BPL: Bias-adaptive Preference Distillation Learning for Recommender System

SeongKu Kang*, Jianxun Lian, Dongha Lee, Wonbin Kweon, Sanghwan Jang, Jaehyun Lee, Jindong Wang, Xing Xie, *Fellow, IEEE,* and Hwanjo Yu

***Abstract*—Recommender systems suffer from biases that cause the collected feedback to incompletely reveal user preference. While debiasing learning has been extensively studied, they mostly focused on the specialized (called *counterfactual*) test environment simulated by random exposure of items, significantly degrading accuracy in the typical (called *factual*) test environment based on actual user-item interactions. In fact, each test environment highlights the benefit of a different aspect: the counterfactual test emphasizes user satisfaction in the long-terms, while the factual test focuses on predicting subsequent user behaviors on platforms. Therefore, it is desirable to have a model that performs well on both tests rather than only one. In this work, we introduce a new learning framework, called Bias-adaptive Preference distillation Learning (BPL), to gradually uncover user preferences with dual distillation strategies. These distillation strategies are designed to drive high performance in both factual and counterfactual test environments. Employing a specialized form of *teacher-student distillation* from a biased model, BPL retains accurate preference knowledge aligned with the collected feedback, leading to high performance in the factual test. Furthermore, through *self-distillation* with reliability filtering, BPL iteratively refines its knowledge throughout the training process. This enables the model to produce more accurate predictions across a broader range of user-item combinations, thereby improving performance in the counterfactual test. Comprehensive experiments validate the effectiveness of BPL in both factual and counterfactual tests. Our implementation is accessible via: https://github.com/SeongKu-Kang/BPL.**

***Index Terms*—Preference learning, Knowledge distillation, Biased feedback, Recommender systems**

I. Introduction

Real-world recommender systems form a feedback loop in which the systems' recommendations influence user behaviors, which in turn serve as training data for the system [1]. This feedback loop leads to the creation and amplification of various biases affected by multiple factors, including but not limited to user selection patterns, item exposure mechanism, and influence of public opinions [2], [3]. These biases progressively cause the training data to deviate from users' true preference, ultimately degrading the user satisfaction.

SeongKu Kang is with the Department of Computer Science and Engineering, Korea University, Seoul, South Korea. E-mail:seongkukang@korea.ac.kr.

Jianxun Lian and Xing Xie are with Microsoft Research Asia. E-mail: {jianxun.lian, xingx}@microsoft.com.

Dongha Lee is with the Department of Aritifial Intelligence, Yonsei University, Seoul, South Korea, E-mail:donalee@yonsei.ac.kr.

Wonbin Kweon, Sanghwan Jang, Jaehyun Lee, and Hwanjo Yu are with the Department of Computer Science and Engineering, POSTECH, Pohang, South Korea, Email: {kwb4453, jsh710101, jminy8, hwanjoyu}@postech.ac.kr

Jindong Wang is with William & Mary, Virginia, United States. E-mail:jwang80@wm.edu.

*Corresponding author.

Over the past decade, debiasing learning has been actively studied for various user behaviors, such as explicit feedback [4]–[6], implicit feedback [7], [8], and post-click conversion rate [9]. We focus on explicit feedback, which has additional challenges arising from its limited availability and the fine-grained levels of prediction targets (e.g., 1 to 5 rating scale). Two foundational approaches are inverse propensity score (IPS) [10] and data imputation [4]. IPS reweights data with propensity scores to correct the skewness of training data, while data imputation assigns missing ratings (or errors) for unrated data. Doubly robust (DR) [11] combines the two approaches to leverage their strengths with enhanced robustness, and recently many subsequent methods [5], [6], [12]–[14] have further improved its effectiveness.

However, prior studies on debiasing learning have predominantly focused on the specialized (called *counterfactual*) test environment, without considering its impact in the typical (called *factual*) test environment. Specifically, test environments for recommender systems can be categorized as:

- Factual test [15]–[17] collects test feedback from *actual interactions* between users and items. It is the most widely used setup to assess a model's capability in predicting subsequent user behaviors.
- Counterfactual test [4], [5], [10] collects test feedback via *randomized controlled trials* (RCT), where system-induced biases are eliminated by exposing items randomly. It is a specialized setup to assess a model's debiasing capability.

It is desirable to have a model that performs well in both tests rather than excelling in only one, as each test environment highlights the benefit of a specific aspect [18]; the factual test emphasizes accurate predictions of subsequent behaviors, which are directly connected to service revenue, while the counterfactual test focuses on unbiased predictions, which influence user satisfaction in the long term. A model with high accuracy in both tests can provide accurate predictions not only for randomly selected items (simulated by the counterfactual test) but also for those influenced by recommendation policies and popularity trends (simulated by the factual test), ultimately offering greater benefits for platform deployment.

Though effective in the counterfactual test, existing debiasing learning methods often significantly degrades accuracy in the factual test, resulting in a sub-optimal trade-off between the two test environments. Figure 1 shows the performance of various methods in these environments. Standard training produces a biased model that performs well in the factual test but

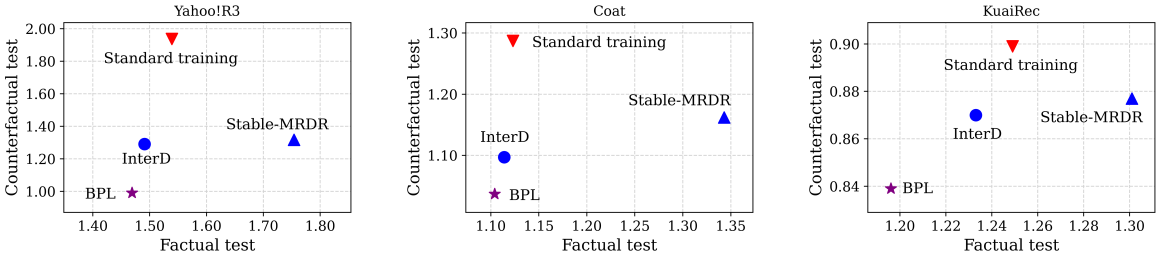

Fig. 1: Mean squared errors of various preference learning methods on factual and counterfactual test across three real-world datasets. We compare four different methods: (a) Standard training without debiasing learning, (b) Stable-MRDR [5], a recent debiasing method, (c) InterD [18], a representative knowledge distillation method, and (Purple) BPL, the proposed method.

poorly in the counterfactual test. Stable-MRDR [5], a representative debiasing learning method, improves performance in the counterfactual test but significantly degrades performance in the factual test. Pointing out these distinct tendencies, a recent method, InterD [18], employs knowledge distillation (KD) from the biased and debiased models by interpolating their predictions.While this approach improves the balance between the two test environments, its effectiveness remains limited by the quality of the pre-trained biased and debiased models used as teachers for distillation. Consequently, this often results in only marginal improvements or, in some cases, performance degradation compared to the teacher models. Furthermore, it is important to note that the teacher models themselves still have considerable room for improvement, as each performs well in only one of the two test environments.

In this work, we introduce a new preference learning framework, called **B**ias-adaptive **P**reference distillation **L**earning (BPL), to gradually uncover underlying preference with dual distillation strategies. First, we formulate our task as a risk minimization problem over all potential user-item combinations, based on risk minimization theory [19]. We then decompose the risk upper bound into three distinct terms, each corresponding to: empirical risk on collected feedback, divergence between collected feedback and the remaining data, and the discriminability of ratings from learned representations. BPL is composed of three major learning objectives, each tailored to minimize one of the upper-bound terms.

A core component of BPL is its dual distillation strategies, designed to enhance rating-discriminability and minimize the third term, which has not been extensively studied in the previous literature. Specifically, BPL employs two types of distillation: (1) **reliability-filtered self-distillation**, which iteratively refines the target model to uncover preferences for unrated data. Utilizing the self-distillation approach [20], we refine the model's knowledge based on its own predictions throughout the training. Equipped with a filtering scheme to identify reliable predictions, the model progressively produces more accurate predictions for an expanding portion of the unrated data. (2) **confidence-penalized preference distillation**, which supplements limited collected feedback using knowledge from the biased teacher model. Employing a specialized form of teacher-student distillation [21], we seek to distill predictions that can be accurately inferred by the biased teacher. To discouraging memorizing patterns that cannot be generalized to the remaining data, we combine the distillation loss with a confidence penalty, allowing the model to learn the teacher predictions while mitigating overconfidence. These two distillation strategies are adaptively balanced based on the affinity of each potential user-item combination to the collected feedback, reflecting the predictive capability of the biased teacher.

These dual distillation strategies enrich preference knowledge within representations, improving rating-discriminability and driving high performance in both factual and counterfactual test environments. Through adaptive distillation from the biased teacher, BPL retains accurate preference knowledge aligned with the collected feedback, leading to high performance in the factual test. Also, rather than relying solely on pre-trained teachers, BPL iteratively refines its knowledge through self-distillation. This allows for progressively uncovering preferences across a broader range of user-item combinations, improving performance in the counterfactual test. Our core contributions are as follows:

- We introduce a new preference learning framework based on risk minimization theory, concurrently optimizing for both factual and counterfactual test environments.
- We propose dual distillation strategies: reliability-filtered self-distillation and confidence-penalized preference distillation, which are adaptively balanced based on the affinity to the collected feedback.
- We validate the effectiveness of BPL through extensive experiments. Also, we provide comprehensive analyses to evaluate the efficacy of each component.

II. Related Work

We introduce previous methods of debiasing learning for explicit feedback and KD, which are closely related to BPL.[1]

Debiasing learning for explicit feedback. Debiasing learning has been actively studied for recommendation. IPS [4], [10], [22] reweights each data with propensity scores to correct the skewness of the training set. The propensity can be computed by using statistics [4] or dedicated models [12], [22]. Also, data imputation [4], which assigns pseudo-labels to unrated data, has been widely studied to learn beyond the biased training set. Furthermore, there have been attempts to utilize special training set [22]–[26], collected by random exposure of items. With the special training set, prior methods have improved the quality of propensity estimation [22], imputation

[1]There are separate research lines for debiasing other behavior types (e.g., implicit feedback, click-through rate), and we guide readers to [1].

value [22], [23], and also achieve a broad debiasing effect [23]. However, obtaining such data can be challenging in real-world applications, as the random exposure policy inevitably degrades user satisfaction [27], [28].

On the one hand, several methods [28]–[30] have formulated the debiasing task as a domain adaptation problem handling distribution shift between the rated (or popular) data and the unrated (or unpopular) data. Employing an adversarial learning approach, they have effectively reduced the divergence between two distributions, mitigating the impact of biases. Furthermore, [30] proposes a new label-conditional clustering to improve representation discriminability. There have also been attempts to directly model the rating process of exposure, selection, and evaluation steps [2], and to use invariant learning to capture environment-invariant preference [31]. A notably dominant approach is DR [11], which combines IPS and data imputation approaches. Many subsequent methods [5], [13], [27], [32]–[34] have further improved its effectiveness and achieved remarkable performance in the counterfactual test environment. However, they often largely degrade accuracy in the factual test environment. BPL does not rely on unbiased training data, and is designed to achieve low errors in both factual and counterfactual tests.

Knowledge distillation. KD is a technique to transfer knowledge from a pre-trained teacher model to improve a target model [21]. For recommender system, KD has been extensively studied to generate a lightweight model that preserves the performance of a massive teacher model [35]–[37]. Leveraging the teacher as ground-truth knowledge sources, existing methods train the target model to mimic the final predictions [35], [36] and intermediate representations [37] from the teacher. On the other hand, a few recent methods [38], [39] utilize distillation with unbiased training data for debiasing. However, as mentioned earlier, a biased model and a debiased model show a clear trade-off in factual and counterfactual test environments, and relying on each of them inevitably leads to poor performance in one of the test environments. Pointing out this problem, InterD [18] introduces a new distillation method that combines the knowledge of biased and debiased models, and transfers it to the target model. While it achieves a better trade-off compared to using one of the models, its effectiveness is often bounded by its teacher models; according to our experiments in Section VI-B, it often fails to bring large improvements compared to the debiased teacher in counterfactual test.

III. Concept and Problem Definition

Definition 1 (Recommendation with explicit feedback). Let $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items, respectively. We focus on explicit feedback where a user u expresses their preference for an item i through multi-level ratings $r_{ui} \in \mathcal{R}$, with $\mathcal{R} = \{1, \dots, K\}$ and higher values indicating stronger preferences. $\mathcal{S} = \mathcal{U} \times \mathcal{I}$ denotes the space of all user-item pairs[2], which can be divided into a subspace of rated pairs $\mathcal{S}^1$ and a subspace of unrated pairs $\mathcal{S}^0$. Let s_{ui} denote a Bernoulli random variable representing the presence of (u, i) pair in $\mathcal{S}^1$, i.e., $s_{ui} = 1$ if $(u, i) \in \mathcal{S}^1$, otherwise 0. Our task is to predict the potential outcome when s_{ui} had been set to 1, i.e., if an item i had been rated by a user u, what would be the feedback?

[2]In this work, we use the terms "data" and "pairs" interchangeably.

Definition 2 (Preference learning as risk minimization). Let $f : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$ denote a recommendation model. If ratings are provided for all user-item pairs in $\mathcal{S}$, we can train the model to minimize the following *true risk*:

$$\mathcal{L}_{true} = |\mathcal{S}|^{-1} \sum_{(u,i)\in\mathcal{S}} \ell(f(u, i), r_{ui}), \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes the error function. We refer to knowledge that can minimize this true risk, which allows for accurate recommendations for all user-item pairs, as *true preference*.

As this true risk is not accessible, the model is typically optimized to minimize the empirical risk: $\mathcal{L}_{emp} = |\mathcal{S}^1|^{-1} \sum_{(u,i)\in\mathcal{S}^1} \ell(f(u, i), r_{ui})$. However, due to various biases, this approach falls short of capturing the true preference. To elaborate, the collected ratings are missing-not-at-random as users do not rate items uniformly; users tend to select items that they like and also actively rate particularly bad or good items (selection bias) [40]. The rating process is also affected by various factors such as item popularity and recommendation policy (exposure, position bias) [26], [41]. Additionally, user judgment is influenced by public opinions, leading to different distributions when users rate items before or after being exposed to public opinions (conformity bias) [3].

Definition 3 (Factual and counterfactual test). Test environments for recommender systems can be grouped into two categories [18]: (1) Factual test collects feedback from actual user-item interactions. (2) Counterfactual test collects feedback via randomized controlled trials (RCT), where system-induced biases are eliminated by exposing items randomly. Because the factual test relies on actual interactions, popular items that users tend to engage with more frequently are often overrepresented in the evaluation. In contrast, the counterfactual test offers an alternative evaluation perspective to a broader range of items, including niche or less popular ones.

Note that the user-item pairs collected in each test complementarily cover $\mathcal{S}$: the factual test covers unseen pairs closer to the collected feedback (i.e., $\mathcal{S}^1$), while the counterfactual test includes unseen pairs more widely distributed across the space. A model that achieves high accuracy in both tests can provide accurate recommendations for items selected both randomly and through various factors (e.g., popularity trends), offering greater benefits to the platform.

Problem formulation. Given a set of rated data in $\mathcal{S}^1$, our goal is to generate a model that minimizes the true risk, performing well on both factual and counterfactual test environments.

IV. Preliminaries

A. *Risk upper bound on $\mathcal{S}$*

To develop a systematic approach to minimize true risk, we first define the upper bound of the expected errors using risk minimization theory [19]. As this theory is based on properties of learned representations, we view the recommendation model as the combination of an encoder and a predictor

$f = \eta \circ \phi$, where the encoder $\phi : \mathcal{U} \times \mathcal{I} \to \mathcal{Z}$ encodes each pair into the representation space, and the predictor $\eta : \mathcal{Z} \to \mathcal{R}$ predicts ratings from the representation.

Theorem 1 [19] *Let $\mathcal{H}$ be the hypothesis class of predictors on representation space. For $\eta \in \mathcal{H}$, let $\epsilon_{\mathcal{S}}(\eta)$ denote the expected errors on $\mathcal{S}$, i.e., $\epsilon_{\mathcal{S}}(\eta) = |\mathcal{S}|^{-1} \sum_{(u,i)\in\mathcal{S}} \ell(f(u,i), r_{ui})$. $\mathcal{Z}_{\mathcal{S}^0}$ and $\mathcal{Z}_{\mathcal{S}^1}$ denote the distributions of $\mathcal{S}^0$ and $\mathcal{S}^1$ over $\mathcal{Z}$, respectively. Given $\lambda_0 = \frac{|\mathcal{S}^0|}{|\mathcal{S}|}$, which is the ratio of unrated data in $\mathcal{S}$, $\forall \eta \in \mathcal{H}$:*

$$\epsilon_{\mathcal{S}}(\eta) \leq \epsilon_{\mathcal{S}^1}(\eta) + \lambda_0 \left(\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{Z}_{\mathcal{S}^0}, \mathcal{Z}_{\mathcal{S}^1}) + (\epsilon_{\mathcal{S}^0}(\eta^*) + \epsilon_{\mathcal{S}^1}(\eta^*)) \right) \quad (2)$$

The upper bound of the expected error consists of three terms:

T1: $\epsilon_{\mathcal{S}^1}(\eta)$: the error on rated space $\mathcal{S}^1$.

T2: $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{Z}_{\mathcal{S}^0}, \mathcal{Z}_{\mathcal{S}^1})$: the $\mathcal{H}\Delta\mathcal{H}$-divergence [19] which is defined as $2 \sup_{\eta,\eta' \in \mathcal{H}} \left| p_{z\sim\mathcal{Z}_{\mathcal{S}^0}} [\eta(z) \neq \eta'(z)] - p_{z\sim\mathcal{Z}_{\mathcal{S}^1}} [\eta(z) \neq \eta'(z)] \right|$. It indicates the discrepancy between $\mathcal{S}^0$ and $\mathcal{S}^1$ over $\mathcal{Z}$.

T3: $\epsilon_{\mathcal{S}^0}(\eta^*) + \epsilon_{\mathcal{S}^1}(\eta^*)$: the combined error of the ideal hypothesis, i.e., $\eta^* = \arg\min_{\eta\in\mathcal{H}} (\epsilon_{\mathcal{S}^0}(\eta) + \epsilon_{\mathcal{S}^1}(\eta))$, on $\mathcal{S}^0$ and $\mathcal{S}^1$.

T1 can be minimized by standard training with ground-truth ratings, and T2 can be approximated and minimized by adversarial learning [42]. Combining standard training (T1) and adversarial learning (T2) has achieved huge success in handling unlabeled data for domain adaptation [42] and has also been applied for recommender systems [28]–[30].

The remaining term is T3, which is the main focus of this paper. T3 measures discriminability of representations [30], [43], which indicates the easiness of distinguishing different ratings (i.e., $\mathcal{R}$) by a supervised predictor trained over the representations. To reduce T3, the representations should contain sufficient preference information that allows for distinguishing ratings based on them. Unlike T1 and T2, directly approximating T3 is infeasible due to the absence of ratings in $\mathcal{S}^0$.

B. *Motivational Analyses*

We present our analysis showing that the existing learning strategies have limitations in minimizing risk across $\mathcal{S}$. Furthermore, we discuss how a biased model can aid in learning more accurate preference. We train the recommendation model using the biased training set and report the results on the RCT test set. The results on Yahoo!R3 dataset are reported, with similar findings observed across other datasets.

$\mathcal{S}^1$-affinity. As users continue to utilize the system, some data in $\mathcal{S}^0$ will be naturally rated. Given that ratings are missing-not-at-random, unrated data have varying probabilities of being rated potentially, i.e., $p(s_{ui} = 1)$. This probability reveals how similar each data point is to the already collected feedback, and we leverage it for analysis. To estimate the probability, we train a binary classifier to distinguish whether each pair (u, i) comes from $\mathcal{S}^0$ or $\mathcal{S}^1$ using the binary cross-entropy loss.[3] We refer to $\hat{p}(s_{ui} = 1)$ as $\mathcal{S}^1$-affinity of each unrated data.

[3] $\mathcal{L} = -\sum_{(u,i)\in\mathcal{S}} s_{ui} \log \hat{p}(s_{ui} = 1) + (1 - s_{ui}) \log(1 - \hat{p}(s_{ui} = 1))$. We use a one-layer MLP with a sigmoid activation function (Appendix A).

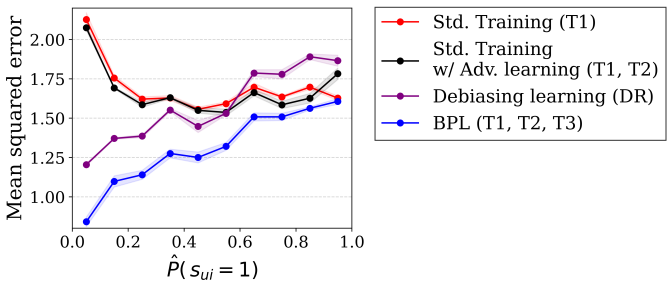

Fig. 2: Relationships between $\mathcal{S}^1$-affinity (i.e., $\hat{p}(s_{ui} = 1)$) and model prediction errors with varying learning strategies.

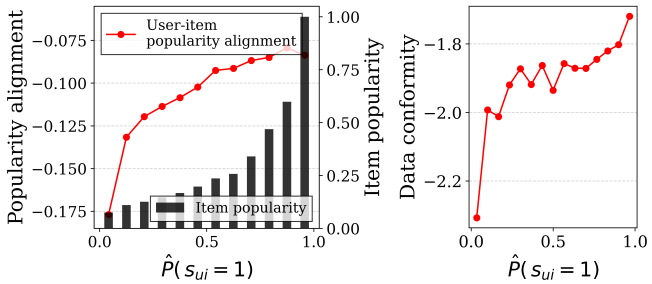

Fig. 3: Relationships between $\mathcal{S}^1$-affinity (i.e., $\hat{p}(s_{ui} = 1)$) and three factors that affect rating process.

Analysis 1: Impact of learning strategy on prediction errors. We first analyze the model prediction errors from four different learning strategies: **(1)** the standard training on $\mathcal{S}^1$ (T1), **(2)** the standard training with adversarial learning (T1 + T2), **(3)** the representative debiasing learning with doubly-robust (DR) [5], **(4)** BPL, the proposed approach, designed to minimize all three terms of the risk upper bound. Note that DR employs different approaches that leverage propensity scores, and is not directly linked with the three terms in Eq. 2.

Figure 2 presents prediction errors according to the affinity. We observe that the debiasing learning and standard training show highly different tendencies. For data with low affinity, debiasing learning significantly reduces the errors. However, this improvement comes at the cost of largely degraded accuracy on high-affinity data. DR employs propensity scores that reweight each data to prevent over-reliance on the collected feedback, which can rather hinders learning on high-affinity data. On the other hand, while the standard training generates a biased model with high errors for data with low affinity, it consistently achieves low errors for a substantial amount of high-affinity data. Lastly, although adversarial learning reduces errors, its effectiveness remains limited, indicating that simply minimizing overall divergence is insufficient.

These observations reveal two insights: (i) The biased model's knowledge of high-affinity data holds potential for improvement by complementing the limited feedback and enriching the supervision for preference learning. (ii) The $\mathcal{S}^1$-affinity score provides a strategic foundation for the selective utilization of biased model knowledge by distinguishing the data that the biased model can aptly comprehend.

Analysis 2: $\mathcal{S}^1$-affinity and rating process-related factors. To gain a deeper insight into why the biased model achieves low errors on high-affinity data, we explore the relationships

between $\mathcal{S}^1$-affinity and three factors that can affect users' rating process[4]: **(1) item popularity** denotes the normalized interaction frequency of each item. It is well known that popularity strongly influences the exposure and selection process [2]. **(2) user-item popularity alignment** denotes the alignment between the average popularity of the user's historical items and the popularity of each item.[5] This factor jointly considers both user-side and item-side popularity. For instance, "blockbuster fan-indie film" has a low alignment value due to their dissimilar popularity levels. **(3) data conformity** measures how closely each user's opinion aligns with the public opinion [3].[6] For all three factors, a higher value indicates greater popularity, alignment, and conformity.

Figure 3 presents the results. Both item popularity and popularity alignment show strong positive correlations with $\mathcal{S}^1$-affinity. This shows that both $\mathcal{S}^1$ and high-affinity data lean towards popular items and users who prefer popular items, and they are likely to be similarly affected by popularity-related factors. Also, data conformity shows a strong correlation with $\mathcal{S}^1$-affinity, suggesting that public opinion generally has a stronger impact on the evaluation of high-affinity data, compared to data deviating from $\mathcal{S}^1$. While specifying the impacts of each bias is not our focus, a possible explanation is that users tend to encounter public opinion more frequently for popular items. These results show that $\mathcal{S}^1$ *and high-affinity data have similar characteristics in terms of various factors affecting the users' rating process*, which also supports the low errors on high-affinity data by the biased model.

These analyses motivate us to selectively leverage the biased model's knowledge, which aids in predicting user preferences for high-affinity data, thereby enhancing preference learning.

V. Methodology

We propose BPL, designed to effectively reduce the risk upper bound on user-item space. BPL employs three major learning objectives, each of which focuses on reducing each term of the upper bound. Specifically, BPL trains the model using $\mathcal{L}_{T1}$ to learn preference from rated data (SectionV-A), $\mathcal{L}_{T2}$ to reduce the divergence between $\mathcal{S}^0$ and $\mathcal{S}^1$ in the representation space (Section V-B), and $\mathcal{L}_{T3}$ to discover preference for unrated data via dual distillation strategies that adaptively leverage the biased model knowledge (Section V-C). These three objectives are jointly optimized to reduce the upper bound (Section V-D). Figure 4 presents the overview of BPL.

A. *Preference learning with rated data (T1)*

We first learn preference with ground-truth ratings from users, by minimizing T1. For encoder ϕ of the model f, various architectures can be flexibly employed, from ID-based [44] to graph-based [17] encoder. For predictor η, we use a K-class softmax classifier, where each class corresponds to each rating level, and $\hat{p}(r_{ui}=k)$ denotes the softmax probability for class k. We use the supervised loss as:

$$\min_{\phi,\eta} \mathcal{L}_{T1} = |\mathcal{S}^1|^{-1} \sum_{(u,i)\in\mathcal{S}^1} \ell_s(f(u,i), r_{ui}), \tag{3}$$

where $\ell_s = -\sum_{k\in\{1,\dots,K\}} \mathbb{1}[r_{ui}=k] \log \hat{p}(r_{ui}=k)$. The predicted rating $\hat{r}_{ui}$ is obtained by the expected value based on the softmax probabilities [17]: $\hat{r}_{ui} = \mathbb{E}_{\hat{p}}[r_{ui}] = \sum_k k\,\hat{p}(r_{ui}=k)$. This choice allows us to readily control the prediction confidence while preserving the accuracy, which will be explained in Section V-C2.

As discussed in Section IV-B, due to various biases in $\mathcal{S}^1$, this approach often falls short of capturing true preferences. The remaining parts of BPL are dedicated to alleviating this problem based on the risk upper bound (Section IV-A).

B. *Distribution alignment learning (T2)*

T2 corresponds to the divergence between $\mathcal{S}^0$ and $\mathcal{S}^1$ in the representation space. We adopt adversarial learning which has been extensively studied in domain adaptation [30], [42]. It trains a discriminator to distinguish representations from different domains, while training the encoder to generate representations that the discriminator cannot distinguish. By doing so, the encoder reduces the divergence between two domain distributions in the representation space [42].

However, we empirically found that treating $\mathcal{S}^0$ and $\mathcal{S}^1$ as independent domains hinders effective optimization. In fact, two spaces are not inherently independent; as users continue to utilize the system, some data in $\mathcal{S}^0$ will be rated. Naturally, the discriminator struggles to find a clear decision boundary that separates high-affinity data from $\mathcal{S}^1$. Such degraded discrimination capability subsequently limits the encoder's ability to achieve alignment, thereby hindering effective optimization. This issue is consistent with the findings in [45] that training a discriminator to distinguish already well-aligned representations leads to suboptimal results.

As a solution, we borrow the idea from [45], which relabels a small set of samples near the decision boundary in adversarial learning. Specially, we expand $\mathcal{S}^1$ by adding a small subset of unrated data with high $\mathcal{S}^1$-affinity. Let $\mathcal{S}^{01}$ denote a set of unrated data with the highest $x\%$ affinities. Given a representation z_{ui}, we train a discriminator $f_d : \mathcal{Z} \to [0,1]$ to distinguish whether z_{ui} comes from the expanded space $\mathcal{S}^1 \cup \mathcal{S}^{01}$ or the remaining unrated data. The encoder ϕ is trained to generate representations that prevent f_d from distinguishing them:

$$\begin{aligned}\min_{\phi}\max_{f_d} \mathcal{L}_{T2} = &\sum_{(u,i)\in\mathcal{S}^1\cup\mathcal{S}^{01}} \log(f_d(z_{ui})) \\ &+ \sum_{(u,i)\in\mathcal{S}^0\setminus\mathcal{S}^{01}} \log(1-f_d(z_{ui}))\end{aligned} \tag{4}$$

$\mathcal{L}_{T2}$ corresponds to the empirical $\mathcal{H}$-divergence between two distributions, which empirically approximates $\mathcal{H}\Delta\mathcal{H}$-divergence [42]. We identify unrated data most likely to be rated potentially and treat them as (pseudo) rated data. This strategy enables the discriminator to find a more separable decision boundary, which in turn makes the encoder better align the distributions [45].

[4]It is infeasible to specify individual factors that affect the rating process. We analyze three factors that are widely recognized in prior literature.

[5]$\mathrm{sim}(pop(u), pop(i))$, where $pop(u) = \mathrm{AVERAGE}(\{pop(i)\}_{s_{ui}=1})$. We use the negative squared difference for the similarity function, i.e., $\mathrm{sim}(x,y) = -(x-y)^2$.

[6]$\mathrm{sim}(r_{ui}, \bar{r}_i)$, where $\bar{r}_i$ is the averaged training rating for item i.

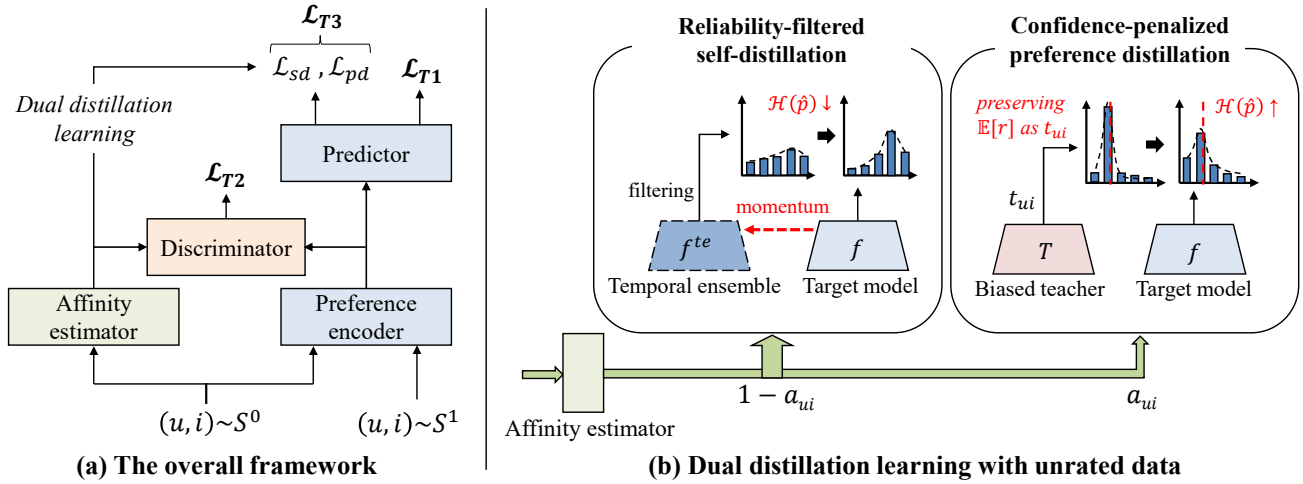

Fig. 4: Illustration of BPL. (a) The overall framework, (b) Dual distillation learning with unrated data. Best viewed in color.

C. *Dual distillation learning with unrated data (T3)*

As discussed in §IV-A, T3 reflects the easiness of discerning ratings from representations. That is, representations should include sufficient preference information that allows for distinguishing ratings based on them. However, due to the skewness and limited availability of ground-truth ratings, relying on $\mathcal{L}_{T1}$ falls short of finding preference for unrated data.

As a solution, we introduce dual distillation strategies that iteratively refine the target model, aided by the adaptive usage of biased model knowledge. We first introduce **reliability-filtered self-distillation** (Section V-C1) to uncover preference for unrated data by iteratively refining the model based on its predictions. To facilitate accurate preference discovery, we propose **confidence-penalized preference distillation** (Section V-C2) that supplements limited ground-truth ratings with the aid of a biased teacher model. The two distillations are balanced based on $\mathcal{S}^1$-affinity of each data (Section V-C3).

*1) **Reliability-filtered self-distillation**:* Self-distillation is a special type of distillation where a model is used to generate "soft labels" for its own training data [20], [46]. Instead of using external teacher models, the model learns from its own predictions, typically probability distributions over classes, iteratively refining its knowledge. This approach has proven effective in improving generalization by directly leveraging previously discovered hidden patterns that are not directly revealed by labeled data [20]. Employing the self-distillation approach, we aim to generate predictions for unrated data and refine the model's preference knowledge over time.

However, unlike typical classification tasks, the rated data is highly limited and skewed, causing the model to generate many unreliable predictions. To address this problem, we introduce a *filtering scheme* to identify reliable predictions. Through filtering, we selectively refine the model's preference knowledge by leveraging only the reliable predictions. As training progresses, the model increasingly produces accurate predictions for a growing number of unrated data, discovering preferences across a broader unrated space.

Reliability filtering. As not all predictions are accurate, it is crucial to leverage reliable ones selectively. To develop an effective filtering scheme tailored to BPL, we examine two distinct metrics that have proven effective for assessing the reliability of model predictions:

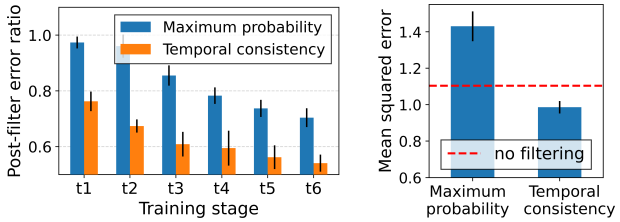

Fig. 5: Comparison of filtering effects. (left) post-filtering error ratio, which denotes the ratio of average errors in the filtered dataset to the average errors in the total dataset, (right) recommendation performance. Results on Yahoo!R3-RCT.

- **Maximum probability** is a widely used metric [47], where a prediction is considered reliable if its maximum class probability exceeds a certain threshold, i.e., $\max_k \hat{p}(r_{ui} = k) > m$. We tune the threshold m from 0.1 to 0.9 based on validation performance.
- **Temporal consistency** assesses how consistent the prediction is during training, supported by findings that inconsistent predictions that frequently change often contain large errors [48]. To evaluate consistency, we employ a temporal ensemble of the model, as done in [48]. This temporal ensemble f^{te} is updated as the moving average of f with a discount factor $\tau \in [0,1]$, i.e., $\theta_{f^{te}} = \tau\theta_{f^{te}} + (1-\tau)\theta_f$. By setting a large value for τ, f^{te} slowly approximates f, providing a temporally consistent baseline. A prediction is considered reliable if it agrees with f^{te}, i.e., $\arg\max_k \hat{p}^{te}(r_{ui}) = \arg\max_k \hat{p}(r_{ui})$.

Figure 5 shows that temporal consistency is better suited for identifying accurate predictions with smaller errors in various training stages, ultimately leading to improved recommendation performance. Further analysis revealed that the maximum probability varies significantly according to the $\mathcal{S}^1$-affinity of each unrated data. The average maximum probability for the unrated data with the lowest 50% affinity was over 20% lower than that observed for the top 50% affinity. This large disparity

can make maximum probability less effective as an indicator. On the other hand, temporal consistency is not directly influenced by the magnitude of probability, making it more robust to varying affinities. Based on these results, we employ temporal consistency as a reliability metric for filtering. For further analysis, refer to Appendix C.

Self-distillation. Using reliable predictions identified through the filtering, we refine the model's preference knowledge. Specifically, we train the model to make more confident preference predictions for the unrated data that it has reliably predicted previously. This is achieved by reducing the entropy of the predicted distribution, which is a well-established technique for enhancing prediction confidence [49]:

$$\ell_{sd} = \mathbb{1}[\arg\max_k \hat{p}^{te}(r_{ui}) = \arg\max_k \hat{p}(r_{ui})] H(\hat{p}(r_{ui})), \quad (5)$$

where $\mathbb{1}[\cdot]$ is the indicator function and the entropy is defined as $H(\hat{p}(r_{ui})) = -\sum_k \hat{p}(r_{ui}=k)\log\hat{p}(r_{ui}=k)$. By iterating this process, the model undergoes gradual refinement based on reliable predictions, acquiring preference knowledge for an increasing number of unrated data throughout the training.

*2) **Confidence-penalized preference distillation**:* To facilitate more accurate preference discovery, we propose to use the biased teacher model. As shown in Section IV-B, the biased teacher can make accurate predictions for potential preference on high-affinity data. We exploit these predictions as additional supervision for model training, supplementing limited rated data and improving its preference knowledge. The biased teacher can be any recommendation model that is trained without debiasing learning.

However, naive training on high-affinity data can lead to reliance on specific patterns within the subset (e.g., high conformity to public opinion, as discussed in Section IV-B) which cannot be generalized to the remaining data. Noticing this issue, we propose a new confidence-penalized distillation. We combine the distillation loss with a confidence penalty, guiding our model to preserve the exact teacher prediction while preventing it from becoming overly confident.

$$\ell_{pd} = \lambda(\mathbb{E}_{\hat{p}}[r_{ui}] - t_{ui})^2 - H(\hat{p}(r_{ui})) \quad (6)$$

The first term distills the biased teacher prediction t_{ui} by compelling our model to produce an output distribution with the expected value of t_{ui}. The second term applies the confidence penalty that discourages distributions with low entropy. λ is a hyperparameter to balance two terms.

This confidence penalty is based on *confidence regularization* for classifiers [50], where overconfident outputs are known to lead to poor generalization. In our context, it helps prevent the model from overfitting to the biased teacher's overly certain outputs, particularly for high-affinity samples. As will be shown in our ablation study (Section VI-C1), this penalty plays a crucial role in ensuring effective distillation. Without it, the model tends to overfit to the teacher's biased predictions, resulting in significantly degraded accuracy on the counterfactual test.

*3) **Soft and hard combination of dual distillations**:* We balance two distinct distillation strategies based on the $\mathcal{S}^1$-affinity of each unrated data. We introduce soft and hard combinations. The soft combination integrates two strategies according to the predicted affinity: $a_{ui} = \hat{p}(s_{ui}=1)$, whereas the hard combination exclusively uses one strategy: $a_{ui} = 1$ if $(u,i) \in \mathcal{S}^{01}$, otherwise 0.

$$\min_{\phi,\eta} \mathcal{L}_{T3} = |\mathcal{S}^0|^{-1} \sum_{(u,i)\in\mathcal{S}^0} a_{ui}\,\ell_{pd}(u,i) + (1-a_{ui})\ell_{sd}(u,i) \quad (7)$$

BPL is trained using either the soft or hard combination, which we call BPL-Soft and BPL-Hard, respectively.

D. The overall training objective

The final training loss of BPL is as follows:

$$\min_{\phi,\eta} \max_{f_d} \mathcal{L}_{T1} + \alpha\mathcal{L}_{T2} + \beta\mathcal{L}_{T3}, \quad (8)$$

where α and β are hyperparameters balancing the losses. The $\mathcal{S}^1$-affinity is precomputed before the training, as explained in Section IV-B. Note that we only use the backbone model f for inference, thus incurring no extra inference costs. The training algorithm is presented in Appendix B. We also provide a detailed hyperparameter study in Section VI-C4.

VI. Experiments

A. Experiment setup

Datasets. We use three real-world datasets: Yahoo!R3 [51], Coat [10], and KuaiRec [52]. They contain pre-split **(a) biased training set** from actual user-item interactions, and **(b) counterfactual test set**.[7] As discussed in Section III, an ideal model for deployment should achieve high accuracy in both factual and counterfactual test environments [18]. For comprehensive evaluation in both tests environments, we randomly split out 10% of the biased training set to form **(c) factual test set**.

TABLE I: Data statistics after preprocessing.

	#User	#Item	#Ratings		
			(a) training	(b) counterfactual test	(c) factual test
Yahoo!R3	15,400	1,000	280,534	54,000	31,170
Coat	290	300	6,264	4,640	696
KuaiRec	7,176	10,728	11,277,725	4,676,570	1,253,081

Evaluation setup. In this work, we specifically target multi-level explicit feedback where a special unbiased training set is not available, which is the identical setup as [4], [28]. We employ two metrics widely used for rating prediction [28]: Mean Squared Error (MSE) and Mean Absolute Error (MAE). We hold out 10% of the training set as the validation set. We report the average value of five independent runs. Note that we do not binarize ratings, as our focus is on multi-level feedback.

Compared methods. We focus on evaluating a model's capability to handle both factual and counterfactual tests simultaneously. We exclude methods that require unbiased training data from the baselines [22]–[24], [38], as we focus on scenarios without such data.

[7]The counterfactual test data is obtained via RCT (Yahoo!R3, Coat) and full exposure of test items (KuaiRec). For KuaiRec, we convert the watch ratio to 1-5 rating scale to maintain consistency with other datasets [52].

As our main competitors, we include recent methods which can be grouped into four relevant categories. The first group follows **typical training** without any debiasing techniques.

- **Standard Training** optimizes the model to minimize empirical risks on the collected feedback ($\mathcal{L}_{T1}$).

The second group includes advanced **adversarial learning**:

- **FADA [53]** proposes fine-grained adversarial learning for class-level feature alignment. It uses pseudo-labeling on the unlabeled target domain and aligns class-level distributions through a fine-grained discriminator.
- **IA [54]** introduces an inference adjustment strategy to mitigate label distribution shift by correcting posterior probabilities. This adjustment is applied alongside FADA.

The third group includes **debiasing learning** methods. We compare BPL with three recent methods that achieve highly competitive performance in the counterfactual test.

- **Stable-DR [5]** proposes a stabilized doubly robust learning framework that cyclically updates imputation, propensity, and prediction models.
- **Stable-MRDR [5]** stabilizes an enhanced variant of doubly robust learning [55], designed to reduce variance while preserving double robustness.
- **DCE-TDR [14]** introduces a doubly calibrated estimator that calibrates both the imputation and propensity models.

The last group includes **knowledge distillation** methods:

- **InterD [18]** integrates knowledge from two teacher models—one biased and one debiased—by interpolating their predictions and distills the combined knowledge.
- **BPL** is our proposed method that employs dual distillation strategies: reliability-filtered self-distillation and confidence-penalized preference distillation. We present results for two variants, BPL-Soft and BPL-Hard, which differ in their loss balancing strategies (Eq.7).

For the **biased teacher** in both InterD and BPL, we employ a recent model [16] that shows superior performance on the factual test. For the debiased teacher in InterD, we utilize the best-performing debiasing method for each dataset.

Implementation details. For backbone models, we employ the ID-based model [44] for main experiments, following [4], [5], [28]. We report results with other backbone models in Section VI-C5. We first tune the embedding size in $\{16, 32, 64, 128\}$ with the standard training on the validation set, and fix the size for all baselines. The selected sizes are 64 (Yahoo!R3, KuaiRec), 32 (Coat), and 16 (Simulation). Implementation details for baselines are provided in Appendix F.

We implement BPL by using PyTorch with CUDA. All hyperparameters are tuned using grid search on the validation set. For affinity estimation, we train a simple binary classifier with binary cross-entropy loss (Section IV-B). The classifier is a one-layer MLP with a sigmoid activation function. We use Adam optimizer where the learning rate and weight decay are chosen from $\{10^{-6}, ..., 10^{-1}\}$. We set $\lambda = 1.0$, $\tau = 0.999$, and tune α, β in $\{0.5, ..., 2.0\}$ and x between 1 and 30. We provide a detailed hyperparameter study and guidelines for choosing values in Section VI-C4.

B. *Results on factual and counterfactual test environments*

We investigate the effectiveness of BPL in both factual and counterfactual test environments. Figure 6 presents the MSE and MAE results for both factual and counterfactual tests. Furthermore, in Figure 7, we report the harmonic mean of the scores on the factual and counterfactual tests to provide an overall comparison of a model's capability [18]. For readability, we report the best performance from either FADA or FADA with IA (denoted as FADA/IA) and from either Stable-DR or Stable-MRDR (denoted as Stable-DR/MRDR).

- The biased teacher shows superior performance in the factual test, but its performance is largely degraded in the counterfactual test. Conversely, debiasing learning (Stable-DR/MRDR, DCE-TDR) improves the performance on the counterfactual test, but it largely degrades performance on the factual test. This highlights the limited capability of the existing preference learning methods in minimizing errors across all user-item data.
- Advanced adversarial learning that performs class-level alignment (FADA/IA) achieve improvements to some extent. They also cause less performance degradation on factual tests. However, their effectiveness in counterfactual tests remains limited compared to debiasing methods.
- InterD achieves a good balance between both tests by combining the knowledge from the biased and debiased teachers. However, it often fails to bring large improvements compared to its debiased teacher (i.e., Stable-DR/MRDR). This shows the limitation of solely relying on fixed teacher knowledge for preference learning.
- BPL achieves the best overall balance between both tests. It shows low errors in the factual test, and also the lowest errors in the counterfactual test. We attribute the performance gain to the proposed strategies that adaptively distill biased teacher knowledge and allow the model to continuously refine itself.
- **BPL-hard vs. BPL-soft:** Between the hard and soft combinations, the hard selection consistently yields lower errors across three different real-world datasets and achieves the lowest harmonic mean of errors on the factual and counterfactual tests. With the hard selection, the teacher's knowledge for data with low affinity is not used for distillation. This exclusive selection can be more beneficial when $\mathcal{S}^1$ and $\mathcal{S}^0$ have a larger discrepancy. Moreover, the hard selection is more robust to the estimated affinity scores, as it does not use the individual values directly which will also be shown in Section VI-C.

C. *Study of BPL*

*1) **Ablation study**:* We provide a comprehensive ablation study to validate the effectiveness of each proposed component. In Table II, we compare six ablations of BPL-Hard. The first four ablations remove each loss: (1) $\mathcal{L}_{T2}$ (Eq.4), (2) ℓ_{sd} (Eq.5), (3) ℓ_{pd} (Eq.6), and (4) $\mathcal{L}_{T2}$ and $\mathcal{L}_{T3}$. Furthermore, we assess the effectiveness of two alternative design choices: (5) 'w/o confidence penalty' omits the penalty term from ℓ_{pd}, which can be thought of as the typical teacher-student distillation where the student is trained to mimic the

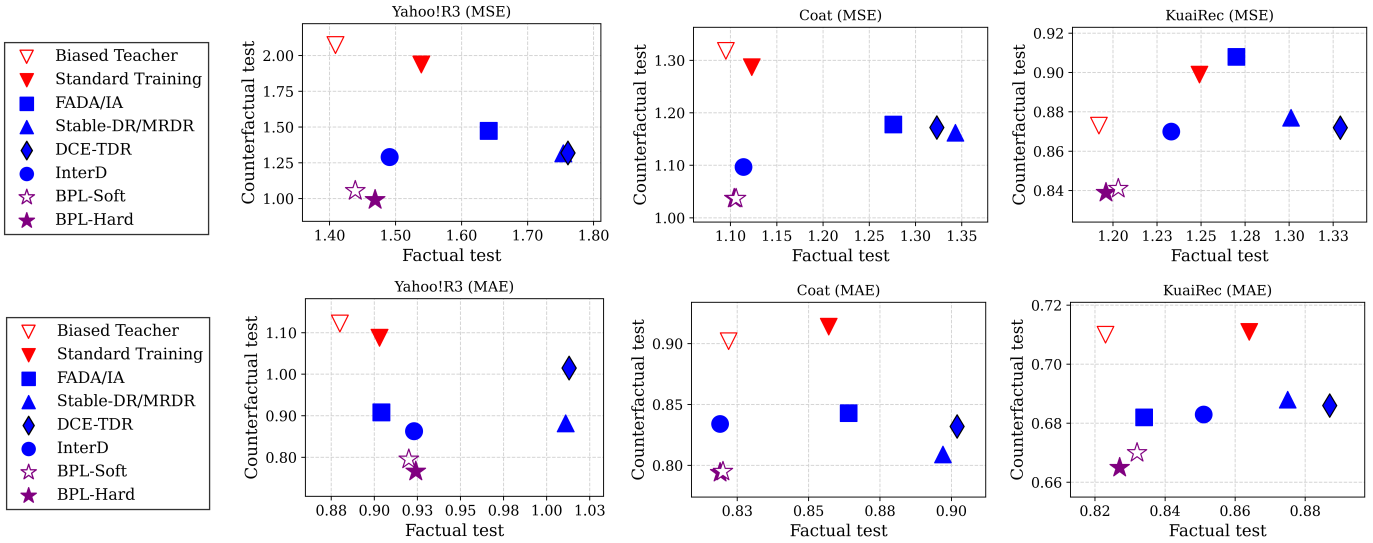

Fig. 6: Factual and counterfactual test results across three real-world datasets. We perform a paired t-test with the best competitor at the 0.05 significance level. BPL-Soft achieves statistically significant improvements in four out of six cases, and BPL-Hard significantly outperforms in five out of six cases on the counterfactual test.

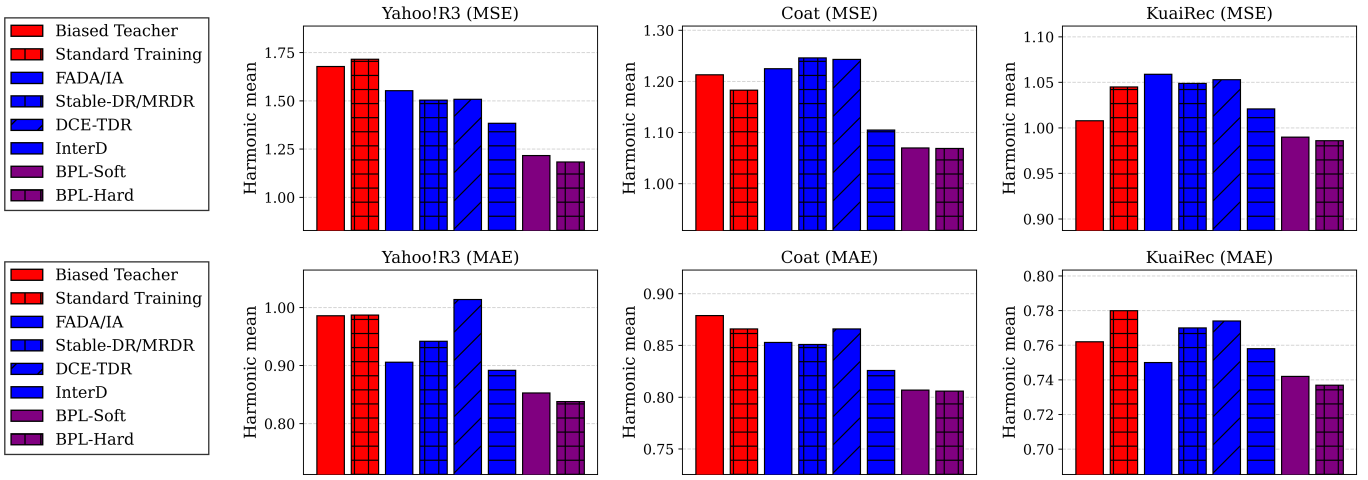

Fig. 7: Harmonic mean of factual and counterfactual test results across three real-world datasets. We perform a paired t-test with the best competitor at the 0.05 significance level. BPL-Soft achieves statistically significant improvements in three out of six cases, and BPL-Hard significantly outperforms in five out of six cases.

teacher's predictions. (6) '$\mathcal{S}^1$-affinity as propensity' reweights the training loss using the estimated affinity for each data, similar to propensity-based approaches [5], [10].

First, the best performance is achieved when using all losses, indicating that each contributes to improved performance on both factual and counterfactual tests. Notably, excluding either of the dual distillation losses (i.e., ℓ_{sd} and ℓ_{pd}) causes a significant performance drop, highlighting that the dual distillation strategies effectively contribute complementary aspect of the preference learning. Second, the confidence penalty is crucial for preference distillation from the biased teacher. Removing the penalty results in a drastic performance drop on counterfactual tests across both datasets, indicating that the model overfits to the biased teacher's predictions without the penalty. This shows the effectiveness of our penalty-combined distillation strategy, which discourage the model from learning patterns specific to high-affinity data. Moreover, reweighting the training loss using affinity scores proves highly ineffective. Lastly, we observe that aligning $\mathcal{S}^0$ and $\mathcal{S}^1$ without considering $\mathcal{S}^{01}$ in Eq.4 slightly degrades the final accuracy (1.043 on Yahoo!R3-RCT) and leads to slower convergence.

*2) **Representation quality analysis**:* We further analyze the quality of representations generated by various preference learning methods. Specifically, we evaluate the rating discriminability of these representations—assessing whether they contain sufficient preference information to distinguish different ratings. As discussed ealier, this measure is closely related to T3. To perform this analysis, we conduct a rating-level classification using *the fixed representations* generated by each method on the counterfactual test set. We employ a two-

TABLE II: Performance of various ablations of BPL.

	Yahoo!R3		Coat	
	Factual	Counter-factual	Factual	Counter-factual
BPL-Hard	**1.469**	**0.991**	**1.104**	**1.037**
Learning objective				
w/o $\mathcal{L}_{T2}$	1.509	1.219	1.119	1.120
w/o ℓ_{sd} ($\mathcal{L}_{T3}$)	1.426	2.035	1.089	1.211
w/o ℓ_{pd} ($\mathcal{L}_{T3}$)	1.494	2.304	1.149	1.242
w/o $\mathcal{L}_{T2}$ & $\mathcal{L}_{T3}$	1.539	1.938	1.123	1.287
Training technique				
w/o confidence penalty	1.454	2.258	1.101	1.371
$\mathcal{S}^1$-affinity as propensity	1.720	1.694	1.294	1.192

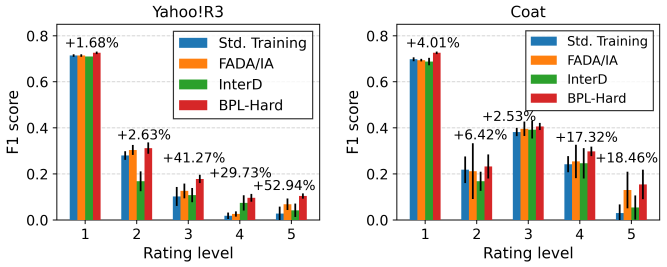

Fig. 8: Evaluation of representation quality. We report the F1 score for each rating level.

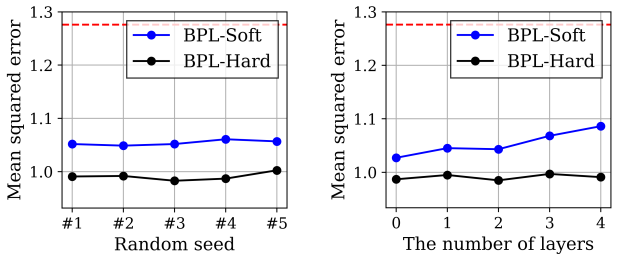

Fig. 9: Results with varying $\mathcal{S}^1$-affinity estimation models. The red dotted line indicates the best competitor, i.e., InterD.

TABLE III: MSE results with varying hyperparameters. H-mean refers to the harmonic mean of the two scores.

		Yahoo!R3				Coat		
		Factual	Counter-factual	H-mean		Factual	Counter-factual	H-mean
α	0.5	1.469	0.991	1.184	0.5	1.104	1.037	1.069
	0.1	1.465	1.016	1.200	0.1	1.109	1.041	1.074
	0.01	1.472	1.020	1.205	0.01	1.098	1.038	1.067
β	2.0	1.471	0.986	1.181	2.0	1.115	1.045	1.079
	1.5	1.469	0.991	1.184	1.5	1.115	1.046	1.079
	1.0	1.478	1.090	1.255	1.0	1.104	1.037	1.069
	0.5	1.484	1.420	1.451	0.5	1.138	1.161	1.149
x	1%	1.469	0.991	1.184	20%	1.108	1.043	1.075
	5%	1.480	1.037	1.220	25%	1.104	1.037	1.069
	10%	1.505	1.051	1.238	30%	1.109	1.044	1.076

layer softmax classifier and perform 5-fold cross-validation.

Figure 8 shows that BPL achieves the highest F1 scores, with particularly significant improvements on high ratings. The counterfactual test is generated through a random exposure process, which typically results in test items with low popularity and a tendency toward lower ratings. This makes high ratings relatively rare and more challenging to predict. This result show that BPL indeed better encodes preference information into the representations, enabling clearer distinctions between different rating levels. Furthermore, BPL outperforms FADA/IA, which also employs adversarial learning strategies. This highlights the superiority of our distillation strategies in capturing underlying preference on unrated data.

*3) **Sensitivity to affinity estimation**:* BPL utilizes $\mathcal{S}^1$-affinity of unrated data obtained by a simple binary classifier (Section IV-B). Here, we investigate how varying affinity estimation models influence BPL. We create five variants by adjusting random seeds, and another five variants by modifying the number of layers. In our experiments, we employ the single-layer model. Figure 9 shows results on the counterfactual test of Yahoo!R3. Both BPL-Soft and BPL-Hard show a considerable degree of robustness and largely outperform the best competitor (i.e., InterD) in all setups. Notably, BPL-Hard shows highly stable performance with all 10 estimation models. BPL-Hard can have such robustness because it uses the estimated affinities only for finding a small set of high-affinity data, without using the individual values.

*4) **Hyperparameter study**:* Table III presents results of BPL-Hard with varying hyperparameters. Overall, BPL shows a relatively stable performance on the factual test. We mainly analyze the results in terms of counterfactual test results.

- For α and β, which control the effects of $\mathcal{L}_{T2}$ and $\mathcal{L}_{T3}$, the best performances are observed around 1 to 2 (α) and around 0.5 (β) on both datasets.
- For x, which controls the size of $\mathcal{S}^{01}$, it shows different tendencies for each dataset. Interestingly, we discover that its optimal value is closely related to the overall divergence between the training and test set. Specifically, Yahoo!R3 shows a higher divergence than Coat [4]; the KL-divergence of the rating distributions are 0.470 (Yahoo! R3) and 0.049 (Coat). As a result, smaller size of $\mathcal{S}^{01}$ is more beneficial for Yahoo!R3, whereas a relatively larger $\mathcal{S}^{01}$ can be used for Coat. This result suggests that x can be selected considering the overall $\mathcal{S}^1$-affinity degree. That is, datasets with higher divergence between observed and unobserved data may benefit from a more conservative (i.e., smaller) selection of $\mathcal{S}^{01}$.

*5) **Results with other backbone models**:* BPL can be flexibly applied to various recommendation models. Figure 10 presents MSE results in the counterfactual test environment using various backbone models, including ID-based, graph neural network (GNN)-based [17], and deep neural network (DNN)-based [15] models, as done in [5], [12], [31]. We compare baselines that show competitive performance in the previous experiments. We observe similar tendencies across the backbone models, which are consistent observations with the previous work [5], [12], [31].

D. Results on benchmark counterfactual (RCT) test

In the previous literature [4], [10], [22], [28], RCT tests of Yahoo!R3 and Coat, have been used as benchmark evaluations. To enable direct comparison under this setup, we report results on the RCT tests as well. Note that the training set used in

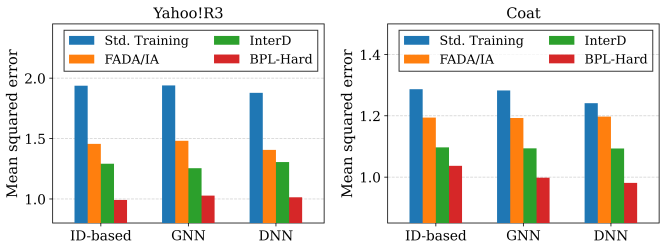

Fig. 10: Results with various backbone models. We perform a paired t-test with the best competitor at the 0.05 significance level. BPL-Hard achieves statistically significant improvements in six out of six cases.

this section is *larger* than that in previous experiments, as we use the **original training set** without splitting it for a factual test set, ensuring consistency with prior research.

Compared methods. Under RCT setup, we compare BPL with various recent methods from related research fields. Further details of baselines are provided in Appendix E.

- **Adversarial & semi-supervised learning**: ADV [42], FADA [53], FADA/IA [54], FreeMatch [47].
- **Debiasing learning for explicit feedback**: IPS [10], AST [56], AT [4], DAMF [28], MR [12], TMRDR-CL [6], Stable-DR [5], Stable-MRDR [5], DCE-TDR [14].
- **Debiasing learning for other behavior types**: TIDE [57], Zerosum [58], ESAM [29], InvCF [31].
- **Knowledge distillation**: InterD [18], DebiasKD, BPL. DebiasKD is a new baseline that combines KD from the biased teacher with the best-performing debiasing method. It uses the biased teacher's predictions for $\mathcal{S}^{01}$ (the same as BPL-Hard) as additional supervision.

From Table IV, we have following observations: First, the basic adversarial learning (ADV) falls short of effectively reducing errors. Also, advanced methods to learn class-discriminative representations (FADA/IA) show limited effectiveness. Similarly, FreeMatch based on semi-supervised learning fails to outperform debiasing methods. These results show the difficulty of capturing user preference from the biased and limited training data. Among the debiasing baselines, methods designed for explicit feedback show competitive performance overall. Also, Stable-DR/MRDR, based on the doubly robust approach, consistently shows superior performance.

KD-based baselines (i.e., InterD, DebiasKD) generally show highly competitive performance. However, DebiasKD performs worse than the best debiasing method, indicating that a naive distillation from the biased teacher can negatively impact preference learning. Conversely, InterD utilizes the prediction interpolation to control the impact of each teacher, outperforming DebiasKD and showing high effectiveness overall. However, as it resorts to the pre-trained teachers, it fails to bring significant improvements compared to the debiased teacher. Lastly, consistent with previous results, BPL generally achieves the lowest errors among all compared methods.

VII. Conclusion

We propose BPL, equipped with new dual distillation strategies, which consist of: (1) reliability-filtered self-distillation, which iteratively refines the model to uncover preferences for unrated data. (2) confidence-penalized preference distillation, which supplements limited feedback using the biased teacher knowledge. These two distillations are adaptively balanced based on the affinity of each potential user-item combination to the collected feedback. Our comprehensive experiments show the effectiveness of BPL in both factual and counterfactual test environments. Future work may explore the applicability of our method to other forms of feedback, such as implicit feedback. In addition, it is important to expand debiasing learning studies to less-studied real-world environments, such as federated learning settings [59]–[61].

TABLE IV: Benchmark counterfactual test (RCT). * denotes significance (paired t-test, $p < 0.05$) against the best baseline.

Method	Yahoo!R3		Coat	
	MSE	MAE	MSE	MAE
Biased Teacher	1.987	1.096	1.306	0.918
Standard Training	1.937	1.086	1.284	0.904
ADV	1.927	1.083	1.254	0.898
FADA	1.499	0.956	1.162	0.838
FADA/IA	1.456	0.898	1.253	0.845
FreeMatch	1.425	0.925	1.172	0.842
IPS	1.712	1.062	1.109	0.873
AST	1.425	0.968	1.178	0.844
AT	1.350	0.945	1.105	0.832
DAMF	1.279	0.926	1.137	0.882
MR	1.349	0.892	1.106	0.786
TMRDR-CL	1.341	0.890	1.104	0.786
Stable-DR	1.327	0.887	1.088	**0.778**
Stable-MRDR	1.316	0.882	1.093	0.779
DCE-TDR	1.319	1.015	1.091	0.832
TIDE	1.547	1.071	1.274	0.917
Zerosum	1.474	0.894	1.228	0.896
ESAM	1.480	0.897	1.129	0.853
InvCF	1.398	1.021	1.194	0.897
InterD	1.278	0.849	1.052	0.814
DebiasKD	1.354	0.895	1.101	0.795
BPL-Soft	1.086*	0.787*	1.006	0.782
BPL-Hard	**0.987***	**0.745***	**0.986***	0.779

Acknowledgments. This work was supported by Microsoft Research Asia and IITP (No.2022-00155958), the NRF funded by the Ministry of Education (NRF-2021R1A6A1A03045425), the IITP grants funded by the MSIT (IITP-2025-RS-2020-II201819, RS-2024-00457882), and 2025 High-Performance Computing Support Program.

References

[1] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–39, 2023.
[2] Q. Zhang, L. Cao, C. Shi, and L. Hu, "Tripartite collaborative filtering with observability and selection for debiasing rating estimation on missing-not-at-random data," in *AAAI*, 2021, pp. 4671–4678.
[3] Y. Liu, X. Cao, and Y. Yu, "Are you influenced by others when rating? improve rating prediction by conformity modeling," in *RecSys*, 2016, pp. 269–272.
[4] Y. Saito, "Asymmetric tri-training for debiasing missing-not-at-random explicit feedback," in *SIGIR*, 2020, pp. 309–318.
[5] H. Li, C. Zheng, and P. Wu, "Stabledr: Stabilized doubly robust learning for recommendation on data missing not at random," in *ICLR*, 2023.
[6] H. Li, Y. Lyu, C. Zheng, and P. Wu, "Tdr-cl: Targeted doubly robust collaborative learning for debiased recommendations," in *ICLR*, 2023.
[7] B. Liu, Q. Luo, and B. Wang, "Debiased pairwise learning for implicit collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[8] X. Zhu, Y. Zhang, X. Yang, D. Wang, and F. Feng, "Mitigating hidden confounding effects for causal recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[9] Z. Xu, P. Wei, W. Zhang, S. Liu, L. Wang, and B. Zheng, "Ukd: Debiasing conversion rate estimation via uncertainty-regularized knowledge distillation," in *WWW*, 2022, pp. 2078–2087.

[10] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation," in *ICML*, 2016, pp. 1670–1679.

[11] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Doubly robust joint learning for recommendation on data missing not at random," in *ICML*, 2019, pp. 6638–6647.

[12] H. Li, Q. Dai, Y. Li, Y. Lyu, Z. Dong, P. Wu, and X.-H. Zhou, "Multiple robust learning for recommendation," in *AAAI*, 2023, pp. 4417–4425.

[13] Z. Song, J. Chen, S. Zhou, Q. Shi, Y. Feng, C. Chen, and C. Wang, "Cdr: Conservative doubly robust learning for debiased recommendation," in *CIKM*, 2023, pp. 2321–2330.

[14] W. Kweon and H. Yu, "Doubly calibrated estimator for recommendation on data missing not at random," in *WWW*, 2024, pp. 3810–3820.

[15] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.

[16] S. C. Han, T. Lim, S. Long, B. Burgstaller, and J. Poon, "Glocal-k: Global and local kernels for recommender systems," in *CIKM*, 2021, pp. 3063–3067.

[17] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[18] S. Ding, F. Feng, X. He, J. Jin, W. Wang, Y. Liao, and Y. Zhang, "Interpolative distillation for unifying biased and debiased recommendation," in *SIGIR*, 2022, pp. 40–49.

[19] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, pp. 151–175, 2010.

[20] Z. Allen-Zhu and Y. Li, "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning," in *ICLR*, 2023.

[21] G. Hinton, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[22] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Combating selection biases in recommender systems with a few unbiased ratings," in *WSDM*, 2021, pp. 427–435.

[23] J. Chen, H. Dong, Y. Qiu, X. He, X. Xin, L. Chen, G. Lin, and K. Yang, "Autodebias: Learning to debias for recommendation," in *SIGIR*, 2021, pp. 21–30.

[24] H. Li, Y. Xiao, C. Zheng, and P. Wu, "Balancing unobserved confounding with a few unbiased ratings in debiased recommendations," in *WWW*, 2023, pp. 1305–1313.

[25] H. Li, Y. Xiao, C. Zheng, P. Wu, and P. Cui, "Propensity matters: Measuring and enhancing balancing for recommendation," in *ICML*. PMLR, 2023, pp. 20 182–20 194.

[26] A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims, "A general framework for counterfactual learning-to-rank," in *SIGIR*, 2019.

[27] W. Kweon and H. Yu, "Doubly calibrated estimator for recommendation on data missing not at random," in *WWW*, 2024, pp. 3810–3820.

[28] Y. Saito and M. Nomura, "Towards resolving propensity contradiction in offline recommender learning," in *IJCAI*, 2022.

[29] Z. Chen, R. Xiao, C. Li, G. Ye, H. Sun, and H. Deng, "Esam: Discriminative domain adaptation with non-displayed items to improve long-tail performance," in *SIGIR*, 2020, pp. 579–588.

[30] H. Pan, J. Chen, F. Feng, W. Shi, J. Wu, and X. He, "Discriminative-invariant representation learning for unbiased recommendation," in *IJCAI*, 2023, pp. 2270–2278.

[31] A. Zhang, J. Zheng, X. Wang, Y. Yuan, and T.-S. ChuI, "Invariant collaborative filtering to popularity distribution shift," in *WWW*, 2023, pp. 1240–1251.

[32] H. Zhang, S. Wang, H. Li, C. Zheng, X. Chen, L. Liu, S. Luo, and P. Wu, "Uncovering the propensity identification problem in debiased recommendations," in *ICDE*. IEEE, 2024, pp. 653–666.

[33] H. Li, C. Zheng, W. Wang, H. Wang, F. Feng, and X.-H. Zhou, "Debiased recommendation with noisy feedback," in *KDD*, 2024, pp. 1576–1586.

[34] P. Li, X. Tong, Y. Wang, and Q. Zhang, "Meta doubly robust: Debiasing cvr prediction via meta-learning with a small amount of unbiased data," *Knowledge-Based Systems*, vol. 310, p. 112898, 2025.

[35] S. Kang, W. Kweon, D. Lee, J. Lian, X. Xie, and H. Yu, "Distillation from heterogeneous models for top-k recommendation," in *WWW*, 2023, pp. 801–811.

[36] G. Chen, J. Chen, F. Feng, S. Zhou, and X. He, "Unbiased knowledge distillation for recommendation," in *WSDM*, 2023, pp. 976–984.

[37] S. Kang, J. Hwang, W. Kweon, and H. Yu, "De-rrd: A knowledge distillation framework for recommender system," in *CIKM*, 2020, pp. 605–614.

[38] D. Liu, P. Cheng, Z. Lin, J. Luo, Z. Dong, X. He, W. Pan, and Z. Ming, "Kdcrec: Knowledge distillation for counterfactual recommendation via uniform data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8143–8156, 2022.

[39] D. Liu, P. Cheng, Z. Dong, X. He, W. Pan, and Z. Ming, "A general knowledge distillation framework for counterfactual recommendation via uniform data," in *SIGIR*, 2020, pp. 831–840.

[40] J. M. Hernández-Lobato, N. Houlsby, and Z. Ghahramani, "Probabilistic matrix factorization with non-random missing data," in *ICML*, 2014, pp. 1512–1520.

[41] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation," in *WWW*, 2016, pp. 951–961.

[42] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[43] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *ICML*, 2019, pp. 1081–1090.

[44] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[45] X. Jin, C. Lan, W. Zeng, and Z. Chen, "Re-energizing domain discriminator with sample relabeling for adversarial domain adaptation," in *ICCV*, 2021, pp. 9174–9183.

[46] L. Wu, H. Lin, Z. Gao, G. Zhao, and S. Z. Li, "A teacher-free graph knowledge distillation framework with dual self-distillation," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[47] Y. Wang, H. Chen, Q. Heng, W. Hou, M. Savvides, T. Shinozaki, B. Raj, Z. Wu, and J. Wang, "Freematch: Self-adaptive thresholding for semi-supervised learning," in *ICLR*, 2023.

[48] T. Zhou, S. Wang, and J. Bilmes, "Time-consistent self-supervision for semi-supervised learning," in *ICML*, 2020, pp. 11 523–11 533.

[49] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," *NeurIPS*, 2004.

[50] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv preprint arXiv:1701.06548*, 2017.

[51] B. Marlin, R. S. Zemel, S. Roweis, and M. Slaney, "Collaborative filtering and the missing at random assumption," *arXiv preprint arXiv:1206.5267*, 2012.

[52] C. Gao, S. Li, W. Lei, J. Chen, B. Li, P. Jiang, X. He, J. Mao, and T.-S. Chua, "Kuairec: A fully-observed dataset and insights for evaluating recommender systems," in *CIKM*, 2022, pp. 540–550.

[53] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *ECCV*, 2020, pp. 642–659.

[54] Y. Liu, J. Deng, J. Tao, T. Chu, L. Duan, and W. Li, "Undoing the damage of label shift for cross-domain semantic segmentation," in *CVPR*, 2022, pp. 7042–7052.

[55] S. Guo, L. Zou, Y. Liu, W. Ye, S. Cheng, S. Wang, H. Chen, D. Yin, and Y. Chang, "Enhanced doubly robust learning for debiasing post-click conversion rate estimation," in *SIGIR*, 2021, pp. 275–284.

[56] T. Xiao, Z. Chen, and S. Wang, "Reconsidering learning objectives in unbiased recommendation: A distribution shift perspective," in *KDD*, 2023, pp. 2764–2775.

[57] Z. Zhao, J. Chen, S. Zhou, X. He, X. Cao, F. Zhang, and W. Wu, "Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 9920–9931, 2022.

[58] W. Rhee, S. M. Cho, and B. Suh, "Countering popularity bias by regularizing score differences," in *RecSys*, 2022, pp. 145–155.

[59] Y. Wang, H. Xu, W. Ali, X. Zhou, and J. Shao, "Bilateral improvement in local personalization and global generalization in federated learning," *IEEE Internet of Things Journal*, 2024.

[60] W. Ali, K. Umer, X. Zhou, and J. Shao, "Hidattack: An effective and undetectable model poisoning attack to federated recommenders," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[61] W. Ali, X. Zhou, and J. Shao, "Privacy-preserved and responsible recommenders: From conventional defense to federated learning and blockchain," *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–35, 2025.