Tail-STEAK: Improve Friend Recommendation for Tail Users via Self-Training Enhanced Knowledge Distillation

Yijun Ma[1], **Chaozhuo Li**[2], **Xiao Zhou**[1*]

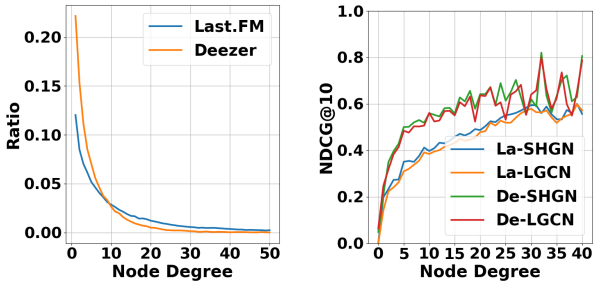
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2] Beijing University of Posts and Telecommunications
mayj_hedgehog@ruc.edu.cn, lichaozhuobupt@163.com, xiaozhou@ruc.edu.cn

Abstract

Graph neural networks (GNNs) are commonly employed in collaborative friend recommendation systems. Nevertheless, recent studies reveal a notable performance gap, particularly for users with limited connections, commonly known as *tail users*, in contrast to their counterparts with abundant connections (*head users*). Uniformly treating head and tail users poses two challenges for tail user preference learning: *(C1) Label Sparsity*, as tail users typically possess limited labels; and *(C2) Neighborhood Sparsity*, where tail users exhibit sparse observable friendships, leading to distinct preference distributions and performance degradation compared to head users. In response to these challenges, we introduce Tail-STEAK, an innovative framework that combines self-training with enhanced knowledge distillation for tail user representation learning. To address *(C1)*, we present Tail-STEAK$_{base}$, a two-stage self-training framework. In the first stage, only head users and their accurate connections are utilized for training, while pseudo links are generated for tail users in the second stage. To tackle *(C2)*, we propose two data augmentation-based self-knowledge distillation pretext tasks. These tasks are seamlessly integrated into different stages of Tail-STEAK$_{base}$, culminating in the comprehensive Tail-STEAK framework. Extensive experiments, conducted on state-of-the-art GNN-based friend recommendation models, substantiate the efficacy of Tail-STEAK in significantly improving tail user performance. Our code and data are publicly available at https://github.com/antman9914/Tail-STEAK.

Introduction

Friend recommender systems play a crucial role in various real-world applications, facilitating the discovery of potential social relationships and enhancing user engagement. The cornerstone of friend recommendation lies in learning effective user representations (Zhou et al. 2019). Recently, inspired by the development of Graph Neural Networks (GNNs), higher-order collaborative signals in social networks have been exploited for user representation learning and achieved significant improvement (Sankar et al. 2021). Despite their success, GNNs usually need qualified and abundant structural connections to learn effective user representations (Liu, Nguyen, and Fang 2021; Zheng et al. 2022), which high-degree users, or *head users*, can provide. However, most real-world social networks follow power-law node degree distribution (Adamic et al. 2001), where the majority of users are *tail users* with few links, as shown in Figure 1(a). As a result, due to limited observable interactions, the preference of tail users is hard to learn, leading to inferior performance in downstream recommendation tasks. As empirically demonstrated in Figure 1(b), for friend recommendation on Deezer and Last.FM social networks based on two state-of-the-art GNN-based models Simple-HGN (Lv et al. 2021) and LightGCN (He et al. 2020), the degree-specific predictive accuracy is approximately proportional to node degree. We denote this phenomenon as *degree-related bias*. Regrettably, contemporary recommendation algorithms often treat head users and tail users uniformly, resulting in the under-representation of tail users. This bottleneck is deemed unacceptable in real-world networks. Therefore, this paper is dedicated to enhancing tail user preference learning for friend recommendation with limited structural information.



(a) Degree Distribution (b) Degree-Related Evaluation Results

Figure 1: Empirical study of degree-related bias in friend recommendation

We contend that mitigating degree-related bias in friend recommendation introduces two challenges: *(C1) Label Sparsity,* where the scarcity of labels for tail users complicates preference learning, leading to an imbalance between head and tail users; and *(C2) Neighborhood Sparsity,* as the sparse interactions of tail users create a distinct preference distribution compared to head users, posing challenges in accurate anticipation and potentially resulting in a preference

*Corresponding author.

gap. Related works mainly focus on *(C2)*, which attempt to transfer accurate structural knowledge of head nodes to tail nodes for neighborhood sparsity alleviation (Liu et al. 2020; Liu, Nguyen, and Fang 2021; Zheng et al. 2022; Hao et al. 2021), or leverage side information to enrich relational data for inactive users (Zheng et al. 2021; Wang et al. 2019a; Yan et al. 2023). Although they effectively enhance the performance of inactive tail users, they not only ignore the more fundamental challenge *(C1)*, but also need external assistance to solve *(C2)*, which are usually overly complex.

To tackle the aforementioned challenges, we propose **Tail-STEAK**, a novel **Tail** user oriented **S**elf-**T**raining **E**nh**A**nced **K**nowledge distillation paradigm for alleviating degree-related bias in GNN-based friend recommendation. To address *(C1)*, we overhaul the training paradigm, introducing a fundamental two-stage self-training approach named Tail-STEAK$_{base}$. Initially, only head users and their well-qualified interactions are employed for model training in the first stage, leveraging their abundant and relatively accurate structural knowledge. Subsequently, in the second stage, we iteratively conduct top-K pseudo link predictions for tail users from a randomly sampled user set using the model derived from the previous iteration. This model is further refined using both the full training set and pseudo links. For *(C2)*, we propose two data augmentation-based self-knowledge distillation pretext tasks. These tasks aim to implicitly familiarize the model with both head and tail user preference distributions, thereby mitigating preference gaps. Conducted separately for head and tail users, these tasks are integrated into the corresponding stages of Tail-STEAK$_{base}$, forming the complete Tail-STEAK framework. To achieve data augmentation, we introduce synthesized tail users generated from original head users through aggressive link dropout and ID embedding disturbance in both stages. Additionally, we impute predicted pseudo links into the tail users' neighborhood and generate synthesized head users in the second stage. All synthesized users are then integrated into the training set for the respective stage. Diverging from mainstream reconstruction-based knowledge distillation methods (Ji et al. 2021), we employ self-discrimination-based distillation through Mutual Information (MI, denoted as MI throughout the paper) maximization between the head view and tail view of the same user.

It is essential to highlight that our proposed training paradigm is entirely model-agnostic and does not rely on additional customized modules or external data. We implement our approach on two cutting-edge GNN-based friend recommendation models, conducting comprehensive experiments across three benchmark social networks. The empirical results showcase a substantial enhancement in predictive accuracy for tail users, while maintaining competitive overall performance. Furthermore, our proposed method is versatile and applicable to general recommendation tasks and various other link prediction scenarios.

In summary, our contributions are highlighted as follows:

- We introduce Tail-STEAK$_{base}$, a foundational two-stage self-training paradigm designed for GNN-based friend recommendation, offering qualified pseudo labels for tail users to effectively address the label sparsity challenge.
- We devise distinct data augmentation strategies for head and tail users, synthesizing tail users through both embedding and structural space augmentation.
- We introduce two self-discrimination-based self-knowledge distillation tasks, seamlessly integrated into Tail-STEAK$_{base}$, enhancing the comprehensive Tail-STEAK framework.
- Empirical experiments conducted on two GNN-based friend recommendation models across two benchmark social networks substantiate the superiority of our method in tail user preference learning, while maintaining competitiveness in head user learning.

Related Work

Degree Bias in GNN-based Recommendations

Although GNNs have become mainstream solution for graph-related tasks and graph-based recommendation (Wang et al. 2019b; He et al. 2020; Wang et al. 2020; Zhao et al. 2023), there are some recent work revealing that GNNs are likely to suffer performance degradation on tail nodes, which raises a degree-related fairness concern. These works mostly focus on the sparse neighborhood of tail nodes, and make attempts to transfer head structural knowledge to them. For instance, DEMO-Net (Wu, He, and Xu 2019) and SL-DSGCN (Tang et al. 2020) assign interrelated degree-specific RNN-based parameters to input nodes with different degrees; A la carte (Khodak et al. 2018) and Nonce2vec (Herbelot and Baroni 2017) propose to conduct two-stage embedding refinement for robust tail node embedding; Meta-tail2vec (Liu et al. 2020) further utilizes meta-learning based two-stage embedding refinement framework for locality-aware tail node embedding; Tail-GNN (Liu, Nguyen, and Fang 2021) and Cold Brew (Zheng et al. 2022) both propose to directly impute the weak neighborhood of tail nodes. Tail-GNN utilize transferable neighborhood translation to predict missing neighborhood, while Cold Brew leverage self-attention based virtual neighborhood discovery. Different from aforementioned works, RawlsGCN (Kang et al. 2022) proposes a gradient modulation method to achieve the degree-level Rawlsian gradient fairness. GRADE (Wang et al. 2022) proposes a graph contrastive learning method to enhance the inherent community effect of networks via data augmentation.

Degree-related bias in graph-based recommendation is also known as cold-start problem, which is usually alleviated by introducing side information and constructing informative heterogeneous graphs, such as the profile of users and items (Zheng et al. 2021; Zhang et al. 2023), knowledge graphs (Wang et al. 2019a) and social networks (Liu et al. 2021). There is also a recent work (Hao et al. 2021) attempting to pre-train GNN-based recommendation models with reconstruction-based pretext task. Despite their success, they either do not specifically designed for improving tail user embeddings, or need additional modules or data, which is overly complex. More importantly, they fail to pay attention to the intuitive but critical challenge *(C2)*.

Self-Knowledge Distillation

Self-knowledge distillation is a kind of knowledge distillation, which has drawn growing attention in computer vision. Related works generally train a student network without auxiliary teacher network, and they can be divided into two groups: First group utilizes auxiliary networks. For example, BYOT (Zhang et al. 2019) introduces a set of auxiliary weak classfiers to perform classification based on the feature map of intermediate layers. FRSKD (Ji et al. 2021) proposes an auxiliary self-teacher network to enable refined knowledge transfer. The second group utilizes data augmentation. DDGSD (Xu and Liu 2019) induces consistent prediction by feeding differently augmented samples into encoder; CSKD (Yun et al. 2020) leverages different instances of the same class as positive pairs for class-level regularization, while SLA (Lee, Hwang, and Shin 2020) proposes to augment data label by combining self-supervision task with the original downstream task.

Our Tail-STEAK is inspired by data augmentation based branch, and we adopt the framework proposed in (Xu and Liu 2019). Most data augmentation based methods tend to make intermediate feature maps or predicted logits of different views to be similar. We also conduct distillation based on the outputs of GNN encoders. However, different from existing works, Tail-STEAK maximizes the MI of embeddings from different views instead of minimizing their Euclidean distance, in order to avoid the reconstruction constraint.

Preliminaries

In this section, we present the problem formulation of alleviating degree-related bias.

Given an undirected social network denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, ..., v_C\}, \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent the user set and the observed link set respectively. Let $\mathbf{X} \in \mathbb{R}^{C \times \delta}$ and $\mathbf{A} \in \mathbb{R}^{C \times C}$ denote the trainable ID embedding matrix and adjacency matrix, where $\mathbf{X}_{v:} \in \mathbb{R}^{\delta}$ is the ID embedding of user v, and $\mathbf{A}_{v:}$ is the adjacency vector originated from v. $\mathbf{A}_{uv} = \mathbf{A}_{vu} = 1$ iff $(u, v) \in \mathcal{E}$.

Let N_v denotes the neighboring node set of node $v \in \mathcal{V}$, and $|N_v|$ denotes the degree of user v. We denote $\mathbf{D} \in \mathbb{R}^{C \times C}$ as diagonal degree matrix, where $\mathbf{D}_{vv} = |N_v|$. Given degree threshold T, we can define the head node set $\mathcal{V}_{head}$ and tail node set $\mathcal{V}_{tail}$ as $\mathcal{V}_{head} = \{v : |N_v| > T\}$ and $\mathcal{V}_{tail} = \{v : |N_v| \leq T\}$ respectively. It is obvious that $\mathcal{V} = \mathcal{V}_{tail} \cup \mathcal{V}_{head}$ and $\mathcal{V}_{tail} \cap \mathcal{V}_{head} = \emptyset$. We further define C_{head} and C_{tail} as the number of head/tail nodes respectively, where $C_{head} + C_{tail} = C$. T is chosen based on the degree distribution of the given network, which is set as the median of given degree distribution in this work. The formal problem definition is presented as follows:

Problem. Given a multi-layer GNN-based user encoder $f(\mathbf{X}, \mathbf{A})$, our objective is to find a mapping $f : \mathcal{V} \rightarrow \mathbb{R}^{\delta}$ that can project each node $v \in \mathcal{V}$ into a δ-dimensional space, and meanwhile obtain more effective tail user embeddings $\{f(\mathbf{X}_{v:}, \mathbf{A}_{v:}) : v \in \mathcal{V}_{tail}\}$.

Methodology

In this section, we start with the introduction of our proposed two-stage self-training paradigm Tail-STEAK$_{base}$ to solve *(C1)*, along with the pseudo label prediction strategy. Next, to solve *(C2)*, we introduce the proposed data augmentation strategy and self-knowledge distillation pretext tasks, and present the full Tail-STEAK framework. An illustration of the overall framework is presented in Figure 2.

Basic Self-Training Paradigm

Most existing methods for degree-related bias mitigation fail to solve the fundamental label sparsity issue. To address *(C1)*, inspired by the widespread application of self-training (Tang et al. 2020; Liu et al. 2022), we propose a basic self-training paradigm denoted as Tail-STEAK$_{base}$ to provide more qualified pseudo links (i.e. labels) for tail users.

Self-training is generally a two-stage procedure, where the model is first trained with available labelled data, and iteratively trained with both labelled data and pseudo-labeled data generated from unlabelled data. Tail users have few links, and directly using the whole $\mathcal{E}$ to train the model in the first stage will be harmful for model performance. Therefore, we first train the model only with interactions of head users to learn more accurate user preference knowledge, and then add interactions of tail users in the second stage.

As for iterative pseudo link prediction, in each iteration, given tail users in the original training set, we first randomly sample U users that have not connected with these tail users from the whole graph, and then select the most relevant top-K users based on model prediction, which will be regarded as highly potential neighbors. The user subset sampling is designed for memory efficient training and diversified gradient provision. We simply set $K = T$ to make head and tail users have similar amount of labels. The weak links between target user and potential neighbors will be regarded as pseudo links. The pseudo links are expected to be less noisy, for that the model should have learned accurate preference distribution from head users in the previous stage, and able to automatically filter out noisy labels during iterative optimization. The predicted pseudo links will be used for both training and data augmentation, and will not participate the message propagation process of original samples.

Self-Knowledge Distillation

Although pseudo links can explicitly provide more supervision signals for tail users, neighborhood sparsity issue (i.e. *(C2)*) can also limit the model performance due to the incomplete observable preference distribution. Existing solutions tend to transfer relatively complete head preference knowledge to tail users via a variety of customized modules, which often make them overly complex (Liu et al. 2020; Zheng et al. 2022; Liu, Nguyen, and Fang 2021). In this work, we propose to leverage data augmentation based self-knowledge distillation to extract effective tail user embeddings via learned head knowledge. These operations are free of additional parameters and fully model-agnostic. We first propose two data augmentation methods in both structural space and embedding space for head and tail users re-

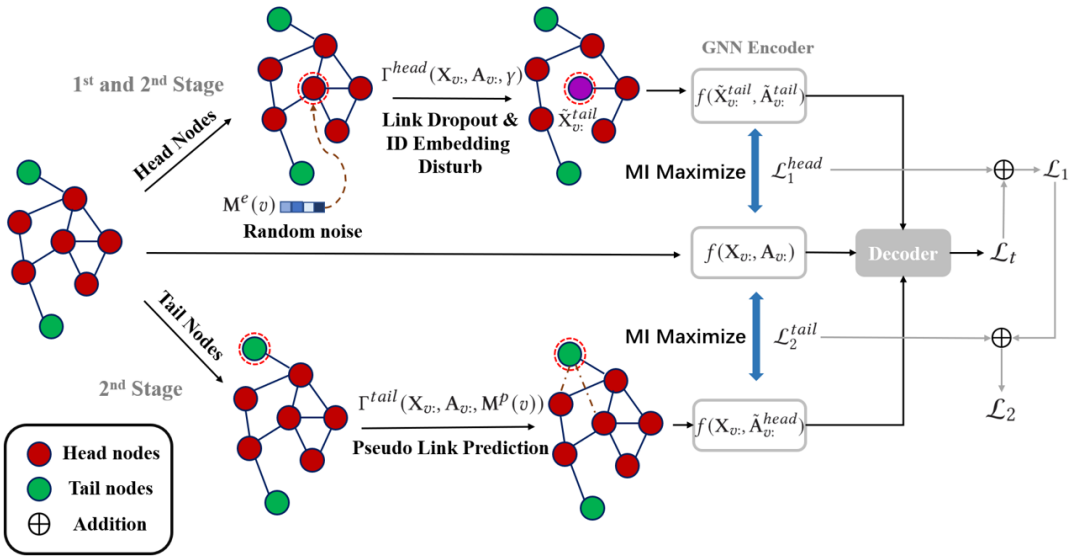


Figure 2: Overview of the proposed Tail-STEAK framework.

spectively, and then introduce the MI maximization based self-knowledge distillation pretext tasks.

Data augmentation. Synthesized data generation is the critical component of data augmentation based self-knowledge distillation (Xu and Liu 2019). In order to mitigate the preference gap between head and tail users, it is natural to consider corrupting the informative neighborhood and ID embedding of head users to simulate tail users, and meanwhile imputing the neighborhood of tail users to predict their head view. Therefore, we propose to conduct aggressive link dropout and ID embedding disturbance on head users to generate synthesized tail users, and impute predicted pseudo links of tail users into their neighborhood to generate synthesized head users, respectively. We first define the two independent data augmentation operator Γ^{head} and Γ^{tail} for synthesized tail user and head user generation respectively:

$$\Gamma^{head}(\mathbf{X}_{v:}, \mathbf{A}_{v:}, \gamma) = (\tilde{\mathbf{X}}^{tail}_{v:}, \tilde{\mathbf{A}}^{tail}_{v:}), v \in \mathcal{V}_{head} \quad (1)$$

$$\Gamma^{tail}(\mathbf{X}_{v:}, \mathbf{A}_{v:}, \mathbf{M}^p(v)) = (\mathbf{X}_{v:}, \tilde{\mathbf{A}}^{head}_{v:}), v \in \mathcal{V}_{tail} \quad (2)$$

Γ^{head} conducts data augmentation in both structure and embedding space. For the structural space, denote γ as the maximum preserved neighbors, we randomly select only a few neighbors of each head node to keep, in order to simulate the sparse neighborhood of tail users. Formally, given $v \in \mathcal{V}_{head}$, node degree $|N_v|$ and amount of preserved neighbors $z = \text{rand}(0, \gamma)$, the neighbor preservation probability will be $p^s_v = \frac{z}{|N_v|}$. Then, we can sample a random mask $\mathbf{M}^s(v) \in \{0,1\}_N$ for v's adjacency vector, where $\mathbf{M}^s_i(v) \sim \mathcal{B}(p^s_v)$ if $\mathbf{A}_{vi} = 1$ and $\mathbf{M}^s_i(v) = 0$ otherwise. The final adjacency vector of v can be denoted as:

$$\tilde{\mathbf{A}}^{tail}_{v:} = \mathbf{A}_{v:} \circ \mathbf{M}^s(v) \quad (3)$$

where $\circ$ is element-wise product. Note that the head link dropout is only conducted in the first-hop, the higher-order neighborhood are not affected.

For the embedding space, considering that the learned tail user ID embeddings are always noisy due to label sparsity, we add random noise $\mathbf{M}^e(v)$ sampled from standard Gaussian distribution to the original input user embedding $\mathbf{X}_{v:}$ to simulate the noisy tail user embedding. The embedding masking operation is only conducted on the center user.

$$\tilde{\mathbf{X}}^{tail}_{v:} = \mathbf{X}_{v:} + \mathbf{M}^e(v), \quad \mathbf{M}^e(v) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (4)$$

As for the Γ^{tail}, we simply impute the neighborhood of tail users with predicted pseudo links to generate synthesized head users. Formally, for each $v \in \mathcal{V}_{tail}$, given predicted pseudo adjacency vector $\mathbf{M}^p(v)$, the imputed adjacency vector of user v can be denoted as:

$$\tilde{\mathbf{A}}^{head}_{v:} = \mathbf{A}_{v:} \vee \mathbf{M}^p(v) \quad (5)$$

where $\vee$ is element-wise union. Note that the imputation is only conducted in the first-hop neighborhood. We denote the synthesized tail/head user generated from user v as v^{tail}/v^{head} respectively.

Pretext tasks. We formulate the output of the final layer of GNN-based user encoder as $\mathbf{H} = f(\mathbf{X}, \mathbf{A})$, where $\mathbf{h}_v$ is v's user embedding. Then, the output embedding matrix of synthesized tail users and head users can be denoted as $\mathbf{H}^{tail}$ and $\mathbf{H}^{head}$ respectively, which are defined as:

$$\mathbf{H}^{tail} = f(\tilde{\mathbf{X}}^{tail}_{v:}, \tilde{\mathbf{A}}^{tail}_{v:}), \quad \mathbf{H}^{head} = f(\mathbf{X}_{v:}, \tilde{\mathbf{A}}^{head}_{v:}) \quad (6)$$

where $\mathbf{h}^{tail}_v$ and $\mathbf{h}^{head}_v$ are the embeddings of v^{tail} and v^{head}. Generally, self-knowledge distillation is reconstruction-oriented (Ji et al. 2021), where the outputs of different input views are expected to be identical. In this work, we instead adopt MI maximization and popular

InfoNCE contrastive loss for distillation to avoid the strict reconstruction constraint. To adapt to the pair-wise objective, the embeddings of original user and the correspondingly generated synthesized users are regarded as positive pairs, i.e. $\mathcal{S}_{tail} = \{(\mathbf{h}_v, \mathbf{h}_v^{head}) : v \in \mathcal{V}_{tail}\}$ and $\mathcal{S}_{head} = \{(\mathbf{h}_v, \mathbf{h}_v^{tail}) : v \in \mathcal{V}_{head}\}$. On the other hand, the embeddings of synthesized users of other users within the same batch will be regarded as negative samples. Using MI-based distillation can make model pay more attention to distributional consistency between synthesized and corresponding source embeddings. Two distillation based pretext tasks are defined on head and tail users respectively, and their pairwise objective functions can be formulated as $\mathcal{L}_{dp}$ and $\mathcal{L}_{im}$:

$$\phi(\mathbf{h}_i, \mathbf{h}_j) = \exp(s(\mathbf{h}_v, \mathbf{h}_v^{tail})/\tau) \tag{7}$$

$$l_{dp}(v, v^{tail}) = -\log \frac{\phi(\mathbf{h}_v, \mathbf{h}_v^{tail})}{\sum_{(u,u^{tail})\in\mathcal{S}_{head}} \phi(\mathbf{h}_v, \mathbf{h}_u^{tail})}, \tag{8}$$

$$l_{im}(v, v^{head}) = -\log \frac{\phi(\mathbf{h}_v, \mathbf{h}_v^{head})}{\sum_{(u,u^{head})\in\mathcal{S}_{tail}} \phi(\mathbf{h}_v, \mathbf{h}_u^{head})} \tag{9}$$

$$\mathcal{L}_{dp}(v) = l_{dp}(v, v^{tail}) + l_{dp}(v^{tail}, v) \tag{10}$$

$$\mathcal{L}_{im}(v) = l_{im}(v, v^{head}) + l_{dp}(v^{head}, v) \tag{11}$$

where s is the cosine similarity function, and τ is the temperature hyperparamter. Note that although our proposed self-knowledge distillation based pretext tasks are similar to graph contrastive learning (GCL) based methods, such as GRACE (Zhu et al. 2020) and SGL (Wu et al. 2021), they are designed for different purposes. GCL methods devote to exploit unlabeled data space to alleviate data sparsity and improve *overall performance*. In contrast, our distillation-based method is designed for alleviating neighborhood sparsity, where the synthesized users are leveraged to conduct self-knowledge distillation, such that the derived model can comprehend both head and tail user preference distributions.

Overall Framework

We first define the BPR loss (Rendle et al. 2009) $\mathcal{L}_t$ for friend recommendation task, which is formulated as Eq. 12:

$$\mathcal{L}_t(v, u) = -\frac{1}{C_n} \sum_{u_n\in\mathcal{V}_n} [\log(\sigma(g(v, u) - g(v, u_n)))] \tag{12}$$

where u_n is negative sample, $\mathcal{V}_n$ is the negative user set, and C_n is the size of $\mathcal{V}_n$; g is the function for friendship prediction, which is defined as a two-layer MLP here. Based on Tail-STEAK$_{base}$ and our proposed self-knowledge distillation mechanism, given link batch $\mathcal{E}_B$, the objective function of the first stage can be formulated as $\mathcal{L}_1$ in Eq. 15:

$$\mathcal{L}_1^{head}(v, u) = \mathcal{L}_t(v, u) + \mathcal{L}_t(v^{tail}, u) + \mathcal{L}_{dp}(v) \tag{13}$$

$$\mathcal{L}_1^{head} = \frac{1}{|\mathcal{E}_B|} \sum_{(v,u)\in\mathcal{E}_B} \mathbb{1}_{head}(v)\mathcal{L}_1^{head}(v, u) \tag{14}$$

$$\mathcal{L}_1 = \mathcal{L}_1^{head} + \lambda\|\Omega\|_2 \tag{15}$$

	# Node	# Edge	$\lvert N_v\rvert$ Median	# Tail
Deezer	28,281	92,752	4	15,814
Last.FM	136,409	1,685,524	8	45,389

Table 1: Dataset Statistics

where $|\mathcal{E}_B|$ denotes batch size, Ω denotes all the trainable parameters in encoder f, $\mathbb{1}_{head}$ is an indicator function which returns 1 if the input user $v \in \mathcal{V}_{head}$ else 0. λ is a hyperparameter to control the strength of L2 regularization.

In the second stage, a new training set is constructed based on both observed and generated links. Both distillation pretext tasks are adopted in this stage. The corresponding objective function can be formulated as $\mathcal{L}_2$ in Eq. 18, where $\mathbb{1}_{tail}$ is another indicator function that returns 1 if the input user $v \in \mathcal{V}_{tail}$ else 0.

$$\mathcal{L}_2^{tail}(v, u) = \mathcal{L}_t(v, u) + \mathcal{L}_t(v^{head}, u) + \mathcal{L}_{im}(v) \tag{16}$$

$$\mathcal{L}_2^{tail} = \frac{1}{|\mathcal{E}_B|} \sum_{(v,u)\in\mathcal{E}_B} \mathbb{1}_{tail}(v)\mathcal{L}_2^{tail}(v, u) \tag{17}$$

$$\mathcal{L}_2 = \mathcal{L}_2^{tail} + \mathcal{L}_1 \tag{18}$$

Note that although Tail-STEAK is two-stage, we make modifications solely to the input data and objective function, and no additional modules are integrated into the base model, which keeps Tail-STEAK an end-to-end framework.

Experiments

Experimental Setup

Dataset. We conducted experiments on 2 public benchmark social networks, Deezer (Benedek and Rik 2020) and Last.FM[1]. Both datasets are friendship networks collected from different services in different time, where nodes and edges represent users and mutual friendships respectively, and node features are not available. The train/val/test split ratio is 70%/10%/20% for all the datasets. For each friendship to predict, We randomly sample 19 and 99 negative samples for training and testing respectively. Relevant statistics are presented in Table 1.

Base GNN Models. To evaluate the flexibility of our method, we adopt LightGCN (He et al. 2020) and Simple-HGN (Lv et al. 2021) (denoted as SHGN) as base GNNs. LightGCN is one of the most popular models for recommendation, while SHGN is a state-of-the-art GNN for heterogeneous graph learning. We remove the edge type embeddings and adapt SHGN to friend recommendation task.

Baseline Methods. Except for the base GNN models, We select four categories of baselines which are designed for mitigating degree-related bias or data sparsity issue to comprehensively evaluate our Tail-STEAK. *(1) Graph contrastive learning methods:* DGI (Veličković et al. 2019) maximizes MI between node views from original and corrupted graphs; GRACE (Zhu et al. 2020) augments graphs

[1]http://lfs.aminer.cn/lab-datasets/multi-sns/lastfm.tar.gz

		SHGN				LightGCN			
		$> T$		$\leq T$		$> T$		$\leq T$	
		NDCG	MRR	NDCG	MRR	NDCG	MRR	NDCG	MRR
Deezer	Base	0.5555	0.4943	0.2669	0.2352	0.5511	0.4948	0.2663	0.2278
	DGI	0.5651	0.5088	0.2860	0.2528	0.5653	0.5115	0.2533	0.2178
	GRACE	**0.6186**	**0.5684**	0.3108	0.2844	0.5743	0.5211	0.2494	0.2143
	MvGRL	0.5778	0.5184	0.2709	0.2414	0.5593	0.5050	0.2611	0.2243
	SGL	0.5881	0.5357	0.2974	0.2713	0.5724	0.5194	0.2489	0.2139
	NCL	0.5921	0.5397	0.3059	0.2758	**0.5788**	**0.5267**	0.2475	0.2121
	SimGCL	0.6023	0.5511	0.2894	0.2647	0.5733	0.5205	0.2501	0.2158
	LFT	0.5486	0.4880	0.2525	0.2211	0.5603	0.5049	0.2674	0.2280
	MoE	0.5457	0.4843	0.2751	0.2399	-	-	-	-
	Tail-GNN	0.4660	0.4180	0.1786	0.1412	0.5644	0.5089	0.2486	0.2121
	SSNet	0.5574	0.4954	0.2649	0.2350	0.5780	0.5250	0.2499	0.2182
	Tail-STEAK$_{\text{no-mask}}$	0.5414	0.4776	0.3258	0.2898	0.5549	0.5109	**0.3225**	**0.2918**
	Tail-STEAK$_{\text{full}}$	0.5845	0.5293	**0.3550**	**0.3337**	0.5687	0.5210	0.3139	0.2755
	Tail Improv.	-	-	14.22%	17.33%	-	-	20.60%	28.09%
Last.FM	Base	0.6239	0.5560	0.3153	0.2879	0.6372	0.5692	0.2965	0.2414
	DGI	0.6292	0.5618	0.3077	0.2747	0.6365	0.5684	0.2978	0.2424
	GRACE	**0.6682**	**0.6070**	0.4214	0.3904	0.6432	0.5762	0.3020	0.2436
	MvGRL	0.6323	0.5652	0.3103	0.2818	0.6395	0.5726	0.3024	0.2495
	SGL	0.6449	0.5794	0.3233	0.2764	0.6397	0.5720	0.3087	0.2498
	NCL	0.6482	0.5838	0.3472	0.3059	0.6302	0.5614	0.2715	0.2123
	SimGCL	0.6483	0.5838	0.3797	0.3475	0.6389	0.5710	0.3034	0.2446
	LFT	0.6175	0.5501	0.1722	0.1399	0.6345	0.5657	0.3101	0.2480
	MoE	0.6208	0.5520	0.3211	0.2923	-	-	-	-
	Tail-GNN	0.6003	0.5308	0.3281	0.2872	0.6421	0.5749	0.2768	0.2202
	SSNet	0.6238	0.5561	0.3167	0.2905	**0.6572**	**0.5953**	0.3328	0.2961
	Tail-STEAK$_{\text{no-mask}}$	0.6154	0.5547	0.3784	0.3241	0.6290	0.5631	**0.4594**	**0.3956**
	Tail-STEAK$_{\text{full}}$	0.6455	0.5822	**0.5409**	**0.5023**	0.6352	0.5698	0.4199	0.3537
	Tail Improv.	-	-	28.36%	28.66%	-	-	48.15%	58.37%

Table 2: Degree-Related NDCG@10 and MRR Evaluation Results. Boldfaced scores are the best ones.

by link dropout and node feature masking, and correlates generated views via self-discrimination; MvGRL (Hassani and Khasahmadi 2020) introduces graph diffusion into graph contrastive learning. They aim to exploit the unlabeled data space and alleviate data sparsity, but they do not specifically focus on tail node improvement. *(2) Adaptive embedding refinement models:* LFT (Zhu and Caverlee 2022) first learn a common prior model with all available labels, which is then fine-tuned with nodes with different degrees. Similarly, MoE (Masoudnia and Ebrahimpour 2014) trains several expert encoders for nodes with different degrees, and then derive the best expert combinations via a degree-aware gating network. Note that MoE is not applicable for LightGCN, for that there is no encoder in LightGCN. *(3) Weak neighborhood imputation models:* Tail-GNN (Liu, Nguyen, and Fang 2021) attempts to directly impute weak neighborhood of tail nodes via transferable neighborhood translation. *(4) Self-supervised learning methods for recommendation:* SGL (Wu et al. 2021) and SimGCL (Yu et al. 2022) perform augmentation over graph structure and user embeddings via random dropout respectively. NCL (Lin et al. 2022) proposes heuristic-based strategies to construct different views based on structural and semantic neighborhood. They are designed specifically for recommendation, in order to alleviate data sparsity issue. We also adopt SSNet (Song et al. 2022) for comparison, which is recently proposed to alleviate scale distortion issue in friend recommendation.

We do not consider meta-tail2vec (Liu et al. 2020) and Cold Brew (Zheng et al. 2022) as baselines, for that embedding reconstruction is required in both methods, which is not suitable for recommendation. GRADE (Wang et al. 2022) is also abandoned due to its massive memory requirement. For our method, we evaluate two versions of Tail-STEAK. Tail-STEAK$_{\text{no-mask}}$ removes the feature-space operation in the first stage, while Tail-STEAK$_{\text{full}}$ is the full version.

Evaluation Metrics. Following previous works of friend recommendation, we adopt 2 commonly used metrics for evaluation, which are MRR (Mean Reciprocal Ranking) and NDCG@K (Normalized Discounted Curriculum Gain). Both MRR and NDCG can reflect the ranking quality. We set $K = 10$ for NDCG@K.

Implementation Details. We implement Tail-STEAK via PyTorch-Geometric (Fey and Lenssen 2019). We adopt 2-layer GNN architecture, and ID embedding dimension δ is fixed as 64. γ in Γ^{head} and $\mathcal{U}$ for pseudo link generation are tuned from $\{1,2,...,8\}$ and $\{200, 500, 1000, 2000\}$ respectively. Adam (Kingma and Ba 2014) is adopted for optimization with learning rate 0.001, and $\lambda = 0.0001$. You can refer to our Github repo for more details.

Comparative Results

We report the average degree-related performance of our proposed Tail-STEAK and other baselines after 5 runs with

Methods	SHGN		LightGCN	
	$> T$	$\leq T$	$> T$	$\leq T$
Tail-STEAK$_{\text{full}}$	0.5845	0.3550	0.5687	0.3139
w/o ID disturb	0.5414	0.3258	0.5549	0.3225
w/o 2nd tail-based KD	0.5983	0.3685	0.5577	0.2585
w/o 2nd KD	0.5939	0.3628	0.5588	0.2575
w/o 2nd stage	0.5951	0.3656	0.5693	0.2481
w/o sep stage	0.5844	0.3458	0.5477	0.2569
w/o user subset sample	0.6108	0.3381	0.5806	0.2663
w/o MI loss	0.5137	0.3132	0.5444	0.2531
w/o KD	0.5588	0.3378	0.5507	0.2603

Table 3: Degree-Related NDCG@10 Evaluation Results of Ablation Study on Deezer.

different seeds in Table 2. The relative improvement of tail user performance is also presented. We have following observations: (1) Our Tail-STEAK consistently outperforms state-of-the-art baselines by a large margin in terms of tail users in both base GNN models and in both selected datasets.(2) Althogh head user performance of Tail-STEAK always drops compared with the best-performing baselines, it alomst consistently outperforms baselines specifically tailored to address degree-related bias like Tail-GNN and LFT. (3) Among all contrastive learning based methods, only GRACE improve both head and tail user performance compared with base model. DGI and MvGRL mainly improve head user performance, while the effect of SGL, NCL and SimGCL depends on specific dataset. (4) LFT and MoE are less beneficial and even harmful for both head and tail user performance. We believe the reason is that model adaptation based on few labels of tail users may cause over-fitting. However, for LightGCN, LFT can significantly improve tail user performance. (5) Both Tail-GNN and SSNet can effectively improve both head and tail user performance of LightGCN in certain datasets. However, they are much less effective and even harmful for SHGN. (6) By comparing Tail-STEAK$_{\text{no-mask}}$ and Tail-STEAK$_{\text{full}}$, we can find that the ID embedding disturbance operation has significant impact on the model performance, which is beneficial for SHGN and harmful for LightGCN. We believe the reason is that random noise in feature space is helpful for the learning of GNNs with non-linear transformations like SHGN.

Ablation Study

To empirically discuss the impact of each proposed component, we conduct two branches of ablation experiments on Deezer. Specifically, in the first branch, we first remove terms $\mathcal{L}_2^{tail}$ (w/o 2nd tail-based KD) and $\mathcal{L}_1^{head} + \mathcal{L}_2^{tail}$ (w/o 2nd KD) in $\mathcal{L}_2$ respectively, and then remove the whole second stage (w/o 2nd stage). In the second branch, we discuss several noteworthy alternatives of Tail-STEAK. We first re-train GNN models only based on the second stage (w/o sep stage), and then use all available users for pseudo link prediction (w/o user subset sample). We also replace the MI maximization based distillation loss with traditional reconstruction-based loss (w/o MI loss), and further replace the distillation strategy with pure data augmentation operation (w/o KD), where the synthesized users are only used as training samples. Note that we have shown the impact of ID embedding disturbance (w/o ID disturb) in previous section, so we will skip relevant discussion here. We report the degree-related evaluation results in Table 3.

Based on the evaluation results, we can find that our proposed components have different impact on different base model. For the first branch: (1) For SHGN, head user based self-knowledge distillation significantly improves both head and tail user performance, while tail user based distillation can be harmful for both user groups. Predicting pseudo links can bring limited improvement for SHGN-based friend recommenders, and the pseudo labels may harm the performance without the guidance of head user based distillation. (2) For LightGCN, head user based distillation has little impact on tail user performance, which is opposite for tail user based distillation. Pseudo link prediction is also critical for improving tail user performance for LightGCN-based recommender. For the second branch: (1) Separating Tail-STEAK into two stages and pre-train a qualified model in the first stage is necessary for SHGN, while it is less helpful for LightGCN. (2) Introducing diversified supervision signals via randomly sampled potential friends is helpful for SHGN-based friend recommender optimization, which can also be harmful for LightGCN-based recommenders. (3) MI maximization based distillation loss is superior than reconstruction-based distillation loss for both base models. (4) Removing all the knowledge distillation based objective terms will lead to significant performance degradation for SHGN, while have little impact on LightGCN.

The reason why proposed components have different impact lies in the significant difference of base model architecture . There is no actual encoder in LightGCN, which only has ID embedding layer and iteratively performs message propagation among adjacent users. In contrast, SHGN has trainable encoders with non-linear transformations.

Conclusion

In this work, we studied the problem of degree-related bias in graph-based friend recommendation. We identify two major challenges in this problem: *(C1) Label sparsity; (C2) Neighborhood sparsity.* To tackle these challenges, we propose Tail-STEAK, a novel model-agnostic self-training enhanced knowledge distillation framework free of additional parameters. Tail-STEAK is developed based on a two-stage self-training paradigm named Tail-STEAK$_{base}$ to address *(C1)*, where only head nodes and their qualified connections are used for model training in the first stage, followed by predicting pseudo links for tail users in the second stage. To address *(C2)*, two data augmentation-based self-knowledge distillation pretext tasks are further incorporated into Tail-STEAK, which conduct data augmentation in both feature and structural space to distill the rich knowledge of head users into tail users, in order to help model comprehend both head and tail user preference distributions. Comprehensive experiments on two GNN-based friend recommendation models and benchmark datasets demonstrate that Tail-STEAK can significantly improve tail user performance, and meanwhile maintains competitive head user performance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC Grant No.62106274); the Fundamental Research Funds for the Central Universities, Renmin University of China (No.22XNKJ24). We also wish to acknowledge the support provided by the Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative.

References

Adamic, L. A.; Lukose, R. M.; Puniyani, A. R.; and Huberman, B. A. 2001. Search in power-law networks. *Phys. Rev. E*, 64: 046135.

Benedek, R.; and Rik, S. 2020. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, 1325–1334.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *Proceedings of the 7th International Conference on Learning Representations (RLGM Workshop)*, ICLR '18.

Hao, B.; Zhang, J.; Yin, H.; Li, C.; and Chen, H. 2021. Pre-Training Graph Neural Networks for Cold-Start Users and Items Representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, 265–273.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 639–648.

Herbelot, A.; and Baroni, M. 2017. High-risk learning: acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, EMNLP '17, 304–309.

Ji, M.; Shin, S.; Hwang, S.; Park, G.; and Moon, I.-C. 2021. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, CVPR '21, 10664–10673.

Kang, J.; Zhu, Y.; Xia, Y.; Luo, J.; and Tong, H. 2022. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. In *Proceedings of the ACM Web Conference 2022*, WWW '22, 1214–1225.

Khodak, M.; Saunshi, N.; Liang, Y.; Ma, T.; Stewart, B.; and Arora, S. 2018. A La Carte Embedding: Cheap but Effective Induction of Semantic Feature Vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL '18, 12–22.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, H.; Hwang, S. J.; and Shin, J. 2020. Self-supervised Label Augmentation via Input Transformations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, 5714–5724.

Lin, Z.; Tian, C.; Hou, Y.; and Zhao, W. X. 2022. Improving Graph Collaborative Filtering with Neighborhood-Enriched Contrastive Learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, 2320–2329.

Liu, H.; Hu, B.; Wang, X.; Shi, C.; Zhang, Z.; and Zhou, J. 2022. Confidence May Cheat: Self-Training on Graph Neural Networks under Distribution Shift. In *Proceedings of the ACM Web Conference 2022*, WWW '22, 1248–1258.

Liu, Z.; Nguyen, T.-K.; and Fang, Y. 2021. Tail-GNN: Tail-Node Graph Neural Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, 1109–1119.

Liu, Z.; Shen, Y.; Cheng, X.; Li, Q.; Wei, J.; Zhang, Z.; Wang, D.; Zeng, X.; Gu, J.; and Zhou, J. 2021. Learning Representations of Inactive Users: A Cross Domain Approach with Graph Neural Networks. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21, 3278–3282.

Liu, Z.; Zhang, W.; Fang, Y.; Zhang, X.; and Hoi, S. C. 2020. Towards Locality-Aware Meta-Learning of Tail Node Embeddings on Networks. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, 975–984.

Lv, Q.; Ding, M.; Liu, Q.; Chen, Y.; Feng, W.; He, S.; Zhou, C.; Jiang, J.; Dong, Y.; and Tang, J. 2021. Are We Really Making Much Progress? Revisiting, Benchmarking and Refining Heterogeneous Graph Neural Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, 1150–1160.

Masoudnia, S.; and Ebrahimpour, R. 2014. Mixture of experts: a literature survey. *The Artificial Intelligence Review*, 42(2): 275.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461.

Sankar, A.; Liu, Y.; Yu, J.; and Shah, N. 2021. Graph Neural Networks for Friend Ranking in Large-Scale Social Platforms. In *Proceedings of the Web Conference 2021*, WWW '21, 2535–2546.

Song, X.; Lian, J.; Huang, H.; Wu, M.; Jin, H.; and Xie, X. 2022. Friend Recommendations with Self-Rescaling Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 3909–3919.

Tang, X.; Yao, H.; Sun, Y.; Wang, Y.; Tang, J.; Aggarwal, C.; Mitra, P.; and Wang, S. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, 1435–1444.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19.

Wang, R.; Wang, X.; Shi, C.; and Song, L. 2022. Uncovering the Structural Fairness in Graph Contrastive Learning. In *Proceedings of the 36th Advances in Neural Information Processing Systems*, NeurIPS '22.

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019a. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, 950–958.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019b. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, 165–174.

Wang, X.; Jin, H.; Zhang, A.; He, X.; Xu, T.; and Chua, T.-S. 2020. Disentangled Graph Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 1001–1010.

Wu, J.; He, J.; and Xu, J. 2019. DEMO-Net: Degree-Specific Graph Neural Networks for Node and Graph Classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 406–415.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-Supervised Graph Learning for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, 726–735.

Xu, T.-B.; and Liu, C.-L. 2019. Data-Distortion Guided Self-Distillation for Deep Neural Networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19.

Yan, H.; Li, C.; Long, R.; Yan, C.; Zhao, J.; Zhuang, W.; Yin, J.; Zhang, P.; Han, W.; Sun, H.; et al. 2023. A Comprehensive Study on Text-attributed Graphs: Benchmarking and Rethinking. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, 1294–1303.

Yun, S.; Park, J.; Lee, K.; and Shin, J. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13876–13885.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ICCV '19, 3712–3721.

Zhang, P.; Guo, J.; Li, C.; Xie, Y.; Kim, J. B.; Zhang, Y.; Xie, X.; Wang, H.; and Kim, S. 2023. Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 168–176.

Zhao, J.; Qu, M.; Li, C.; Yan, H.; Liu, Q.; Li, R.; Xie, X.; and Tang, J. 2023. Learning on large-scale text-attributed graphs via variational inference. *ICLR*.

Zheng, J.; Ma, Q.; Gu, H.; and Zheng, Z. 2021. Multi-View Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-Start Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, 2338–2348.

Zheng, W.; Huang, E. W.; Rao, N.; Katariya, S.; Wang, Z.; and Subbian, K. 2022. Cold Brew: Distilling Graph Node Representations with Incomplete or Missing Neighborhoods. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22.

Zhou, X.; Liu, D.; Lian, J.; and Xie, X. 2019. Collaborative metric learning with memory network for multi-relational recommender systems. *arXiv preprint arXiv:1906.09882*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep Graph Contrastive Representation Learning. In *Proceedings of the 33rd Advances in Neural Information Processing Systems*, NeurIPS '20.

Zhu, Z.; and Caverlee, J. 2022. Fighting Mainstream Bias in Recommender Systems via Local Fine Tuning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, 1497–1506.