill-Based Teams from Collaboration Networks. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. https://doi.org/10.1145/3404835.3463105
[48] 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*. ACM, 2049–2052. https://doi.org/10.1145/3340531.3412140
[49] 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 '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 4716–4720. https://doi.org/10.1145/3459637.3481992
[50] Radin Hamidi Rad, Hoang Nguyen, 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
[51] William B. Rouse, Janis A. Cannon-Bowers, and Eduardo Salas. 1992. The role of mental models in team performance in complex systems. *IEEE Trans. Syst. Man Cybern.* 22, 6 (1992), 1296–1308. https://doi.org/10.1109/21.199457
[52] 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
[53] Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. 2019. Data Parameters: A New Family of Parameters for Learning a Differentiable Curriculum. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 11093–11103. https://proceedings.neurips.cc/paper/2019/hash/926ffc0ca56636b9e73c565cf994ea5a-Abstract.html
[54] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum Learning: A Survey. *Int. J. Comput. Vis.* 130, 6 (2022), 1526–1565. https://doi.org/10.1007/S11263-022-01611-X
[55] 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
[56] Joanna L Stollings, John W Devlin, John C Lin, Brenda T Pun, Diane Byrum, and Juliana Barr. 2020. Best practices for conducting interprofessional team rounds to facilitate performance of the ICU liberation (ABCDEF) bundle. *Critical care medicine* 48, 4 (2020), 562–570.
[57] Damjan Strnad and Nikola Guid. 2010. A fuzzy-genetic decision support system for project team formation. *Appl. Soft Comput.* 10, 4 (2010), 1178–1187. https://doi.org/10.1016/j.asoc.2009.08.032
[58] D. Strnad and N. Guid. 2010. A fuzzy-genetic decision support system for project team formation. *Applied Soft Computing* 10, 4 (2010), 1178–1187. https://doi.org/10.1016/j.asoc.2009.08.032 Optimisation Methods & Applications in Decision-Making Processes.
[59] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. [n. d.]. ArnetMiner: Extraction and Mining of Academic Social Networks - https://www.aminer.org/citation. https://www.aminer.org/citation
[60] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2020. Assessing ranking metrics in top-N recommendation. *Inf. Retr. J.* 23, 4 (2020), 411–448. https://doi.org/10.1007/S10791-020-09377-X
[61] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rklz9iAcKQ
[62] Lin Wang, Yifeng Zeng, Bilian Chen, Yinghui Pan, and Langcai Cao. 2020. Team Recommendation Using Order-Based Fuzzy Integral and NSGA-II in StarCraft. *IEEE Access* 8 (2020), 59559–59570. https://doi.org/10.1109/ACCESS.2020.2982647
[63] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 1717–1725. https://doi.org/10.1145/3447548.3467249
[64] 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
[65] Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 5235–5243. http://proceedings.mlr.press/v80/weinshall18a.html
[66] Hyeongon Wi, Seungjin Oh, Jungtae Mun, and Mooyoung Jung. 2009. A team formation model based on knowledge and collaboration. *Expert Syst. Appl.* 36, 5 (2009), 9121–9134. https://doi.org/10.1016/j.eswa.2008.12.031
[67] Hyeongon Wi, Seungjin Oh, Jungtae Mun, and Mooyoung Jung. 2009. A team formation model based on knowledge and collaboration. *Expert Systems with Applications* 36, 5 (2009), 9121–9134. https://doi.org/10.1016/j.eswa.2008.12.031
[68] Anita Williams Woolley, Rosalind M Chow, Anna T Mayo, Christoph Riedl, and Jin Wook Chang. 2023. Collective attention and collective intelligence: The role of hierarchy and team gender composition. *Organization Science* 34, 3 (2023), 1315–1331.
[69] Armen Zakarian and Andrew Kusiak. 1999. Forming teams: An analytical approach. *IIE Transactions* 31 (01 1999), 85–97. https://doi.org/10.1023/A:1007580823003
[70] Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. 2023. Automatic Grouping for Efficient Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
[71] Licheng Zhang, Zhendong Mao, Benfeng Xu, Quan Wang, and Yongdong Zhang. 2021. Review and Arrange: Curriculum Learning for Natural Language Understanding. *IEEE ACM Trans. Audio Speech Lang. Process.* 29 (2021), 3307–3320.