2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386104

Dual-Teacher Knowledge Distillation for Strict Cold-Start Recommendation

Weizhi Zhang[1], Liangwei Yang[1], Yuwei Cao[1], Ke Xu[1], Yuanjie Zhu[1] and Philip S. Yu[1]
[1]*Department of Computer Science, University of Illinois Chicago*, USA
{wzhan42, lyang84, ycao43, kxu25, yzhu224, psyu}@uic.edu

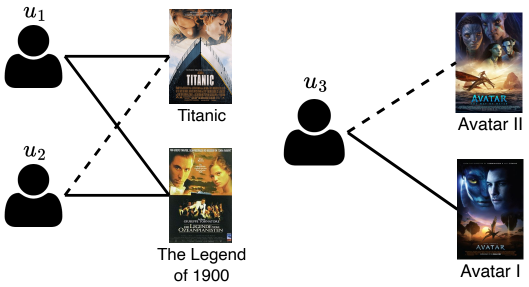
***Abstract*—Recommender systems (RecSys) aim to predict users' preferences based on historical interactions and content profiles, and they are vital components of many online services. However, the strict cold-start (SCS) issue, i.e., users/items have no prior interactions, poses significant challenges for RecSys. The existing methods seek to transfer content knowledge, collaborative filtering (CF) knowledge, or combine the two from the warm-start scenario towards the (strict) cold-start scenarios. However, these approaches either ignore the available information or model the information in rough manners such that the two types of knowledge interfere with each other, leading to ineffective and uncontrolled knowledge transfer. In this work, we propose a novel dual-teacher knowledge distillation (DTKD) framework that simultaneously and effectively transfers both content and CF knowledge. The proposed DTKD framework contains two teachers, one for each knowledge type, that is specifically designed according to the characteristics of the content and CF data to distill the knowledge fully. Soft scoring is calculated during the distillation to denoise and augment the original hard-labeled interactions. A knowledge fusion module is then proposed to collect the consensus of the two teachers' opinions. Finally, DTKD transfers both content and CF knowledge into a student module that learns the shared viewpoints of the teachers. We conduct extensive experiments on real-world datasets under the warm-start as well as three different SCS settings (i.e., strict cold users, strict cold items, and strict cold users & items). Experimental results show that DTKD outperforms strong baselines by large margins under all settings, especially the SCS ones.**

***Index Terms*—Recommendation, Strict cold-start recommendation, Knowledge distillation**

I. Introduction

As one of the most profound application areas in machine learning, the recommender systems (RecSys) play a significant role in many online services [1]–[3] and can be found in almost every aspect of our lives [4]. Based on the historical interactions and content profiles, RecSys aims to predict users' preferences for items. It helps users find interesting items and contributes to the success of many online platforms [1].

Despite the ongoing progress in RecSys, little attention has been devoted to addressing the significant issue of strict cold-start (SCS), which arises when users/items have no prior interaction. The SCS issue is commonly observed in early-stage online platforms and applications with a constant influx of new items, such as news recommendation systems [5]. Effectively addressing SCS recommendations is crucial for achieving business success. For instance, imagine an online system that struggles to provide accurate product recommendations to newly registered users. Without engaging these users, they will likely lose interest and eventually abandon the platform. From the sellers' perspective, numerous new items may remain in a state of zero comments if no appropriate recommendations are presented to existing users. Therefore, SCS is essential for RecSys.



(a) **CF knowledge** (b) **Content knowledge**

Fig. 1: **Two types of knowledge in the RecSys. The solid lines are the observed interactions, while the dashed line represents potential interactions inferred from either historical interactions or the content profiles.**

RecSys leverages two types of knowledge for making recommendations, i.e., *collaborative filtering (CF) [6] knowledge* and *content knowledge*. Figure 1 illustrates how RecSys explore these two types of knowledge. Specifically, in Figure 1a, based on the CF knowledge, one can infer that u_2 may be interested in "Titanic" as u_2 and u_1 share the same interest in "The Legend of 1900". Figure 1b, on the other hand, recommends "Avatar II" to u_3 because it is the same series as "Avatar I" with similar content profiles (director, actors, plots, etc.). Note that different from the warm-start scenario, where we have the two types of knowledge available for all the users and items, in SCS recommendation, the CF knowledge about the cold users/items is missing, as they have no previous interactions. Therefore, solving the SCS issue is equivalent to asking the following two questions: *1) how can we effectively model the available knowledge? 2) how to transfer the knowledge from the warm-start scenario to the SCS scenarios?*

The existing recommenders, however, fail to fully answer the above two questions. Specifically, the CF-based models [1], [6]–[8] depend solely on the CF knowledge and cannot work in the SCS scenarios in which the CF information is unavailable. The content-based models [2], [9]–[11], on the other hand, merely leverage the content (auxiliary user profiles and item descriptions) and ignore the CF knowledge. Though able to recommend in the SCS scenarios, these models show inferior performance in the warm-start scenario as compared to the CF-based models, which suggests that their knowledge modeling is ineffective due to the CF information loss. Hybrid methods [12]–[15] that combine CF knowledge and content knowledge are increasingly being adopted as a prevalent solution for addressing cold-start recommendation. These methods typically leverage the warm data to train a model that aligns the CF and the content embedding spaces. The trained model is then used to reconstruct the CF knowledge about the new users and items from their content knowledge. However, directly aligning the CF and the content embedding spaces with enforced similarity loss cannot guarantee how much CF/content knowledge is preserved, leading to ineffective knowledge modeling. Moreover, these methods transfer knowledge uncontrolled: they rely on the neural network's fitting ability to implicitly conduct the knowledge transfer, in which two types of knowledge can interfere. These methods didn't consider the fact that the two types of knowledge play distinct roles and can complement rather than interfere with each other. Consequently, a unified recommender that effectively models the hybrid knowledge and explicitly transfers knowledge from the warm-start to the SCS scenarios in a guided manner is needed.

In this work, we propose a novel dual-teacher knowledge distillation (DTKD) framework that simultaneously transfers both content and CF knowledge. We first seek to address the challenge of **effective knowledge modeling**. The proposed DTKD framework contains two teachers, one for each knowledge type, that is designed according to the characteristics of the content and CF data to distill the knowledge fully. Specifically, one teacher leverages the GNN [7] to model the CF interactions that are represented as edges in a user-item bipartite graph. The other teacher leverages a linear layer and focuses on extracting useful information from each first-order embedded content field. In addition, we observe that the original hard-labeled interaction data is noisy and sparse. To address this, soft scoring is performed during the knowledge distillation process to denoise and augment the data. Such dual-teacher design and soft score distillation guarantee effective modeling of both types of knowledge. To this end, though the CF and content models may be directly used for warm-start and SCS recommendation separately, it is infeasible to explicitly define a boundary of cold-start and warm-start users/items and partition them in real-world recommendations and a universal model that consists of hybrid knowledge and is capable of performing on all settings is required for efficient RecSys. Then, towards the second challenge of **controlled knowledge transfer**, we propose a soft score knowledge fusion module that collects the consensus of the two teachers' opinions. Specifically, the fusion module calculates a weighted sum of the soft scores provided by the two teachers, with tunable weights that control the levels of authority of the teachers. This allows more comprehensive knowledge to be transferred and thus contributes to a wiser student. Upon the agreement of the two teachers, DTKD then transfers both content and CF knowledge into a multi-layer perceptron (MLP) student module that learns the shared viewpoints of the teachers. In this way, DTKD manages to transfer both types of knowledge in an explicit and controlled manner and avoid the student's confusion raised by the chaotically combined knowledge. We conduct extensive experiments on real-world datasets under the warm-start as well as three different SCS settings (i.e., strict cold users, strict cold items, and strict cold users & items), demonstrating that DTKD outperforms strong baselines by large margins under all settings, especially the SCS ones. We empirically verify the effectiveness of the components of DTKD, especially the soft scoring KD mechanism, and also find that DTKD enables the student to surpass its master teachers on large sparse datasets. Finally, we analyze the effects of changing hyper-parameters, i.e., the weights of the teachers and the distillation temperature. Our code and preprocessed data are publicly available [1]. Our main contributions are summarized into three folds:

- We propose a dual-teacher-designed framework DTKD that simultaneously learns both content and CF knowledge for warm-start and SCS recommendations.
- We propose the soft score knowledge fusion strategy, including the soft score augmentation/denoising and a fusion module that effectively transfers the hybrid knowledge into a single unified model in a controlled manner.
- We conduct extensive experiments on three real-world datasets to verify the effectiveness of DTKD, especially under different SCS settings.

II. Problem formulation

In this section, we formalize the warm-start recommendation and three strict cold-start recommendation problems. Formally, assume we are given a set of warm users and items $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$, $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, where m and n represent the number of users and items, respectively, and each of user/item gains at least one interactions. Then we also have a list of p strict cold-start users $\hat{\mathcal{U}} = \{u_1, u_2, \dots, u_p\}$ and q strict cold-start items $\hat{\mathcal{V}} = \{v_1, v_2, \dots, v_q\}$ with zero interaction records. During training, the SCS model can learn from both interaction history ω and corresponding content information ϕ of warm users/items. Then in the testing phase, the model can only take the content information of the cold users or items to make recommendations. Specifically, to thoroughly evaluate the effectiveness of DTKD, we define one warm-start and three SCS recommendation tasks.

- **Warm-start (Warm user, Warm item).** The task is to recommend warm items $v_j \in \mathcal{V}$ to warm users $u_i \in \mathcal{U}$, regarding interactions ω_i, ω_j and the attribute ϕ_i, ϕ_j.
- **User SCS (Cold user, Warm item).** Given a cold user $u_i \in \hat{\mathcal{U}}$, the goal is to make personalized recommendations of warm items $v_j \in \mathcal{V}$ to u_i based on the attribute ϕ_i, ϕ_j.

[1] https://github.com/DavidZWZ/DTKD_SCS

- **Item SCS (Warm user, Cold item).** When cold items $v_j \in \hat{\mathcal{V}}$ appear with side information ϕ_j, the goal is to recommend such items v_j to warm users $u_i \in \mathcal{U}$.
- **User-item SCS (Cold user, Cold item).** In the condition where both when users $u_i \in \hat{\mathcal{U}}$ and items $v_j \in \hat{\mathcal{V}}$ appear without any interaction, we have to recommend these cold items to cold users with attributes ϕ_i and ϕ_j.

III. Methodology

In this section, we introduce our proposed DTKD framework. Section III-A presents the overview of DTKD. Sections III-B - III-E illustrate its components, including a GNN-based CF teacher model, a linear content teacher model, a soft score knowledge fusion module, and an MLP student model.

A. Dual-Teacher KD Framework

As discussed in Section I, the content-based models lack CF knowledge and existing hybrid methods may incur information loss and lack a controlled way to integrate two types of information. Therefore, we aim to extract more comprehensive knowledge (preserve both CF and content information) to enhance the model performance in SCS scenarios.

Knowledge distillation (KD) [16] presents a good way for comprehensive knowledge transfer. It can leverage multiple teachers with different initialization [16] or model architectures [17] to obtain diverse views of prediction, thereby improving the student performance. As demonstrated in Section I, the recommendation task naturally has two different information sources, i.e., CF data and content data, which allow models to get distinct views. Motivated by this, we propose a dual-teacher knowledge distillation framework to ensemble both CF and content information. Specifically, our framework mainly consists of two teacher models, a knowledge fusion module followed by a student module, as shown in Figure 2. The first teacher model is designed to learn the fruitful collaborative filtering signals from the historical interaction records. The second teacher model corresponds to learning the content embedding from the content materials. Then a soft score knowledge fusion module is proposed to refine and combine the two teachers' knowledge in a controlled manner. Finally, the student model can distill the fused knowledge for both warm-start and cold-start recommendations.

B. GNN-based CF Teacher

The CF teacher is designed to aggregate the neighborhood information and learn the collaborative filtering knowledge, and we utilize the graph mining power of GNN.

1) Light graph convolution.: Suppose given a randomly initialized input user ID embedding $\mathbf{h}_u^0$, and we adopt the same graph convolution operation as in [7], by consistently averaging the connected neighbors' embeddings: $\mathbf{h}_u^{k+1} = \sum_{v \in \mathcal{N}_u} \frac{\mathbf{h}_u^k}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_v|}}$, where $\mathbf{h}_u^k$ and $\mathbf{h}_v^k$ are embeddings of user u and item i at the k-th layer, respectively. $\frac{1}{\sqrt{|\mathcal{N}_v|}\sqrt{|\mathcal{N}_u|}}$ is to normalize the embedding aggregation for each layer. $\mathcal{N}_v$ is the neighbor set of node v. Afterward, the collaborative filtering item embedding is obtained by combining the user representation of different layers $\mathbf{h}_u = \sum_{k=0}^{K} \frac{\mathbf{h}_u^k}{K}$. A similar graph convolution process can be conducted for item representation learning.

2) Teacher model pretraining.: In terms of ranking loss, we use the negative log-likelihood:

$$\mathcal{L}_{rank} = - \sum_{v^+ \in \mathcal{V}_u^+} \log(\sigma(\mathbf{h}_u \cdot \mathbf{h}_{v^+})) - \sum_{v^- \in \mathcal{V}_u^-} \log(1 - \sigma(\mathbf{h}_u \cdot \mathbf{h}_{v^-})), \tag{1}$$

where $\mathcal{V}_u^+$ and $\mathcal{V}_u^-$ are the sets of interacted and non-interacted items, and σ is sigmoid function. During training, 1 negative item v_u^- is sampled for each positive item v_u^+.

In order to generate collaborative filtering embeddings with high-quality representation while enabling efficient knowledge distillation, we combine point-wise ranking loss with alignment ($\mathcal{L}_{align}$) and uniformity regularity ($\mathcal{L}_{uniform}$) as in [6], [18]. And the training loss for our GNN teachers is:

$$\mathcal{L}_{cf} = \gamma \mathcal{L}_{rank} + (1 - \gamma)(\mathcal{L}_{align} + \mathcal{L}_{uniform}), \tag{2}$$

where γ is the loss balancing weight.

C. Linear Content Teacher

The objective of the content teacher is to acquire knowledge from pertinent content profiles associated with users' adoption behavior. Despite the accomplishments attained through incorporating auxiliary information in warm-start recommendations, our empirical investigation in Table II reveals that prevailing content models struggle to exhibit robust generalization capabilities when faced with strict cold-start scenarios, despite being equipped with numerous content features. The underlying reason might be that the complicatedly designed DNN model is prone to prioritize memorizing historical interactions rather than understanding the relationship between key features and interactions.

1) First-order content learning: Towards the above-mentioned limitation of the existing DNN-based content model, instead of concatenating all the content features as model input, we design a simple yet effective linear teacher to focus on learning the first-order feature values for each content material. Specifically, given a set of content features of users/items, we first construct a content embedding look-up table, where each element corresponds to a specific instance of one feature. For example, for the gender of the user, both male and female will be mapped to a content embedding. Note that the content embedding table is randomly initialized, and only the title/name features that contain meaningful semantic information are fixed after processing by the sentence BERT [19]. Then, the l-th content embedding inputs $\mathbf{e}_l$ are first converted to the first-order value by the corresponding feature transformation function $F(\mathbf{e}_l; u, v)$. Then a linear model is constructed to produce the final preference score $R_{cont}(u, v)$ based on the first-order values of all features:

$$R_{cont}(u, v) = \sum_{l=1}^{|\mathcal{F}_{u,v}|} w_l F(\mathbf{e}_l; u, v) + b, \tag{3}$$

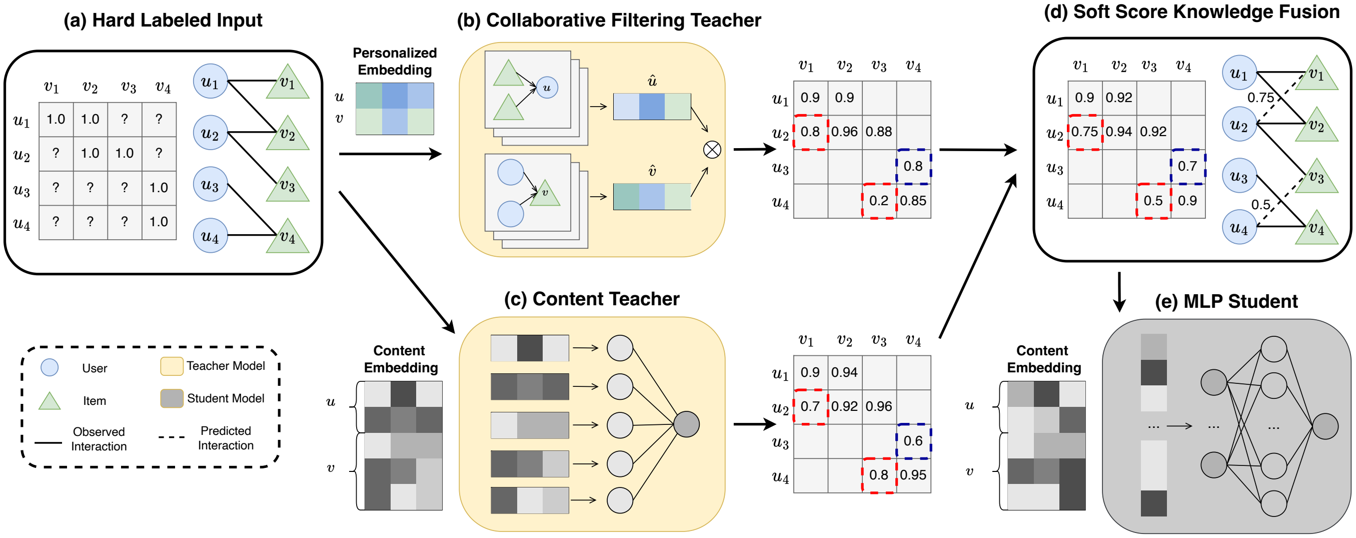


Fig. 2: Framework of DTKD. (a) Hard-labeled input is provided for the two teachers. (b) Collaborative filtering teacher module, where a well-trained GNN model generates the soft score based on the CF knowledge. (c) Content teacher module, where a well-trained linear model produces the soft score from the view of the content knowledge. (d) The soft score knowledge fusion module refines and combines two types of knowledge; The red dashed squares represent the augmented interactions, while the black ones represent the denoised interaction. (e) MLP student, which distills the fused soft score for SCS recommendation.

where $\mathcal{F}_{u,v}$ is the combined feature set of u and v, b is the bias and w_l is the weight of each content feature.

2) Teacher model pretraining.: We directly use Equation 1 as the ranking loss function and adopt the sigmoid function to convert the recommendation score $R_{cont}(u, v)$ into the probability of being relevant $\sigma(R_{cont}(u, v))$.

D. Soft Score Knowledge Fusion

The key motivation of DTKD to use knowledge distillation is to incorporate additional prior of soft scores to smooth the original hard targets (i.e., '1' for positive interactions and '0' for missing interactions in the implicit feedback) so as to maximize the information transferred from two well-learned teachers. We propose this soft score knowledge fusion to distill and fuse two types of knowledge to boost student model's performance. During the distillation process, the quantity (the number of interactions with non-zero scores) and quality (the correctness of interactions of high soft scores) of observed interactions can be improved. Furthermore, the fusion can be regarded as a voting process of two teachers, where they can make an agreement/disagreement to complement each other, thus contributing to a comprehensive knowledge transfer.

1) Soft score definition: The soft score, denoted as $\mathcal{S}_{u,v}$, characterizes the likelihood of a favorable user-item interaction for a given user-item pair (u, v).

2) Augment the unobserved interactions: In the recommender system, the missing/unobserved items in the implicit feedback are regarded as negative samples, which are marked as '0' in the point-wise loss and treated as the opposite of positive items in the pair-wise ranking loss. However, most datasets of the recommender system is collected in an implicit manner. The massive part of the adjacency matrix is still blank, not only because the users are not interested in those items but also due to the missing interactions with those items.

The core idea of using the soft score for the unobserved/missing interactions is that these unseen items could be missing data instead of a dislike signal. Moreover, the soft score can maintain meaningful correlation patterns among user-item pairs and reflect the preference level. Technically, for each positive user-item pair, we randomly sample an unseen item for that user and replace the original hard label '0' of the unobserved user-item pairs with the soft score. Note that we only sample one corresponding unseen item for unbiased learning on both observed and unobserved user-item pairs. This sampling-based soft scoring strategy adds additional supervisory signals for sampled instances, providing better guidance to the student model learning.

3) Denoise the observed interactions: In [20], the authors empirically find that in GNN-based recommendation, active users with rich interactions are poorly modeled compared to inactive users with scarce interactions. Arguably, one could contend that the primary rationale behind this observation is the presence of highly engaged users who exhibit a multitude of noisy interactions, which, in turn, may potentially impede the accurate modeling of user preferences. Additionally, the introduction of additional layers of graph convolutions in the graph model further exacerbates the influx of noise.

In this regard, using a soft score rather than the hard label '1' for the positive items can alleviate the noise from some non-representative interactions. In particular, for every observed user-item pair, we directly use the soft score to replace the original hard label '1'. In the case of an active user purchasing many items, the impact of redundant/noisy interactions could be weakened by replacing the hard labels with reduced soft scores.

4) Voting via knowledge fusion: In the knowledge distillation stage, the outputs of the two teachers are soft score predictions $\mathcal{S}_{teacher}(u, v)$ of positive and sampled negative interactions, which are adjusted by the temperature T:

$$\mathcal{S}_{teacher}(u, v) = \frac{1}{1 + e^{-R(u,v)/T}}, \tag{4}$$

where $R(u, v) = \mathbf{h}_u \cdot \mathbf{h}_v$ for CF teacher soft score ($\mathcal{S}_{cf}(u, v)$), and $R(u, v) = R_{cont}(u, v)$ for content teacher soft score ($\mathcal{S}_{cont}(u, v)$).

The process of combining two types of knowledge can be considered as the voting of two teachers' predictions, i.e., the weighted sum of two soft score outputs of two teacher models:

$$\mathcal{S}(u, v) = \lambda \mathcal{S}_{cf}(u, v) + (1 - \lambda)\mathcal{S}_{cont}(u, v). \tag{5}$$

In Figure 2 (d), the collaborative filtering teacher generates the soft score from the view of historical interactions. The content teacher produces the soft score in regard to the given users' and items' content features. The soft scores from two teachers reach either consensus, e.g., for (u_2, v_1), or disagreement, e.g., for (u_4, v_3). In both situations, the two teachers' predictions complement each other to make a more comprehensive knowledge transfer. At the same time, the number of observed interactions (with the non-zero soft score) is increased to mitigate the data sparsity issue. DTKD also calculates scores for existing links. *E.g.*, for the given (u_3, v_4) user-item pair, two teachers both output relatively lower soft scores, suggesting it is a non-representative/noisy interaction, and DTKD will distill lower scores for the student model. The detailed algorithm design of knowledge fusion is shown in Algorithm 1. Finally, the newly generated soft score matrix will be the input of the student model.

Algorithm 1 Soft score knowledge fusion

```
Input: A well-trained CF-based teacher model $\theta_{cf}$, a well-trained content-based teacher model $\theta_{cont}$, a randomly initialized student model $\theta_s$, a batch of interaction records $\omega_b$, along with the content information $\phi_b$.
Output: A batch of new interaction records with soft scores.
$\hat{\omega}_b \leftarrow \omega_b$
for user-item pair $(u, v) \in \omega_b$ do
    Sample 1 unobserved items for user $u$ into $\hat{\omega}_b$.
end for
Generate CF soft score matrix $\mathcal{S}^b_{cf}$ for $\hat{\omega}_b$ by $\theta_{cf}$
Generate content soft score matrix $\mathcal{S}^b_{cont}$ for $\hat{\omega}_b$ by $\theta_{cont}$
Knowledge fusion: generate $\mathcal{S}^b$ by Equation 5
Update $\hat{\omega}_b$ with $\mathcal{S}^b$
return $\hat{\omega}_b$
```

E. Student Model

The student model is proposed to learn from both content and CF teachers and is generalized for cold users/items. Therefore, we construct a multi-layer perceptron (MLP) student model, which is capable of learning the fused knowledge from two teachers and generating recommendations in SCS situations.

In the training stage, the knowledge distillation loss is formed by minimizing the cross entropy between the MLP student output soft score $\mathcal{S}_{MLP}(u, v) = \sigma(MLP(u, v)/T)$ and the fused soft score from two teachers $S(u, v)$:

$$\begin{aligned} \mathcal{L}_{kd} = -\mathcal{S}(u, v) \cdot \log(\mathcal{S}_{MLP}(u, v)) \\ -(1 - \mathcal{S}(u, v)) \cdot \log(1 - (\mathcal{S}_{MLP}(u, v))), \end{aligned} \tag{6}$$

After trained with $\mathcal{L}_{kd}$, DTKD distills the fused knowledge from both CF teacher and content teacher to the student model. The MLP-based student model is able to learn from both teachers and generalizes to the cold-start scenario.

IV. Experiemnt

This section empirically evaluates the proposed DTKD over one warm-start recommendation and three strict cold-start recommendation tasks on three real-world datasets. The goal is to answer 4 following research questions (RQ).

- **RQ1:** How does DTKD perform compared to other state-of-the-art content models and cold-start models in 4 evaluating settings (one warm-start and three cold-start scenarios)?
- **RQ2:** How effective are the two teacher models and the soft score KD compared with other KD strategies?
- **RQ3:** How does the student model perform compared with the two types of teacher models?
- **RQ4:** What are the impacts of the two hyper-parameters: the CF teacher weight λ and distillation temperature T?

A. Dataset

In this section, we consider three real-world datasets to evaluate our proposed DTKD and other baselines. MovieLens dataset [2] is widely adopted in cold-start recommendation tasks since both ML-1M and ML-100K datasets (summarized in Table I) contain useful auxiliary information about users and items, such as users' age and items' classes/genres. Note that we use sentence BERT [19] to generate content embedding based on the item title as it comprises more meaningful textual information. According to our problem formulation, we separate the original data concerning the states of users and items. For both the user and item sides, we divide 80% of the users/items to warm states and treat the rest 20% of them as in SCS condition. Therefore, we obtain four sub-dataset partitions corresponding to a warm-start and three cold-start conditions. Next, we respectively randomly sample 5% and 20% of user-item interactions of each user as the validation and testing data from the warm state and leave the remaining data as the training dataset. In all four evaluations, if the number of positive items of one user in testing exceeds 5, we only retain five of them for the convenience of performance comparison in different scenarios.

The Yelp dataset [3] is adopted from the Yelp challenge of 2018. Wherein local businesses such as restaurants and bars

[2] https://grouplens.org/datasets/movielens/latest/

[3] https://www.yelp.com/dataset

are regarded as the items. To ensure data quality, we follow [7] to use the 10-core setting [21] (i.e., filtering out users and items with less than ten interactions) to pre-process the data. The statistics of the Yelp dataset after the preprocessing are summarized in Table I. We generate fixed content embeddings only for the item name via sentence BERT [19]. The same data-splitting strategy is adopted as the MovieLens dataset. We only keep at most 20 positive items for each user in testing.

TABLE I: Statistics of the Datasets

Dataset	ML-1M	ML-100K	Yelp
Users	6,040	943	31,668
Items	3,952	1,682	38,048
Features	7	7	16
Interactions	1,000,209	100,100	1,561,406
Sparsity	95.81%	93.7%	99.87%

B. Experimental Setup

1) Baselines: We first consider four state-of-the-art content-based models to highlight the performance of the proposed DTKD on warm-start and different strict cold-start settings. Only a few cold-start methods can work under strict cold-start conditions. Thus we only select two hybrid-based methods to emphasize the effectiveness of our proposed knowledge distillation framework. In addition, a GCN model incorporating the content side information is selected to validate the superiority of the knowledge fusion strategy. We do not choose any meta-learning cold-start algorithm for comparison since they only fit normal cold-start scenarios under the few-shot setting. Due to the design of rank distillation [22]–[24], they cannot directly learn from two teacher models, and we only compare with the embedding-based KD [16] in the ablation study to show the advantages of soft score KD strategy.

- **Wide&Deep** [2] jointly trains wide linear models and deep NNs for the benefits of memorization and generalization.
- **XDeepFM** [10] jointly learns explicit and implicit high-order feature interaction by generating them in an explicit fashion and at the vector-wise level.
- **PNN** [9] is specially designed with a product layer for capturing representation interactive patterns and a fully connected layer to explore high-order feature interactions.
- **DCNV2** [11] keeps the benefits of a DNN of learning implicit features and introduces an improved cross-network.
- **DropoutNet** [13] involves dropout on CF input during training so as to transfer the hybrid knowledge implicitly via enhanced model robustness.
- **Heater** [12] utilizes embedding alignment and randomized training to incorporate hybrid knowledge.
- **ContGCN** is a content-enhanced version of LightGCN [7], which can learn user/item content embeddings via linearly propagating content information on the bipartite graph.

2) Evaluation Metrics: We conduct full ranking for corresponding items in different evaluation scenarios and adopt ranking evaluation metrics, including NDCG@K and Recall@K. Since in 4 evaluation settings of ML-1M, we only retain at most 5 positive items for each user, and thus we set the K as 3 to better evaluate the ranking performance. Similarly, we set the K as 10 in Yelp as the maximum positive item number is 20.

3) Parameter Settings: We implement all the methods using Pytorch [25] and RecBole [26] for a fair comparison. We use Adam [27] as the default learning optimizer and set the maximum number of training epochs as 300. An early stop strategy is adopted if the validation NDCG@K does not increase for 10 epochs. We search the learning rate among [1e-4, 5e-4, 1e-3, 5e-3]. To align with the sentence BERT's output size [19], the content embedding size is set to 384 for each feature. The training batch size is set as 1024 for all datasets. For the CF teacher training loss, we set the weight γ as 0.9. During knowledge distillation, the distillation temperature T is tuned ranging in [2, 4, 6, 8], and the teacher weight λ is tuned within [0.5, 0.6, 0.7, 0.8, 0.9].

C. RQ1: Performance Evaluation

We conducted experiments on 1 (1) warm-start scenario and 3 cold-start scenarios: (2) user SCS, (3) item SCS, (4) and user-item SCS. Three kinds of cold-start scenarios are commonly seen in online recommender systems. The first type can be considered as a recommendation to newly registered users. The second one is used for frequently incoming new items such as news recommendation [5]. The third and most challenging condition is used for starting a new platform, where users and items are all new to each other. Table II demonstrates the performance of our proposed DTKD compared to baseline methods on ML-1M, ML-100K, and Yelp. Beyond empirical results, we calculate the improvements based on DTKD and the strongest baseline. We can observe that:

- In the warm-start evaluation, our DTKD performs the best compared with all the other baselines. It is noteworthy to see the benefits of DTKD on the Yelp dataset, with 57.61% and 38.03% increases on NDCG@10 and Recall@10, respectively. This result reveals that DTKD can effectively distill knowledge from the CF teacher to the student model.
- In the user SCS experiments, DTKD still achieves the best performance in most cases. DropoutNet, Heater, and ContGCN can adapt to those cold users and surpass most content-based baselines on ML-100K and Yelp. These three methods are specially designed with techniques (dropout for DropoutNet, randomized training for Heater, and graph convolution for ContGCN) to enhance the robustness of user-side representation. Therefore, the model design to increase the generalization capability plays an important role in adapting to strict cold users.
- In two cold-item scenarios (item SCS and user-item SCS), it can be found that the DTKD undoubtedly outperforms the alternatives in all cases, yielding dramatic performance improvement on ML-1M and Yelp datasets. In particular, for the item SCS, DTKD surpasses the best baseline by 138.24% and 158.55% regarding the NDCG@3 and Recall@3 on ML-1M, and in the meanwhile, makes large improvements on NDCG@10 by 113.56% and Recall@10

TABLE II: Performance comparison on three datasets (ML-1M, ML-100K, and Yelp) in 4 scenarios (warm-start, user SCS, item SCS, and user-item SCS). The best results are in boldface, and the second-best ones are underlined.

Scenarios	Methods	ML-1M		ML-100k		Yelp	
		NDCG@3	Recall@3	NDCG@3	Recall@3	NDCG@10	Recall@10
Warm user warm item (Warm-start)	Wide&Deep	0.1395	0.0825	0.1756	0.1073	0.0324	0.0394
	xDeepFM	0.1086	0.0644	0.1454	0.0919	0.0359	0.0447
	PNN	0.1107	0.066	0.1511	0.0917	0.0356	0.0427
	DCNV2	0.1394	0.0827	0.1805	0.1109	0.0357	0.0436
	DropoutNet	0.0985	0.0587	0.1811	0.1127	0.0284	0.034
	Heater	0.1035	0.0627	0.1498	0.0912	0.0343	0.0409
	ContGCN	0.1277	0.0774	0.1837	0.1113	0.0368	0.0394
	DTKD	**0.1499**	**0.089**	**0.2088**	**0.1261**	**0.058**	**0.0617**
	Improvement	7.46%	7.62%	13.66%	11.89%	57.61%	38.03%
Cold user warm item (User SCS)	Wide&Deep	0.0504	0.0273	0.0985	0.0543	0.0071	0.0067
	xDeepFM	0.051	0.0298	0.0872	0.0436	0.0003	0.0003
	PNN	0.0566	0.0315	0.1063	0.0574	0.0033	0.0024
	DCNV2	0.0574	0.0318	0.0844	0.0521	0.0068	0.0059
	DropoutNet	0.054	0.0305	0.1029	0.0553	0.0083	0.0079
	Heater	0.0444	0.0262	0.1123	0.0617	0.0102	**0.0098**
	ContGCN	0.0565	0.0313	0.1091	0.0585	0.018	0.0087
	DTKD	**0.0586**	**0.0329**	**0.1135**	**0.0628**	**0.0194**	0.0092
	Improvement	2.09%	3.46%	1.07%	1.78%	7.78%	-6.12%
Warm user cold item (Item SCS)	Wide&Deep	0.0384	0.0233	0.0273	0.016	0.0162	0.0199
	xDeepFM	0.0358	0.0206	0.0567	0.0348	0.0058	0.0065
	PNN	0.0387	0.0234	0.0663	0.0381	0.0059	0.0053
	DCNV2	0.0252	0.0156	0.0515	0.0321	0.0142	0.0162
	DropoutNet	0.0224	0.0133	0.0151	0.0099	0.0058	0.0064
	Heater	0.0152	0.009	0.0351	0.0204	0.0031	0.0031
	ContGCN	0.022	0.0138	0.0272	0.0159	0.0177	0.0175
	DTKD	**0.0922**	**0.0605**	**0.0673**	**0.0402**	**0.0378**	**0.0379**
	Improvement	138.24%	158.55%	1.51%	5.51%	113.56%	90.45%
Cold user cold item (User-item SCS)	Wide&Deep	0.0037	0.0025	0.0078	0.0053	0.0006	0.0008
	xDeepFM	0.0177	0.0113	0.0395	0.0266	0.0004	0.0005
	PNN	0.0071	0.0042	0.0094	0.0064	0.0135	0.0108
	DCNV2	0.0069	0.0038	0.0304	0.0194	0.0005	0.0005
	DropoutNet	0.0145	0.0095	0.0173	0.0117	0.001	0.0009
	Heater	0.0228	0.0119	0.0169	0.0096	0.0012	0.0017
	ContGCN	0.0077	0.0048	0.0225	0.0138	0.0007	0.0006
	DTKD	**0.0877**	**0.0613**	**0.058**	**0.0372**	**0.0279**	**0.0273**
	Improvement	284.65%	415.13%	46.84%	39.85%	106.67%	152.78%

by 90.45% on Yelp. In the extreme case where the historical interaction of both users and items is not presented, the DTKD shows surprisingly boosting performance on all three datasets, and we highlight the improvement on ML-1M of 284.62% and 415.13% for NDCG@3 and Recall@3, respectively. The results validate the efficiency of our proposed DTKD model in transferring two types of knowledge into the extremely strict cold-start scenario.

- Interestingly, the two cold-start methods (DropoutNet, Heater) and the ContGCN are the least competitive in most of the strict cold-item scenarios. These cold-start methods aim to incorporate different modules for transferring CF knowledge and overlook the importance of content information, while the ContGCN cannot preserve all meaningful content knowledge during the CF graph convolution process. This observation differs from the user SCS scenario and highlights the importance of effectively utilizing and transferring content features in item SCS. In situations where the item-side content features are more meaningful and useful, finding ways to leverage these features and transfer content knowledge to address the strict cold-start problems becomes crucial.

D. RQ2: Ablation Study

We conducted the ablation study to investigate the impacts of the distillation strategy and different components of the DTKD framework and to gain deep insight into the performance improvement from DTKD. The complete version of the proposed DTKD is compared with four types of variants: (i) The student MLP model without the linear content teacher (denoted as w/o content teacher), wherein the student model only learns from soft score matrix from CF teacher; (ii) The student MLP model without the GNN-based CF teacher (noted as w/o CF teacher), which learns from the soft score matrix from the content model directly; (iv) The student MLP model learn from the embedding-based KD [16] (denoted as w/o soft score KD) via matching logits layer from the teacher model combined with the hard-labeled learning in Equation 1; (iv)

TABLE III: Ablation study on Movielens-1M dataset. We report DTKD's performance when removing each teacher and the soft score component. The best results are in boldface, and the second-best ones are underlined. Note that we also report the averaged results of the four evaluations to show the trade-offs. N@3 and R@3 represent NDCG@3 and Recall@3, respectively.

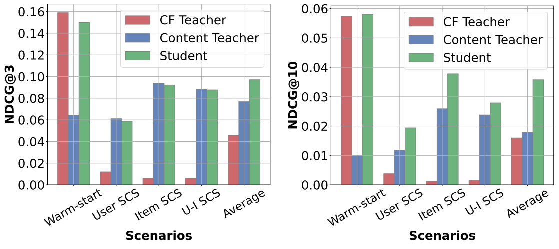
Methods	Warm-start		User SCS		Item SCS		User-item SCS		Average	
	N@3	R@3	N@3	R@3	N@3	R@3	N@3	R@3	N@3	R@3
DTKD	0.1499	0.089	0.0586	0.0329	**0.0922**	**0.0605**	**0.0877**	**0.0613**	**0.0971**	**0.0609**
w/o content teacher	**0.1568**	**0.0931**	0.0585	0.0325	0.0039	0.0026	0.0004	0.0002	0.0549	0.0321
w/o CF teacher	0.0646	0.0406	**0.0628**	**0.0374**	0.0758	0.0524	0.0724	0.0495	0.0689	0.0450
w/o soft score KD	0.1275	0.0756	0.0532	0.0316	0.022	0.013	0.0013	0.0008	0.051	0.0303
w/o KD	0.1411	0.0832	0.0534	0.0301	0.0295	0.0182	0.0035	0.002	0.0569	0.0334

The student MLP model without the knowledge distillation (denoted as w/o KD), which learn from the hard labels via the point-wise ranking loss as in Equation 1.

As shown in Table III, the DTKD generally outperforms other variants in most evaluation settings, especially in achieving the best performance on average of all four scenarios. This indicates the effectiveness of the proposed two teacher modules and soft score knowledge fusion method. To be noted that, it is hard to classify and divide users/items in warm or cold situations in real-life applications. The average performance of all scenarios best matches real-life situations. It is noted that learning solely from the CF teacher (w/o content teacher) obtains the best results on warm-start recommendations, whereas the NDCG@3 and Recall@3 are extremely low (less than 0.001) on two cold item scenarios. In comparison, learning only from the content teacher (w/o CF teacher) gains the strongest performance on user SCS and the second-best on two cold-item evaluations while performing poorly on the warm state, with a large value gap in terms of NDCG@3 and Recall@3. The DTKD can well balance its performances on both warm-state and cold-state conditions and achieves a trade-off to be the average best model. In comparison with MLP model learning from embedding-based KD, the DTKD beats the pure MLP (w/o KD) on all four evaluation settings, while the embedding-based KD brings adverse effects on recommendation performance, demonstrating the significance of the soft score KD and the effectiveness of fusion strategy.

E. RQ3: Student Model vs. Teacher Models

We compare the performance of two teachers and the student model and further investigate the benefits of DTKD. In Figure 3, we demonstrate the MLP student performance compared to two teacher models on ML-1M and Yelp datasets. In the ML-1M, it is observed that the GNN-based CF teacher achieves the best results in the warm state and the lowest results in the cold state, which is vice versa for the content teacher. The student model is competitive with the content teacher on three SCSs and performs close to the CF model in warm conditions, thereby being the best on average of 4 evaluation settings. On the Yelp dataset. our student outperforms the two teachers on all warm-start and cold-start recommendation tasks. It shows the advantage from soft score knowledge distillation of DTKD. Yelp dataset is much sparser than ML-1M, and the soft score knowledge distillation can augment the interaction graph for better student performance.



(a) ML-1M (b) Yelp

Fig. 3: The comparison with two teachers and student model.

F. RQ4: Impacts of Hyperparameters

On ML-1M, we study the impacts of two key hyperparamters, including the CF teacher weight λ, which controls the voting weight of two teachers during knowledge fusion, and the distillation temperature T, which determines the soft level of the soft score distribution.

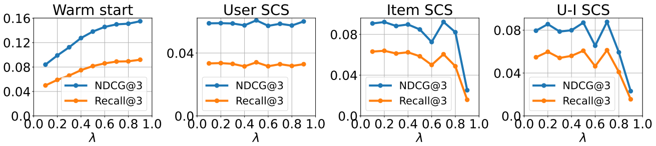


Fig. 4: The impact of CF teachers' weight λ on ML-1M.

1) CF teacher weight λ: We first vary the CF teacher weight λ to adjust the ratio of two teachers' knowledge during the fusion process. Specifically, the weight λ varies from 0.1 to 0.9 with an interval of 0.1. We report the NDCG@3 and Recall@3 results on the ML-1M dataset in Figure 4. Since the CF-based teacher can behave well on the warm-start recommendation compared to the content teacher, it is reasonable that a large weight of CF teacher can bring positive effects. In the warm-start scenario, we can observe that with the increase of CF teacher weight λ, the recommendation quality can be improved and then converge to a level. For user SCS, the performance of DTKD only fluctuates with minor variance. In two cold item scenarios, with an increase of the CF teacher weight T, the NDCG@3 and Recall@3 first oscillate and then start to drop sharply as the ratio of the CF teacher is higher than 0.7. It is interesting to see a larger ratio of CF knowledge can sometimes enhance the cold item recommendation quality. In a nutshell, the above results suggest that a relatively high weight of CF can maintain high performance on both warm-start and cold-start recommendations.

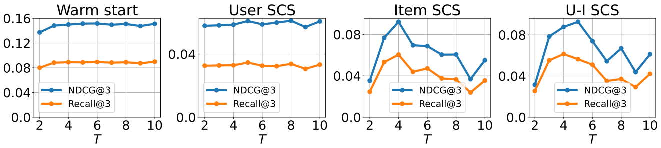


Fig. 5: The impact of distillation temperature T on ML-1M.

2) *Knowledge distillation temperature T:* During the MLP student model training process, one of the most significant factors influencing the knowledge distillation quality is the predefined temperature T. As observed in Figure 5, though the temperature value does not affect the recommendation performance much on warm items, for the cold item evaluations, the NDCG and Recall scores start to increase with the rise of distillation temperature T and reach the maximum value at around T of 4 and then drop to a lower level quickly when T is higher than 6. It makes sense because a small value of temperature generates a soft score approximating the hard label, which cannot bring additional information on user-item correlations, and a large temperature leads all the soft scores to 0.5 losing much preference information.

V. Relate Works

A. Cold-Start Recommendation

Current machine learning online platforms suffer from the data sparsity issue [28], [29], and the cold-start recommendation is one of the most challenging problems. Towards building the cold-start RecSys, existing methods mostly focus on the few-shot cold-start setting and still demand a few interactions for the cold users/items for further adaptation. Inspired by the success of meta-learning algorithms in the few-shot settings, many cold-start methods [14], [30]–[35] follow the learning-to-learn [36] paradigm to improve the cold-start recommendation by improving the learning strategy. They try to exploit the meta-knowledge gained from the warm-start by mimicking the cold-start scenario in the meta-training and then apply such knowledge to the cold-start meta-testing stage. On another line, from model design perspectives, many hybrid approaches are proposed to either learn a mapping function from the content input to the CF embedding [12], [14], [37] or try to implicitly align the content and interaction embedding space jointly [13], [15], aiming at transferring the useful CF and content knowledge towards the cold-start scenarios. Nevertheless, most of these works, especially those meta-learning-based models, fail to work in the SCS conditions [38]. In contrast, pure content-based methods such as Wide&Deep [2] and its variants [9]–[11] can be directly applied in the SCS settings since they merely depend on the user profiles and item contents. Some specific hybrid-based cold-start recommender systems are also enabled to work on SCS settings, as they only take CF input during training, or they can generate CF embedding in the SCS testing. For instance, DropoutNet [13] randomly drops the CF input in the training stage, and the model can adapt to the SCS with improved robustness. Heater [12], on the other hand, is trained to directly convert the content input to the CF embedding in SCS recommendation. However, both the pure content-based models and hybrid-based methods lack an explicit and efficient knowledge transfer design to balance and exploit the complicated knowledge gained in the warm-start scenario. Moreover, they have some inherent limitations during the knowledge encoding process, such as CF/content information loss. By contrast, we explicitly construct a novel knowledge transfer framework DTKD and fully encode and fuse both content and CF information via two separately trained teachers and the soft score knowledge fusion module.

B. Knowledge Distillation in Recommendation

KD has been adopted as an effective method to transfer and compress knowledge from a large teacher model to a smaller student model. It has shown success in various domains, including computer vision [16] and text generation [39]. In the recommendation tasks, it is hard to utilize the KD directly due to the task difference between ranking and classification, and only a few research [22]–[24] focus on how to compress the recommender system model size. With the help of KD, the student model can achieve a similar performance compared to the large teacher model but cannot learn from multiple teachers due to the ranking distillation design. In comparison, we employ the KD as an explicit way to conduct the knowledge fusion and transfer, and we design a dual-teacher structure for the purpose of learning from diverse information sources. Beyond that, this work is the first attempt to use knowledge distillation to resolve the cold-start issue.

VI. Discussion and Conclusion

In this work, we propose a novel dual-teacher knowledge distillation framework to address the strict cold-start recommendation problem. Specifically, to fully utilize the historical interactions and the content profiles, we teach the student model from both GNN-based CF teacher and a linear content teacher. We design a soft scoring mechanism to augment and denoise the original observed interaction graph, which further improves the knowledge modeling quality. Then a score fusion module is proposed to distill and combine the CF and content knowledge to the student model. Extensive experiments conducted on three real-world datasets show that the proposed DTKD significantly outperforms the other state-of-the-art baselines in both warm-start and strict cold-start recommendations. We expect that our solution to exploit the content and interaction information may also bring some positive extensions in different recommendation tasks that suffer from the data sparsity issue.

Acknowledgment

This work is supported in part by NSF under grant III-2106758.

References

[1] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[2] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, “Wide & deep learning for recommender systems,” in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[3] L. Yang, Z. Liu, Y. Wang, C. Wang, Z. Fan, and P. S. Yu, “Large-scale personalized video game recommendation via social-aware contextualized graph neural network,” in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3376–3386.

[4] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[5] M. Trevisiol, L. M. Aiello, R. Schifanella, and A. Jaimes, “Cold-start news recommendation with domain-dependent browse graph,” in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 81–88.

[6] L. Yang, Z. Liu, C. Wang, M. Yang, X. Liu, J. Ma, and P. S. Yu, “Graph-based alignment and uniformity for recommendation,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 4395–4399.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “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*, 2020, pp. 639–648.

[8] L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P. S. Yu, and T. Wang, “Dgrec: Graph neural network for recommendation with diversified embedding generation,” in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 661–669.

[9] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, “Product-based neural networks for user response prediction,” in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 1149–1154.

[10] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, “xdeepfm: Combining explicit and implicit feature interactions for recommender systems,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1754–1763.

[11] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, “Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems,” in *Proceedings of the web conference 2021*, 2021, pp. 1785–1797.

[12] Z. Zhu, S. Sefati, P. Saadatpanah, and J. Caverlee, “Recommendation for new users and new items via randomized training and mixture-of-experts transformation,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1121–1130.

[13] M. Volkovs, G. Yu, and T. Poutanen, “Dropoutnet: Addressing cold start in recommender systems,” *Advances in neural information processing systems*, vol. 30, 2017.

[14] Y. Zhu, R. Xie, F. Zhuang, K. Ge, Y. Sun, X. Zhang, L. Lin, and J. Cao, “Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1167–1176.

[15] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, “Contrastive learning for cold-start recommendation,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5382–5390.

[16] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[17] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, “Efficient knowledge distillation from an ensemble of teachers.” in *Interspeech*, 2017, pp. 3697–3701.

[18] C. Wang, Y. Yu, W. Ma, M. Zhang, C. Chen, Y. Liu, and S. Ma, “Towards representation alignment and uniformity in collaborative filtering,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1816–1825.

[19] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[20] Z. Fan, K. Xu, D. Zhang, H. Peng, J. Zhang, and P. S. Yu, “Graph collaborative signals denoising and augmentation for recommendation,” *arXiv preprint arXiv:2304.03344*, 2023.

[21] R. He and J. McAuley, “Vbpr: visual bayesian personalized ranking from implicit feedback,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[22] J. Tang and K. Wang, “Ranking distillation: Learning compact ranking models with high performance for recommender system,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2289–2298.

[23] J.-w. Lee, M. Choi, J. Lee, and H. Shim, “Collaborative distillation for top-n recommendation,” in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 369–378.

[24] S. Kang, J. Hwang, W. Kweon, and H. Yu, “De-rrd: A knowledge distillation framework for recommender system,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 605–614.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.

[26] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian *et al.*, “Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4653–4664.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[28] H. P. Zou, Y. Zhou, W. Zhang, and C. Caragea, “Decrisismb: Debiased semi-supervised learning for crisis tweet classification via memory bank,” *arXiv preprint arXiv:2310.14577*, 2023.

[29] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence, “Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments,” *arXiv preprint arXiv:1301.2303*, 2013.

[30] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle, “A meta-learning perspective on cold-start recommendations for items,” *Advances in neural information processing systems*, vol. 30, 2017.

[31] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, “Melu: Meta-learned user preference estimator for cold-start recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1073–1082.

[32] M. Dong, F. Yuan, L. Yao, X. Xu, and L. Zhu, “Mamo: Memory-augmented meta-optimization for cold-start recommendation,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 688–697.

[33] Y. Du, X. Zhu, L. Chen, Z. Fang, and Y. Gao, “Metakg: Meta-learning on knowledge graph for cold-start recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[34] B. Hao, J. Zhang, H. Yin, C. Li, and H. Chen, “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*, 2021, pp. 265–273.

[35] Y. Lu, Y. Fang, and C. Shi, “Meta-learning on heterogeneous information networks for cold-start recommendation,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1563–1573.

[36] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[37] O. Barkan, N. Koenigstein, E. Yogev, and O. Katz, “Cb2cf: a neural multiview content-to-collaborative filtering model for completely cold item recommendations,” in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 228–236.

[38] Y. Cao, L. Yang, C. Wang, Z. Liu, H. Peng, C. You, and P. S. Yu, “Multi-task item-attribute graph pre-training for strict cold-start item recommendation,” *arXiv preprint arXiv:2306.14462*, 2023.

[39] Y.-C. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Liu, “Distilling knowledge learned in bert for text generation,” *arXiv preprint arXiv:1911.03829*, 2019.