
Unbiased Recommender Learning from Implicit Feedback via Weakly Supervised Learning

Hao Wang¹ Zhichao Chen¹ Haotian Wang² Yanchao Tan¹ Licheng Pan¹ Tianqiao Liu³ Xu Chen⁴
Haoxuan Li⁵ Zhouchen Lin^{6,7,8}

Abstract

Implicit feedback recommendation is challenged by the missing negative feedback essential for effective model training. Existing methods often resort to negative sampling, a technique that assumes unlabeled interactions as negative samples. This assumption risks misclassifying potential positive samples within the unlabeled data, thereby undermining model performance. To address this issue, we introduce PURL, a model-agnostic framework that reframes implicit feedback recommendation as a weakly supervised learning task, eliminating the need for negative samples. However, its unbiasedness hinges on the accurate estimation of the class prior. To address this challenge, we propose Progressive Proximal Transport (PPT), which estimates the class prior by minimizing the proximal transport cost between positive and unlabeled samples. Experiments on three real-world datasets validate the efficacy of PURL in terms of improved recommendation quality. Code is available at <https://anonymous.4open.science/r/PURL-1AF3/>.

1. Introduction

Recommendation systems aim to capture user preferences through their feedback, subsequently offering personalized content or product suggestions. They find extensive applications across various domains, including e-commerce (Smith

and Linden, 2017; Wang et al., 2022), advertising (Zhou et al., 2018), and entertainment (Gomez-Urbe and Hunt, 2016). By accurately predicting user preferences, recommendation systems play a pivotal role in enhancing user experience and increasing business revenues (Gao et al., 2022a;b).

The core of recommendation system is the handling of user feedback, which can be broadly categorized into explicit and implicit types. Explicit feedback, such as ratings, comments, or like/dislike tags, reflects user preferences faithfully but is challenging to collect, as it requires active user participation, potentially diminishing user experience. Conversely, implicit feedback, such as page views or click logs, is readily available and does not require active user participation. Therefore, recommendation with implicit feedback, termed as ImplicitRec, has gained significant attention in industrial applications (Togashi et al., 2021; Lee et al., 2022).

A unique challenge in ImplicitRec is missing negative feedback (MNF) (Lim et al., 2015; Yang et al., 2018). Specifically, the available data only consists of positive and unlabeled samples. This absence of negative feedback complicates the task of differentiating between items that are either disliked or simply unexposed to the user across the unlabeled interactions (Zhou et al., 2021; Ren et al., 2023; Wang et al., 2021). For instance, a user’s failure to click on a video might not necessarily indicate disinterest; it could simply mean the user was not exposed to that particular video. Treating unlabeled interactions as negative instances (Zhou et al., 2022; Kang and McAuley, 2018) ignores the existence of positive samples within the unlabeled data, thereby introducing bias and compromising recommendation performance (Wang et al., 2021).

To counteract the MNF issue, one line of works advocate downweighting unobserved samples (Hu et al., 2008; Liang et al., 2016), employing pairwise ranking loss (Rendle et al., 2009). While promising, these methods are largely heuristic and lack a solid theoretical foundation for unbiasedness (??). Recent works have shifted focus towards the construction of theoretically unbiased estimators of the ideal risk, using only positive and unlabeled samples. Representative methods such as RMF (Saito et al., 2020), UBPR (Saito, 2020)

¹Zhejiang University, ²National University of Defense Technology ³TAL Education Group ⁴Renmin University of China ⁵Center for Data Science, Peking University ⁶State Key Lab of General AI, School of Intelligence Science and Technology, Peking University ⁷Institute for Artificial Intelligence, Peking University ⁸Pazhou Laboratory (Huangpu), Guangzhou, China. Correspondence to: Haoxuan Li <hxli@stu.pku.edu.cn>, Zhouchen Lin <zlin@pku.edu.cn>.

and CJMF (Zhu et al., 2020) primarily leverage propensity re-weighting, a methodology rooted in causal inference (Li et al., 2023b), which offers unbiased estimates provided accurate propensity score estimation (Li et al., 2023c;a). However, identifying accurate propensity scores remains an elusive goal in ImplicitRec¹. Furthermore, the derived risk estimators do not necessarily maintain non-negativity (Zhu et al., 2020; Saito, 2020; Saito et al., 2020), posing the risk of non-convergence. Consequently, the MNF issue remains an open challenge in ImplicitRec and warrants further investigation.

This work introduces Positive-Unlabeled Recommender Learning (PURL), a model-agnostic framework based on learning from weak supervision paradigm (Sugiyama et al., 2022), to develop rigorous formulations and practical solutions for handling MNF without reliance on vulnerable propensity scores. Specifically, we first reformulate the ideal risk and construct an empirical estimator that obviates the need for negative feedback. We show that this estimator is unbiased, consistent with the ideal risk, and amenable to optimization via stochastic gradient descent, with unique attainable optimum for representative recommender families. However, the estimator’s performance is contingent upon a predefined class prior, which requires expert knowledge for empirical selection. To address this, we develop a Progressive Proximal Transport (PPT) method which achieves automatic class prior estimation by minimizing the proximal transport cost between the positive and unlabeled samples. Extensive evaluations on three publicly available real-world datasets substantiate the efficacy of the proposed method.

The main contributions of this paper are summarized below:

- We propose PURL, a model-agnostic framework addressing the missing-not-at-random (MNF) issue from a weakly supervised learning perspective. At its core, PURL features a risk estimator that eliminates the need for negative samples and is theoretically unbiased and consistent with the ideal risk, given accurate class prior estimation.
- We introduce PPT, an optimal transport-based methodology specifically designed for class prior estimation, a key factor in ensuring the unbiasedness of PURL.
- We empirically validate the efficacy of the proposed PURL approach through comprehensive evaluations on three real-world datasets.

¹The term propensity means the likelihood that the sample would be observed. It is quite difficult to be estimated in the realm of ImplicitRec due to MNF, since the samples that are observed but disliked is unknown.

2. Preliminary

In this section, we establish technical preliminaries essential for comprehending the proposed methods. Initially, we delineate the fundamental symbols and formalize the problem definition of ImplicitRec. Subsequently, we offer a concise overview of optimal transport, a mathematical framework that is intrinsically linked to the proposed PPT method for class prior estimation.

2.1. Implicit Feedback Recommendation

Let \mathcal{U} and \mathcal{I} denote the sets of users and items, respectively, and $\mathcal{X} = \mathcal{U} \times \mathcal{I}$ represent the set of all possible user-item interactions, where each sample is denoted by $x = (u, i)$. We introduce Y as the binary indicator variable for positive feedback and designate p_{data} as the underlying distribution governing all interactions. The objective is to construct a recommendation model, symbolized as g , that is proficient in discriminating between samples associated with positive and negative feedback. To quantify the performance of the model, we employ an error measure $\ell(\hat{y}, y)$, where y signifies the ground truth feedback and \hat{y} represents the model’s estimated feedback. In line with existing literature (Saito et al., 2020; Luo et al., 2021), the ideal risk function is formulated as:

$$R_{\text{ideal}}(g) = \mathbb{E}_{p_{\text{data}}(x,y)} [\ell(g(x), y)]. \quad (1)$$

The samples with negative feedback, i.e., $p_{\text{data}}(x|y = -1)$, is unavailable in ImplicitRec due to the MNF issue, rendering the ideal risk in (1) incalculable. Current approaches typically assume that unobserved samples are all negative, resorting to negative sampling on these samples to generate negative feedback. However, this assumption does not hold in real-world practice as the negatively sampled data inevitably contain a proportion of positive feedback. Assigning negative labels to them can lead to an unstable training process and compromise overall performance (Rendle et al., 2009; Saito, 2020).

2.2. Optimal Transport

Optimal transport (OT) is a mathematical framework designed to measure the discrepancy between two distributions by calculating the minimum transport cost, which has been used in many fundamental fields due to its empirical flexibility and theoretical rigor: computer vision (Courty et al., 2017), text analysis (Xu et al., 2021), and domain adaptation (Shen et al., 2018). Initially, Monge (1781) first formulated it as finding an optimal mapping between two continuous distributions. However, this formulation had limitations in terms of the existence and uniqueness of solutions. A more practical formulation was later proposed by Kantorovich (2006), with an entropic regularizer for enabling fast computation via Sinkhorn algorithm (Altschuler

et al., 2017). We provide its specification for comparing two empirical distributions in Definition 1.

Definition 1. For empirical distributions $\alpha_{1:n}$ and $\beta_{1:m}$ with n and m samples, respectively, the Kantorovich problem aims to find an optimal plan $\pi \in \mathbb{R}_+^{n \times m}$ that minimizes the transport cost between α and β . Formally, the problem is defined as:

$$\begin{aligned} \mathbb{W}(\alpha, \beta) &:= \min_{\pi \in \Pi(\alpha, \beta)} \langle \mathbf{C}, \pi \rangle, \\ \Pi(\alpha, \beta) &:= \{ \pi \in \mathbb{R}_+^{n \times m} : \pi \mathbf{1}_m = \mathbf{a}, \pi^\top \mathbf{1}_n = \mathbf{b} \}, \end{aligned} \quad (2)$$

where $\mathbb{W}(\alpha, \beta) \in \mathbb{R}$ is the transport cost; $\mathbf{C} \in \mathbb{R}_+^{n \times m}$ denotes the sample-wise distance between α and β . $\mathbf{1}_n$ and $\mathbf{1}_m$ are column vectors filled with ones. The set Π defines feasible transport plans. \mathbf{a} and \mathbf{b} specify the mass of units in α and β .

Research in optimal transport primarily follows two trajectories. The first aims to reduce the computational complexity of solving OT problems. While exact solutions can be obtained through linear programming algorithms, these come with cubic complexity in relation to the number of samples (Bonneel et al., 2011). To address this, various approximate algorithms for acceleration have been developed, such as the Sinkhorn and sliced OT algorithm (Altschuler et al., 2017) with quadratic and linear complexity, respectively. The second line of research focuses on modifying the transport problem to suit specific applications. Examples include the weak transport problem in domain adaptation (Chizat et al., 2018), the Schrödinger bridge problem in generative modeling (Marino and Gerolin, 2020), and the Gromov problem in graph matching (Xu et al., 2019).

3. Methodology

This section presents our proposed PURL framework. We first construct a surrogate estimator of the ideal risk (1) utilizing only positive and unlabeled samples. A thorough theoretical analysis is provided to expound upon its advantageous attributes. We then formulate the Progressive Proximal Transport (PPT) method to achieve automatic estimation of the class prior. The overall workflow of PURL is finally encapsulated for implementation.

3.1. Positive-Unlabeled Recommender Learning

Assuming that positive and negative samples arise from marginal distributions $p_p(x) = p_{\text{data}}(x|y = 1)$ and $p_n(x) = p_{\text{data}}(x|y = -1)$, respectively, the data distribution can be expressed as:

$$p_{\text{data}}(x) = \kappa_p p_p(x) + \kappa_n p_n(x), \quad (3)$$

where $\kappa_p = p(y = +1)$ and $\kappa_n = p(y = -1)$ are the class priors, satisfying $\kappa_p + \kappa_n = 1$. Decomposing the ideal risk

in (1) yields:

$$\begin{aligned} R_{\text{ideal}}(g) &= \kappa_p \mathbb{E}_{p_p}[\ell(g(x), +1)] + \kappa_n \mathbb{E}_{p_n}[\ell(g(x), -1)] \\ &:= \kappa_p R_p(g) + \kappa_n R_n(g), \end{aligned}$$

where $R_p(g)$ and $R_n(g)$ denote the expected error on positive and negative samples, respectively. Since negative samples are inaccessible due to MNF, a workaround involves leveraging the unlabeled data distribution, which is a mixture of positive and negative data distributions:

$$p_u(x) = \kappa_p p_p(x) + \kappa_n p_n(x).$$

On the basis of this, the negative risk $R_n(g)$ can be expressed in terms of the unlabeled and positive samples:

$$\begin{aligned} \kappa_n R_n(g) &= \kappa_n \mathbb{E}_{p_n}[\ell(g(x), -1)] \\ &= \kappa_n \int \ell(g(x), -1) p_n(x) dx \\ &= \int \ell(g(x), -1) p_u(x) dx \\ &\quad - \kappa_p \int \ell(g(x), -1) p_p(x) dx \\ &= \mathbb{E}_{p_u}[\ell(g(x), -1)] - \kappa_p \mathbb{E}_{p_p}[\ell(g(x), -1)]. \end{aligned} \quad (4)$$

Introducing $\tilde{\ell}(g(x)) = \ell(g(x), +1) - \ell(g(x), -1)$, we can eliminate the expectation over the negative samples in (1) to derive a surrogate of the ideal risk:

$$R_{\text{purl}}(g) = \kappa_p \mathbb{E}_{p_p}[\tilde{\ell}(g(x))] + \mathbb{E}_{p_u}[\ell(g(x), -1)], \quad (5)$$

where subtracting the risk of misclassifying positive samples as negative compensates for the bias introduced by treating all unlabeled samples as negative. Consequently, the PURL risk excludes unavailable negative feedback, enabling empirical estimation using positive and unobserved samples:

$$\hat{R}_{\text{purl}}(g) = \frac{\kappa_p}{n_p} \sum_{i=1}^{n_p} \tilde{\ell}(g(x_i^p)) + \frac{1}{n_u} \sum_{i=1}^{n_u} \ell(g(x_i^u), -1). \quad (6)$$

Theoretical Justification. We demonstrate that the *PURL estimator is unbiased relative to the ideal risk* enabling training and evaluation without reliance on negative samples (Theorem 1). Moreover, *adding more unlabeled samples consistently reduces estimation error* (Theorem 3), suggesting an opportunity to improve ImplicitRec performance. Thanks to the convexity of $\hat{R}_{\text{purl}}(g)$, *the convergence of PURL is guaranteed with bounded errors in mild conditions* (Lemma 3). Detailed proofs are provided in Appendix B.

3.2. Progressive Proximal Transport

The estimator in (6) necessitates a predefined positive class prior κ_p , a.k.a., the proportion of positive samples concealed within unlabeled data. Although canonical positive-unlabeled learning scenarios often assume κ_p to be known,

such an assumption is untenable in ImplicitRec where the class prior is costly and difficult to obtain, thereby necessitating class prior estimation. However, the task is difficult in ImplicitRec due to MNF. Existing methods are mainly based on partial alignment (du Plessis et al., 2015a; 2017) which minimizes the f-divergence between labeled and unlabeled distributions. However, these approaches suffer from an inherent positive bias (Sugiyama et al., 2022) and are ill suited for data based on sparse ids.

Definition 2 (Proximal Transport, PT). Suppose $\alpha_{1:n}$ and $\beta_{1:m}$ are empirical distributions corresponding to samples with positive and unlabeled feedback, respectively. Given a mass weight $w \in \mathbb{R}_+$, we aim to find a transport plan $\pi^* \in \mathbb{R}^{n \times m}$ minimizing the transport cost between α and β . Formally, the problem is defined as:

$$\begin{aligned} \pi^*(\alpha, \beta; w) &:= \arg \min_{\pi \in \Pi(\alpha, \beta; w)} \langle \mathbf{C}, \pi \rangle, \\ \Pi(\alpha, \beta; w) &:= \{ \pi \in \mathbb{R}_+^{n \times m} : \pi \mathbf{1}_m = \mathbf{a}, \pi^\top \mathbf{1}_n \leq w * \mathbf{b}, \\ &\quad \mathbf{1}_n^\top \pi \mathbf{1}_m = 1 \}, \end{aligned} \quad (7)$$

where $\langle \mathbf{C}, \pi \rangle$ represents the transport cost; $\mathbf{C} \in \mathbb{R}_+^{n \times m}$ denotes the sample-wise distance² between α and β . $\mathbf{1}_n$ and $\mathbf{1}_m$ are column vectors filled with ones. The set Π defines the feasible transport plans. \mathbf{a} and \mathbf{b} specify the mass of units in α and β , respectively, and are implemented as $\mathbf{1}_n/n$ and $\mathbf{1}_m/m$ under the assumption of a uniform mass distribution.

To address this gap, we introduce the Progressive Proximal Transport (PPT) method, specifically designed for class prior estimation in the context of our PURL framework. The core concept of PPT is the proximal transport (PT) problem, as outlined in Definition 2. Unlike the traditional Kantorovich problem, our approach adjusts the mass of β by a weight factor $w \geq 1$ and modifies the mass preservation equality to allow for non-matched samples in β . Figure 1 illustrates how the PT transport strategy evolves with different values of w . When $w = 1$, PT matches all samples in β , effectively reducing to the standard Kantorovich problem. As w increases, PT allows each sample in β to be matched with multiple samples in α . As a result, the positively labeled samples α will be matched towards the nearest sub-distribution within β , driven by minimizing the transport cost in (7).

The proposed PPT approach leverages this property of PT to estimate the class prior κ_p by progressively adjusting the weight w . Starting with a large initial value of w , the matched samples in β are primarily positive, as they are expected to be closely aligned with α . As w decreases, more samples in β are considered for matching, without significantly increasing the transport cost due to the substantial overlap between the positive samples labeled and unlabeled.

²Here, we calculate the sample-wise distance with the squared Euclidean metric in line with Altschuler et al. (2017).

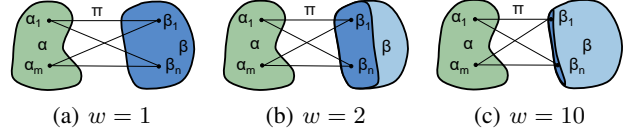


Figure 1: Overview of the PT transport plan under varying mass weight parameters w . The dark fields in β demarcate the domain containing samples that are transported. Lines connecting α and β signify the transport plans $\pi(\alpha, \beta; w)$.

Algorithm 1 The computation workflow of PURL.

Input: a mini-batch of samples \mathbf{x} , recommender model g with user and item embedding look-up table \mathbf{U} and \mathbf{V} .

Parameter: η : learning rate; $w_{1:K}$: candidate mass weights.

Output: updated embedding look-up tables \mathbf{U}^\dagger and \mathbf{V}^\dagger .

- 1: $x_{1:n_p}^p, x_{1:n_u}^u \leftarrow \text{split}(\mathbf{x})$.
 - 2: $\mathbf{u}_i^p, \mathbf{v}_i^p \leftarrow \text{look-up}(\mathbf{U}, \mathbf{V}, x_i^p), \mathbf{z}_i^p \leftarrow \text{concat}(\mathbf{u}_i^p, \mathbf{v}_i^p), i = 1 : n_p$.
 - 3: $\mathbf{u}_j^u, \mathbf{v}_j^u \leftarrow \text{look-up}(\mathbf{U}, \mathbf{V}, x_j^u), \mathbf{z}_j^u \leftarrow \text{concat}(\mathbf{u}_j^u, \mathbf{v}_j^u), j = 1 : n_u$.
 - 4: $c_{ij} \leftarrow \|\mathbf{z}_i^p - \mathbf{z}_j^u\|_2^2, \mathbf{C} \leftarrow [c_{ij}]_{i,j=1}^{n_p, n_u}$.
 - 5: $\pi_k^* \leftarrow \arg \min_{\pi \in \Pi(\mathbf{z}_{1:n_p}^p, \mathbf{z}_{1:n_u}^u; w_k)} \langle \mathbf{C}, \pi \rangle, k = 1 : K$.
 - 6: $\mathbb{W}_k \leftarrow \langle \mathbf{C}, \pi_k^* \rangle + \mathbf{KL}(\pi_k^{\top} \mathbf{1}_{n_p}, \mathbf{1}_{n_u}), k = 1 : K$.
 - 7: $k^* \leftarrow \arg \min \mathbb{W}_{k=1:K}, \hat{\kappa} \leftarrow 1/w_{k^*}$.
 - 8: $\hat{R}_{\text{purl}} \leftarrow \frac{\hat{\kappa}_p}{n_p} \sum_{i=1}^{n_p} \tilde{\ell}(g(\mathbf{u}_i^p, \mathbf{v}_i^p))_+ + \frac{1}{n_u} \sum_{j=1}^{n_u} \ell(g(\mathbf{u}_j^u, \mathbf{v}_j^u), -1)$.
 - 9: $\mathbf{U}^\dagger \leftarrow \mathbf{U} + \eta \cdot \nabla_{\mathbf{U}} \hat{R}_{\text{purl}}, \mathbf{V}^\dagger \leftarrow \mathbf{V} + \eta \cdot \nabla_{\mathbf{V}} \hat{R}_{\text{purl}}$.
-

Upon reaching a threshold w^* , the inclusion of negative samples becomes unavoidable, leading to a sharp increase in the transport cost, since the distributions of positive and negative samples are heterogeneous. The class prior is then estimated as $\hat{\kappa}_p = 1/w^*$, justified by the fact that only approximately $1/w^*$ of the samples in β are matched by PT as illustrated in Figure 1.

The PT developed in Definition 2 is inherently a linear programming problem and thus can be solved with well-established convex optimizers. Interestingly, PT can be viewed as an augmented Kantorovich problem, which enables the utilization of specialized optimal transport solvers, known for superior convergence rates and computational efficiency. A comprehensive analysis of this reformulation is provided in Appendix B.2.

3.3. Implementation of PURL

In this section, we delineate the implementation of the PURL framework, employing matrix factorization as an exemplar recommender model. The computational workflow for a

Table 1: Recommendation performances in terms of NDCG@k on Yahoo! R3, Coat and KuaiRec.

Dataset	Yahoo! R3			Coat			KuaiRec		
Metrics	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3	NDCG@5	NDCG@1	NDCG@3	NDCG@5
RMF	0.632 \pm 0.003	0.666 \pm 0.003	0.706 \pm 0.003	0.374 \pm 0.026	0.403 \pm 0.020	0.442 \pm 0.021	0.211 \pm 0.007	0.209 \pm 0.005	0.212 \pm 0.003
CRMF	0.633 \pm 0.004	0.667 \pm 0.003	0.706 \pm 0.003	0.377 \pm 0.022	0.401 \pm 0.018	0.445 \pm 0.023	0.212 \pm 0.005	0.218 \pm 0.004	0.222 \pm 0.002
WMF	0.707 \pm 0.003	0.737 \pm 0.003	0.772 \pm 0.002	0.389 \pm 0.025	0.425 \pm 0.023	0.466 \pm 0.027	0.226 \pm 0.009	0.244 \pm 0.008	0.261 \pm 0.008
UBPR	0.755 \pm 0.003	0.794 \pm 0.001	0.826 \pm 0.001	0.476 \pm 0.024	0.509 \pm 0.014	0.541 \pm 0.007	0.248 \pm 0.009	0.265 \pm 0.006	0.281 \pm 0.006
CUBPR	0.755 \pm 0.003	0.794 \pm 0.002	0.826 \pm 0.002	0.478 \pm 0.026	0.509 \pm 0.013	0.542 \pm 0.007	0.254 \pm 0.007	0.268 \pm 0.007	0.284 \pm 0.007
UPL	0.759 \pm 0.003	0.798 \pm 0.002	0.829 \pm 0.003	0.512 \pm 0.019	0.508 \pm 0.013	0.539 \pm 0.012	0.250 \pm 0.006	0.265 \pm 0.006	0.281 \pm 0.005
RecVAE	0.650 \pm 0.002	0.683 \pm 0.004	0.719 \pm 0.005	0.465 \pm 0.021	0.499 \pm 0.017	0.545 \pm 0.009	0.422 \pm 0.006	0.434 \pm 0.005	0.448 \pm 0.007
DCF	0.762 \pm 0.003	0.796 \pm 0.003	0.826 \pm 0.002	0.544 \pm 0.0025	0.538 \pm 0.019	0.566 \pm 0.007	0.448 \pm 0.007	0.457 \pm 0.008	0.466 \pm 0.004
ORACLE	0.652 \pm 0.004	0.692 \pm 0.003	0.732 \pm 0.003	0.443 \pm 0.019	0.470 \pm 0.012	0.507 \pm 0.011	0.204 \pm 0.004	0.225 \pm 0.005	0.246 \pm 0.005
CDR	0.669 \pm 0.005	0.695 \pm 0.003	0.727 \pm 0.004	0.398 \pm 0.015	0.420 \pm 0.014	0.457 \pm 0.014	0.491 \pm 0.012	0.425 \pm 0.005	0.412 \pm 0.004
SDR	0.666 \pm 0.001	0.688 \pm 0.001	0.719 \pm 0.001	0.452 \pm 0.030	0.467 \pm 0.022	0.502 \pm 0.018	0.493 \pm 0.006	0.419 \pm 0.010	0.400 \pm 0.010
UIDR	0.679 \pm 0.030	0.705 \pm 0.032	0.737 \pm 0.033	0.382 \pm 0.015	0.400 \pm 0.011	0.436 \pm 0.014	0.493 \pm 0.085	0.446 \pm 0.060	0.435 \pm 0.050
PURL	0.784* \pm 0.003	0.814* \pm 0.003	0.843* \pm 0.002	0.552* \pm 0.031	0.555* \pm 0.019	0.589* \pm 0.015	0.498 \pm 0.008	0.486 \pm 0.007	0.488* \pm 0.004

Note: The results are reported in mean \pm std with 5 runs. The best and second best metrics are bolded and underlined, respectively. "*" marks the metrics that PURL surpasses the best baseline with p-value < 0.05 over paired samples t-test.

round of parameter updating is outlined in Algorithm 1.

Initially, we sample a mini-batch of samples, denoted as \mathbf{x} . This batch is subsequently partitioned into two subsets: positively-labeled samples $x_{1:n_p}^p$ and unlabeled samples $x_{1:n_u}^u$ (step 1). Following this partitioning, we look up the embedding tables to generate user and item embeddings (steps 2-3). The sample-wise distances between these embeddings are then computed (step 4). For each candidate mass weight w_k , we calculate the transport strategy by solving the PT problem in (7) and calculate the associated transport cost (steps 5-6). Notably, the cost is augmented with a KL divergence term to facilitate comprehensive matching of the positive samples within the unlabeled set.

Upon identifying the optimal mass weight w_{k^*} that minimizes the transport cost, we estimate the class prior as $\hat{\kappa} = 1/w_{k^*}$ (step 7). Subsequently, we employ $\hat{\kappa}$ to calculate the surrogate unbiased risk (5). Notably, while the surrogate risk (5) is inherently non-negative, its empirical approximation in (6) does not preserve this non-negativity. If ℓ is not upper-bounded, the empirical risk (6) may lack a lower bound, thereby risking model overfitting (Kiryo et al., 2017; Chaudhari and Shevade, 2012)³. Therefore, we nullify the negative values of $\tilde{\ell}$ by setting them to zero, so that negative terms are excluded from gradient calculation (step 8). Finally, the embedding tables are updated using stochastic gradient descent (step 9).

³This challenge is not exclusive to our framework; it has also been observed but not adequately addressed in prior methodologies within the ImplicitRec field (Saito et al., 2020; Saito, 2020).

4. Experiments

4.1. Experimental Setup

Dataset. We utilize three real-world datasets: **Yahoo! R3**, **Coat**, and **KuaiRec**. These datasets are selected since they uniquely provide negative feedback for evaluating model performance and include unbiased test sets that simulate production environments, aligning with established ImplicitRec studies (Saito et al., 2020; Saito, 2020; Ren et al., 2023). In Yahoo! R3 and Coat, user-item pairs with ratings above 4 are labeled as positive, and the rest as negative. In KuaiRec, records with counts below two are considered negative, while the others are positive. Each dataset is chronologically split into training, validation, and testing sets in a ratio of 0.8:0.1:0.1.

Baselines. In line with Ren et al. (2023), the collection of baselines encompasses models as follows.

- **RecVAE** and **DCF** are representative heuristic ImplicitRec models. **WMF** (Hu et al., 2008), **RMF** (Saito et al., 2020) and the clipped **CRMF** (Saito et al., 2020) are *unbiased* ImplicitRec methods with point-wise error measures.
- **UBPR** (Saito, 2020), **CUBPR** (Saito, 2020) and **UPL** (Ren et al., 2023) are *unbiased* ImplicitRec methods with pair-wise error measures. **UPL** (Ren et al., 2023) integrates the point-wise and pair-wise error measures, which to our knowledge is *currently the state-of-the-art method tailored for ImplicitRec*.
- **Oracle** is the recommender model that is trained using the ideal loss, but suffers from data sparsity of available explicit feedback (Wang et al., 2022) and selection bias. **SDR** (Li et al., 2023c), **CDR** (Song et al., 2023) and

UIDR (Li et al., 2024) correct this bias via propensity weighting, where propensities are heuristically estimated following Saito et al. (2020).

Training Protocol. We use matrix factorization as the primary recommendation model, following existing studies (Ren et al., 2023; Saito et al., 2020; Saito, 2020; Li et al., 2023c). Experiments are conducted with two Intel Xeon Platinum 8383C CPUs (2.70 GHz) and eight NVIDIA GeForce RTX 4090 GPU. We tune the learning rate in $\{0.005, 0.01, 0.05\}$, batch size in $\{256, 512, 1024, 2048\}$, and embedding size in $\{8, 16, 32\}$. All experiments are implemented in PyTorch using the Adam optimizer (Kingma and Ba, 2015) with early stopping (patience = 5). The candidate mass weights (w_k) are set to $[5, 10, 20, 50, 100]$ which balances searching spectrum and efficiency.

Evaluation Protocol. In the evaluation phase, the negative feedback is employed for performance evaluation. Two commonly used metrics are adopted for evaluation: Top- k recall (Recall@ k) and Top- k normalized discounted cumulative gain (NDCG@ k). We report results on Recall@ $\{1, 3, 5\}$ and NDCG@ $\{1, 3, 5\}$. To perform a significance test, all experiments are repeated for 5 times.

4.2. Overall Performance

The evaluation of our proposed methods and the baselines is summarized using the NDCG in Table 2 and Recall metrics in Table 5. We outline the key findings as follows:

- Conventional point-wise baselines exhibit practical performance but have inherent limitations: the sensitivity to small propensity scores leads to large estimation variances. Techniques like CRMF address this by capping propensities below a certain threshold, which introduces a bias and compromises performance.
- The pair-wise baselines such as UBPR and CUBPR also rely on propensity scores and hence share the limitations of point-wise baselines but generally outperform them. This improvement may come from the pair-wise loss, which penalize lower ratings for unlabeled samples rather than fixing them as negatives. Among the baselines, the UPL (Ren et al., 2023) approach, which combines point-wise and pair-wise strategies, is notably effective.
- Our PURL approach outperforms baselines across most datasets and metrics. This superiority may come from its unbiasedness and consistency without theoretical guarantees. Crucially, PURL operates independently of propensity score identification, avoiding issues related to small propensities and inaccurate propensity estimation in baselines (Saito, 2020; Saito et al., 2020).

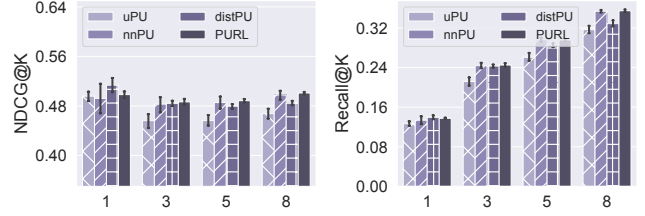


Figure 2: Varying risk estimator results on KuaiRec.

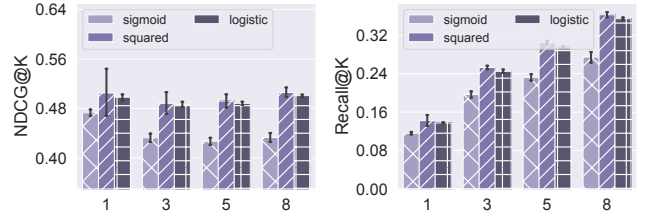


Figure 3: Varying error measure (ℓ) results on KuaiRec.

4.3. Discussion on the PURL estimator

In this section, we extend the PURL framework to accommodate various error measures and weakly supervised learning estimators.

- **Varying risk estimator results.** We incorporate three widely-used weakly supervised learning estimators into the PURL framework: uPU (du Plessis et al., 2014), nnPU (Kiryo et al., 2017), and distPU (Zhao et al., 2022). Our empirical results in Figure 2 indicate that all three estimators deliver commendable performance. Specifically, uPU, a seminal work in this area, offers theoretically unbiased risk estimates. However, it suffers from consequent overfitting due to the absence of a lower bound on the loss function. Subsequent nnPU and distPU address this issue by introducing non-negativity constraints, resulting in improved performance. However, they impose non-negativity constraints in an expectation sense, overlooking the role of individual non-negative terms within the expectation and thereby compromising sample efficiency. In contrast, our implementation advocates to adding non-negative constraint for each term individually, which enhances sample efficiency and marginally improves performance across six out of eight metrics.
- **Varying error measure results.** According to Theorem 4, the surrogate risk function (5) with a logistic error measure is convex, making it amenable to optimization via stochastic gradient methods. This property also extends to the squared loss error measure, which explains its comparable performance relative to the logistic measure in Figure 3. On the other hand, the sigmoid error measure renders the surrogate risk non-convex, which complicates optimization and causes suboptimal performance.

Table 2: Recommendation performances in terms of Recall@k on Yahoo! R3, Coat and KuaiRec.

Dataset	Yahoo! R3			Coat			KuaiRec		
Metrics	Recall@1	Recall@3	Recall@5	Recall@1	Recall@3	Recall@5	Recall@1	Recall@3	Recall@5
RMF	0.042 \pm 0.002	0.132 \pm 0.005	0.226 \pm 0.003	0.054 \pm 0.017	0.170 \pm 0.018	0.285 \pm 0.022	0.008 \pm 0.001	0.022 \pm 0.002	0.037 \pm 0.001
CRMF	0.042 \pm 0.002	0.132 \pm 0.004	0.226 \pm 0.004	0.055 \pm 0.014	0.168 \pm 0.018	0.289 \pm 0.030	0.009 \pm 0.002	0.029 \pm 0.003	0.047 \pm 0.001
WMF	0.094 \pm 0.002	0.203 \pm 0.003	0.288 \pm 0.002	0.061 \pm 0.011	0.192 \pm 0.025	0.311 \pm 0.035	0.015 \pm 0.003	0.058 \pm 0.007	0.104 \pm 0.008
UBPR	0.125 \pm 0.002	0.265 \pm 0.000	0.344 \pm 0.001	0.113 \pm 0.008	0.270 \pm 0.009	0.375 \pm 0.013	0.023 \pm 0.004	0.069 \pm 0.005	0.115 \pm 0.005
CUBPR	0.126 \pm 0.003	0.265 \pm 0.002	0.344 \pm 0.002	0.113 \pm 0.009	0.270 \pm 0.009	0.378 \pm 0.010	0.024 \pm 0.002	0.071 \pm 0.008	0.119 \pm 0.009
UPL	0.129 \pm 0.002	0.268 \pm 0.002	0.346 \pm 0.003	0.118 \pm 0.009	0.256 \pm 0.011	0.369 \pm 0.010	0.022 \pm 0.003	0.070 \pm 0.004	0.117 \pm 0.006
RecVAE	0.054 \pm 0.002	0.149 \pm 0.001	0.234 \pm 0.002	0.096 \pm 0.012	0.267 \pm 0.021	0.389 \pm 0.031	0.110 \pm 0.001	0.183 \pm 0.002	0.214 \pm 0.002
DCF	0.131 \pm 0.003	0.262 \pm 0.003	0.339 \pm 0.004	0.134 \pm 0.014	0.272 \pm 0.024	0.386 \pm 0.027	0.119 \pm 0.001	0.231 \pm 0.005	0.288 \pm 0.004
ORACLE	0.055 \pm 0.003	0.162 \pm 0.004	0.258 \pm 0.004	0.085 \pm 0.009	0.233 \pm 0.009	0.350 \pm 0.015	0.011 \pm 0.001	0.052 \pm 0.005	0.101 \pm 0.006
CDR	0.066 \pm 0.002	0.157 \pm 0.003	0.235 \pm 0.006	0.064 \pm 0.012	0.185 \pm 0.018	0.297 \pm 0.017	0.140 \pm 0.004	0.191 \pm 0.002	0.207 \pm 0.003
SDR	0.061 \pm 0.001	0.149 \pm 0.003	0.223 \pm 0.002	0.085 \pm 0.017	0.224 \pm 0.014	0.337 \pm 0.013	0.138 \pm 0.002	0.184 \pm 0.007	0.191 \pm 0.010
UIDR	0.075 \pm 0.021	0.168 \pm 0.035	0.245 \pm 0.037	0.060 \pm 0.009	0.163 \pm 0.011	0.268 \pm 0.013	0.133 \pm 0.037	0.206 \pm 0.038	0.235 \pm 0.033
PURL	0.146* \pm 0.002	0.280* \pm 0.003	0.354* \pm 0.002	0.131 \pm 0.016	0.302* \pm 0.019	0.425* \pm 0.019	0.137 \pm 0.002	0.245* \pm 0.005	0.296* \pm 0.004

Note: The results are reported in mean \pm std with 5 runs. The best and second best metrics are bolded and underlined, respectively. "*" marks the metrics that PURL surpasses the best baseline with p-value < 0.05 over paired samples t-test.

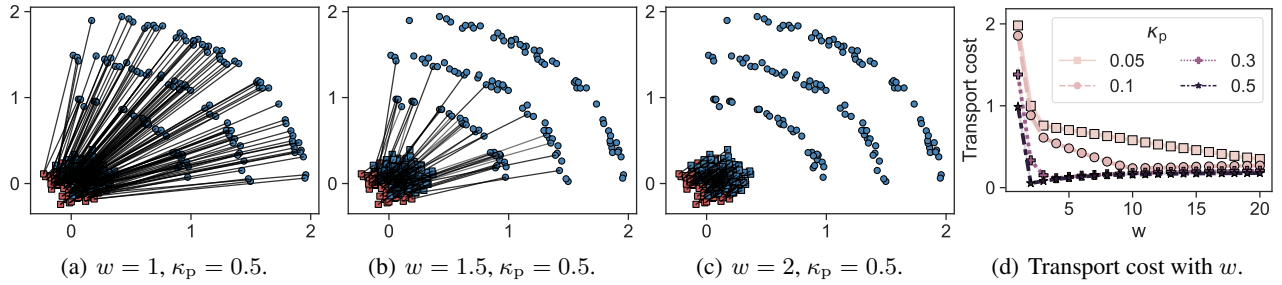


Figure 4: PT transport plan (a-c) and cost (d) with different w . Circular (rectangular) markers indicate positive (negative) samples. Blue (red) markers indicate labeled (unlabeled) samples.

Table 3: Varying fixed class prior results.

$\hat{\kappa}_p$	W	ΔW	NDCG@3	Δ NDCG@3	NDCG@5	Δ NDCG@5	Recall@3	Δ Recall@3	Recall@5	Δ Recall@5
0.2	6.1143	-	0.4022	-	0.4232	-	0.2102	-	0.2732	-
0.005	5.5727	8.85% \downarrow	0.4696	16.75% \uparrow	0.4717	11.46% \uparrow	0.2293	9.08% \uparrow	0.2857	4.57% \uparrow
0.01	5.5727	8.85% \downarrow	0.4731	17.62% \uparrow	0.4781	12.97% \uparrow	0.2326	10/65% \uparrow	0.2930	7.24% \uparrow
0.02	5.5716	8.87% \downarrow	0.4847	20.51% \uparrow	0.4867	15.00% \uparrow	0.2419	15.08% \uparrow	0.2954	8.12% \uparrow
0.05	5.5620	9.03% \downarrow	0.4847	20.51% \uparrow	0.4871	15.09% \uparrow	0.2455	16.79% \uparrow	0.2979	9.04% \uparrow
0.1	5.5951	8.49% \downarrow	0.4427	10.06% \uparrow	0.4564	7.84% \uparrow	0.2296	9.22% \uparrow	0.2886	5.63% \uparrow

Note: Δ denotes the relative deviation from the results with $\hat{\kappa}_p = 0.2$.

4.4. Discussion on the PPT Approach

We investigate the efficacy of the proposed PPT approach in class prior estimation. Owing to missing negative feedback, the actual class prior κ is unavailable. To showcase the efficacy of PPT, we make efforts from two aspects as follows.

4.4.1. NUMERICAL VERIFICATION

We conduct a case study using a simulated dataset with a predefined class prior κ_p . Figure 4 (a-c) illustrates the dataset and transport strategies for different class prior estimates w . The key observations are summarized as follows:

- Different values of w result in distinct matching strategies π . At $w = 1$, PT reduces to the standard Kantorovich problem, matching all unlabeled samples. With $w = 1.5$, PT matches positively labeled samples with concealed positives but also matches some negatives due to mass preservation, leading to high transport costs. Increasing w to 2 allows partial matching of only concealed positive samples, rectifying the transport plan and minimizing transport cost. Further increases in w , as shown in Figure 4 (d), do not rectify the transport plan and instead increase transport costs.
- The behavior of w proves suitable for estimating class priors. Figure 4 (d) shows that for a true class prior $\kappa_p = 0.3$, transport cost decreases sharply towards about 0.2 at $w = 0.4$, then gradually increases. According to Section 4.4, PPT estimates the class prior to be between 0.25 and 0.33, closely matching the predefined $\kappa_p = 0.3$. For other class priors of 0.05, 0.1, and 0.5, the minimum transport costs correspond to w values of 20, 10, and 2 respectively, supporting the relationship $\kappa_p = 1/w$.

4.4.2. IN-LOOP VERIFICATION

A reasonable hypothesis is that the class prior yielding the optimal performance likely corresponds to the ground truth, aligning well with both theoretical properties and practical requirements. Therefore, we firstly modify PURL to use fixed class prior estimates, $\hat{\kappa}$, and evaluate the resulting performance. Afterwards, we examine whether PPT’s estimated class prior approximates the fixed prior that achieves the best performance. This analysis is conducted using the large-scale KuaiRec dataset, with key results summarized in Table 3 and analyzed below.

- Accurate class prior estimation is critical for PURL’s performance. In Table 3, setting $\hat{\kappa}_p = 0.2$ results in an NDCG@5 of approximately 0.423. Performance consistently improves as $\hat{\kappa}_p$ decreases, reaching the peak of around 0.487 at $\hat{\kappa}_p = 0.05$. Further increasing $\hat{\kappa}_p$ to 0.1

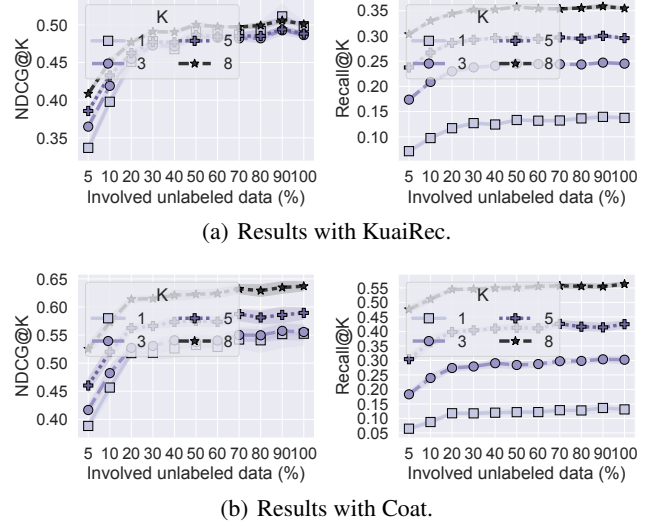


Figure 5: Results with varying ratios of unlabeled data for model training. The colored lines and shadowed areas represent mean values and 90% confidence intervals, respectively.

causes an immediate drop. Thus, the optimal class prior estimate is approximately $\hat{\kappa}_p = 0.05$.

- PPT effectively identifies the ground-truth class prior. Specifically, the PT cost is minimized at $\hat{\kappa}_p = 0.05$ at 5.5620, which corresponds to the best overall performance in Table 3. Therefore, the PPT approach successfully identifies the class prior that optimizes performance.

4.5. Scalability Analysis

According to Theorem 3, PURL correlates strongly with the volume of unlabeled samples used in training. Utilizing more unlabeled samples reduces the variance in PURL’s approximation of the ideal risk. This is particularly beneficial in ImplicitRec due to the abundance of unlabeled samples. This scalability is demonstrated in our experiments, as per Figure 5, where PURL is trained using varying proportions of unlabeled samples.

The results show a consistent improvement in performance with increasing proportions of unlabeled samples. For instance, using just 5% of unlabeled samples achieves an NDCG@1 of approximately 0.34 on the KuaiRec dataset, with performance peaking at around 0.5 when all unlabeled samples are included. These trends, which hold across other metrics and datasets, validate PURL’s theoretical consistency. Additionally, the rate of performance improvement tends to plateau beyond a certain threshold, indicating that while it is possible to use all unlabeled data, a subset may be sufficient to balance performance and training cost.

5. Related Works

5.1. Learning from Weak Supervision

Weakly supervised learning refers to a type of machine learning paradigm where the training data is partially labeled or noisily labeled (Sugiyama et al., 2022). Unlike supervised learning wherein each data point has a corresponding accurate label, weak supervised learning operates on data where the labels may be imprecise, incomplete, or altogether missing for some instances. This approach is particularly useful when obtaining fully labeled data is expensive or difficult.

Positive-Unlabeled (PU) learning is an important instance of weakly supervised learning, focusing on problems where the training set consists of labeled positive examples and unlabeled examples that could either be positive or negative, with applications in fundamental fields such as computer vision (Loghmani et al., 2020; Chapel et al., 2020) and text processing (Le et al., 2020; S. et al., 2022). The core challenge arises from the absence of negative labels, which impedes the application of traditional supervised learning models. There are two primary paradigms to tackle this challenge: re-weighting and pseudo-labeling. The re-weighting paradigm adjusts sample weights to achieve unbiased estimation of the ideal risk, exemplified by uPU (du Plessis et al., 2014). Subsequent modifications make uPU convex (du Plessis et al., 2015b) and non-negative (Kiryo et al., 2017; Chaudhari and Shevade, 2012; Zhang and Zuo, 2009), to accommodate with stochastic gradient optimizers. Recently, further refinements have been proposed to adapt handle data imbalance (Su et al., 2021) and sample selection bias (Hammoudeh and Lowd, 2020). On the other hand, the pseudo-labeling paradigm identifies reliable negative and positive samples from the unlabeled data and then turns the problem into a supervised learning task. Methods differ in the techniques of pseudo labels assignment, such as graph neural network (Chaudhari and Shevade, 2012; Zhang and Zuo, 2009), generative models (Hou et al., 2018), confidence score (Northcutt et al., 2017), reinforcement learning (Luo et al., 2021) and clustering analysis (Gong et al., 2019).

5.2. Recommendation with Implicit Feedback

ImplicitRec aims to infer user preferences from user behaviors such as clicks or views rather than explicit comments.

Compared with explicit feedback, implicit feedback is characterised by missing negative feedback (MNF). Specifically, the data involves some positive feedback and unlabeled data. The absence of negative feedback makes it difficult to differentiate between items that are disliked and those simply unexposed to the user. Current approaches treating unlabeled data as negative can lead to biased and inaccurate recommendations (Zhou et al., 2022; Kang and McAuley, 2018; Wang et al., 2022; Li et al., 2023c). An exemplar

approach to handle this issue is the weighted matrix factorization (Hu et al., 2008), which heuristically downweights the unobserved user-item samples versus the observed ones. ExpoMF (Liang et al., 2016) further extends it to a generative formulation. However, these methods are largely heuristic in nature and lack a rigorous theoretical underpinning, indicating a need for further exploration of this line of works.

Recent advancement further considers sample selection bias in the positively labeled data. Specifically, the observed positive feedback is not a random sample of a user’s true preferences; rather, it is influenced by various factors such as item popularity and visibility (Lim et al., 2015; Yang et al., 2018). This non-random sampling can distort the true underlying preference structure, resulting in inaccurate recommendations (Zhou et al., 2021; Gupta et al., 2023; Togashi et al., 2021). Current efforts mainly employ inverse propensity weighting (IPW) to correct such selection bias. It re-weights each observed sample using a propensity score, which theoretically ensures an unbiased modeling of user-item preferences (Wang et al., 2022; Li et al., 2023c;a). In practice, methods mainly differ in terms of the ways to estimate propensity, represented by employing the item popularity (Saito et al., 2020; Saito, 2020) and constructing a propensity estimator (Zhu et al., 2020; Lee et al., 2022). Despite its theoretical merits, IPW faces several practical challenges: (i) the propensity estimation is often inaccurate (Li et al., 2023c), exacerbated by the absence of negative feedback; (ii) the estimators are sensitive to small propensity scores (Li et al., 2023a), complicating the training process; and (iii) the non-negativity constraint, essential for model convergence, is often violated for the sake of theoretical unbiasedness (Saito et al., 2020; Saito, 2020). These limitations are not only theoretically complex but also empirically restrict the application of IPW-based approaches in ImplicitRec.

Recently, PU learning has also been incorporated for enhancing ImplicitRec (Togashi et al., 2021; Zhou et al., 2021). For example, Zhou et al. (2021) develop an adversarial approach using PU loss for discriminator training, Togashi et al. (2021) introduce a density-ratio-based method that leverages PU learning for density-ratio estimation. Our contributions are distinct from them: we are the first to integrate PU learning to enhance the binary classification objective for ImplicitRec and propose a class prior estimation method based on optimal transport, a significant contribution given the complexity of this issue in PU learning.

6. Conclusions

In conclusion, the challenge of missing negative samples has long been a significant obstacle in ImplicitRec. To handle this issue, we develop PURL, a model-agnostic framework

that adapts principles from weakly supervised learning to the domain of ImplicitRec. A cornerstone of our approach is the PPT approach designed to estimate the class prior by minimizing the proximal transport cost between positive and unlabeled samples. Extensive evaluations on real-world datasets confirm the efficacy of our approach and support the theoretical analysis empirically.

Limitation & Future works This study adheres to the setting of ID-based recommendation research and does not consider user profiles. Future work could extend this framework to more content-rich scenarios. Additionally, the scalability of PURL highlights the potential of using implicit feedback to enhance recommenders. Therefore, how to harmonize explicit and implicit feedback presents a promising direction for further research.

Acknowledgements

This work is supported by National Key R&D Program of China (2022ZD0160300) and the NSF China (No. 62276004, 623B2002).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

- Jason M. Altschuler, Jonathan Weed, and Philippe Rigollet. 2017. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *NeurIPS*. 1964–1974.
- Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM J. SCI. COMPUT.* 37, 2 (2015), A1111–A1138.
- Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. 2011. Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph.* 30, 6 (2011), 158.
- Laetitia Chapel, Mokhtar Z. Alaya, and Gilles Gasso. 2020. Partial Optimal Transport with applications on Positive-Unlabeled Learning. In *NeurIPS*.
- Sneha Chaudhari and Shirish K. Shevade. 2012. Learning from Positive and Unlabelled Examples Using Maximum Margin Clustering. In *NeurIPS*, Vol. 7665. 465–473.
- Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018. Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.* 87, 314 (2018), 2563–2609.
- Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2017. Optimal Transport for Domain Adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 9 (2017), 1853–1865.
- Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of Learning from Positive and Unlabeled Data. In *NeurIPS*. 703–711.
- Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. 2015a. Class-prior Estimation for Learning from Positive and Unlabeled Data. In *Proceedings of The 7th Asian Conference on Machine Learning*, Vol. 45. 221–236.
- Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. 2015b. Convex Formulation for Learning from Positive and Unlabeled Data. In *ICML*, Vol. 37. 1386–1394.
- Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. 2017. Class-prior estimation for learning from positive and unlabeled data. *Machine Learning* 106, 4 (2017), 463–492.
- Pavel E. Dvurechensky, Alexander V. Gasnikov, and Alexey Kroshnin. 2018. Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn’s Algorithm. In *ICML*, Vol. 80. 1366–1375.
- Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. 2021. POT: Python Optimal Transport. *J. Mach. Learn. Res.* 22 (2021), 78:1–78:8.
- Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022a. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *CIKM*. ACM, 540–550.
- Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022b. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *CIKM*. ACM, 3953–3957.

- Carlos A. Gomez-Urbe and Neil Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manage. Inf. Syst.* 6, 4 (2016), 19 pages.
- Chen Gong, Hong Shi, Tongliang Liu, Chuang Zhang, Jian Yang, and Dacheng Tao. 2019. Loss decomposition and centroid estimation for positive and unlabeled learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3 (2019), 918–932.
- Shashank Gupta, Harrie Oosterhuis, and Maarten de Rijke. 2023. A Deep Generative Recommendation Method for Unbiased Learning from Implicit Feedback. In *ICTIR*. ACM, 87–93.
- LLC Gurobi Optimization. 2020. Gurobi optimizer reference manual (2020).
- Zayd Hammoudeh and Daniel Lowd. 2020. Learning from Positive and Unlabeled Data with Arbitrary Positive Shift. In *NeurIPS*.
- William E Hart, Jean-Paul Watson, and David L Woodruff. 2011. Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation* 3 (2011), 219–260.
- Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. 2018. Generative adversarial positive-unlabeled learning. In *IJCAI*. 2255–2261.
- Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. IEEE Computer Society.
- Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. 197–206.
- Leonid V Kantorovich. 2006. On the translocation of masses. *J. Math. Sci.* 133, 4 (2006), 1381–1382.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
- Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. 2017. Positive-Unlabeled Learning with Non-Negative Risk Estimator. In *NeurIPS*. 1675–1685.
- Triet Huynh Minh Le, David Hin, Roland Croft, and Muhammad Ali Babar. 2020. PUMiner: Mining Security Posts from Developer Question and Answer Websites with PU Learning. In *MSR*. ACM, 350–361.
- Jae-woong Lee, Seongmin Park, Joonseok Lee, and Jongwuk Lee. 2022. Bilateral Self-unbiased Learning from Biased Implicit Feedback. In *SIGIR*. ACM, 29–39.
- Haoxuan Li, Yan Lyu, Chunyuan Zheng, and Peng Wu. 2023a. TDR-CL: Targeted Doubly Robust Collaborative Learning for Debiased Recommendations. In *ICLR*.
- Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, Peng Wu, and Peng Cui. 2023b. Propensity Matters: Measuring and Enhancing Balancing for Recommendation. In *ICML*, Vol. 202. PMLR, 20182–20194.
- Haoxuan Li, Chunyuan Zheng, Shuyi Wang, Kunhan Wu, Eric Wang, Peng Wu, Zhi Geng, Xu Chen, and Xiao-Hua Zhou. 2024. Relaxing the Accurate Imputation Assumption in Doubly Robust Learning for Debiased Collaborative Filtering. In *Forty-first International Conference on Machine Learning*.
- Haoxuan Li, Chunyuan Zheng, and Peng Wu. 2023c. StableDR: Stabilized Doubly Robust Learning for Recommendation on Data Missing Not at Random. In *ICLR*.
- Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation. In *WWW*. ACM, 951–961.
- Daryl Lim, Julian J. McAuley, and Gert R. G. Lanckriet. 2015. Top-N Recommendation with Missing Implicit Feedback. In *RecSys*. ACM, 309–312.
- Haibin Ling and Kazunori Okada. 2007. An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 5 (2007), 840–853.
- Mohammad Reza Loghmani, Markus Vincze, and Tatiana Tommasi. 2020. Positive-unlabeled learning for open set domain adaptation. *Pattern Recognit. Lett.* 136 (2020), 198–204.
- Chuan Luo, Pu Zhao, Chen Chen, Bo Qiao, Chao Du, Hongyu Zhang, Wei Wu, Shaowei Cai, Bing He, Saravanakumar Rajmohan, and Qingwei Lin. 2021. PULNS: Positive-Unlabeled Learning with Effective Negative Sample Selector. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. 8784–8792.
- Simone Di Marino and Augusto Gerolin. 2020. An Optimal Transport Approach for the Schrödinger Bridge Problem and Convergence of Sinkhorn Algorithm. *J. Sci. Comput.* 85, 2 (2020), 27.
- Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.* (1781), 666–704.
- Gang Niu, Marthinus Christoffel Du Plessis, Tomoya Sakai, Yao Ma, and Masashi Sugiyama. 2016. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. *NeurIPS* 29 (2016).

- Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang. 2017. Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels. In *UAI*. 10 pages.
- Ofir Pele and Michael Werman. 2009. Fast and robust Earth Mover’s Distances. In *ICCV*. 460–467.
- Yi Ren, Hongyan Tang, Jiangpeng Rong, and Siwen Zhu. 2023. Unbiased Pairwise Learning from Implicit Feedback for Recommender Systems without Biased Variance Control. In *SIGIR*. ACM, 2461–2465.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, 452–461.
- Vinay M. S., Shuhan Yuan, and Xintao Wu. 2022. Fraud Detection via Contrastive Positive Unlabeled Learning. In *IEEE Big Data*. IEEE, 1475–1484.
- Yuta Saito. 2020. Unbiased Pairwise Learning from Biased Implicit Feedback. In *ICTIR*. ACM, 5–12.
- Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *WSDM*. ACM, 501–509.
- Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In *AAAI*. 4058–4065.
- Brent Smith and Greg Linden. 2017. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing* 21, 3 (2017), 12–18. <https://doi.org/10.1109/MIC.2017.72>
- Zijie Song, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng, Chun Chen, and Can Wang. 2023. CDR: Conservative doubly robust learning for debiased recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2321–2330.
- Guangxin Su, Weitong Chen, and Miao Xu. 2021. Positive-Unlabeled Learning from Imbalanced Data. In *IJCAI*. 2995–3001.
- Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, and Tomoya Sakai. 2022. *Machine learning from weak supervision: An empirical risk minimization approach*. MIT Press.
- Riku Togashi, Masahiro Kato, Mayu Otani, and Shin’ichi Satoh. 2021. Density-Ratio Based Personalised Ranking from Implicit Feedback. In *WWW*. ACM, 3221–3233.
- Hao Wang, Tai-Wei Chang, Tianqiao Liu, Jianmin Huang, Zhichao Chen, Chao Yu, Ruopeng Li, and Wei Chu. 2022. ESCM2: Entire Space Counterfactual Multi-Task Model for Post-Click Conversion Rate Estimation. In *SIGIR*. 363–372.
- Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 373–381.
- Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin. 2019. Gromov-Wasserstein Learning for Graph Matching and Node Embedding. In *ICML*, Vol. 97. PMLR, 6932–6941.
- Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary Learning via Optimal Transport for Neural Machine Translation. In *ACL/IJCNLP (1)*. ACL, 7361–7373.
- Renjun Xu, Pelen Liu, Yin Zhang, Fang Cai, Jindong Wang, Shuoying Liang, Heting Ying, and Jianwei Yin. 2020. Joint Partial Optimal Transport for Open Set Domain Adaptation. In *IJCAI*. 2540–2546.
- Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge J. Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*. ACM, 279–287.
- Bangzuo Zhang and Wanli Zuo. 2009. Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples. *Journal of Computers* 4, 1 (2009), 94–101.
- Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. 2022. Dist-PU: Positive-Unlabeled Learning from a Label Distribution Perspective. In *CVPR*. IEEE, 14441–14450.
- Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *SIGKDD*. 1059–1068.
- Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *WWW*. 2388–2399.
- Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi Nasrabadi, Evren Körpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In *SIGKDD*. ACM, 2409–2419.

Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020.
Unbiased Implicit Recommendation and Propensity Estimation via Combinational Joint Learning. In *RecSys*. ACM, 551–556.

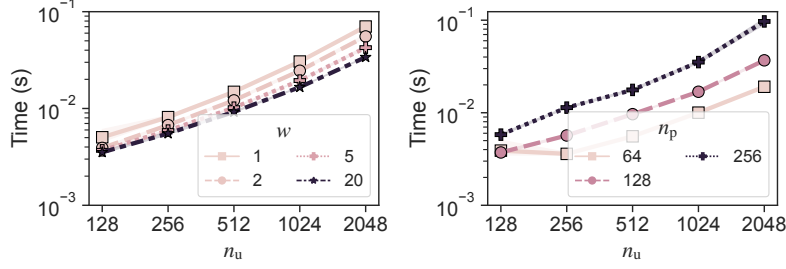


Figure 6: Running time of solving the PT problem. By default we set $n_p = 128$ and $w = 10$.

A. Additional Experimental Results

A.1. Dataset Description

In this section, we present a brief introduction to the datasets involved in this study.

- **Yahoo! R3** contains 311,704 biased ratings for training, which involve 15,400 users and 1,000 items. Additionally, 5400 users rate 10 randomly selected items, yielding 54,000 unbiased ratings for evaluation. The user-item pairs with rating greater than 4 are seen as positive, and others are viewed as negative.
- **Coat** is a public dataset which consists of 290 users and 300 items; each user subjectively selects 24 items to rate based on their preference, yielding 6,960 biased ratings in the training set. Additionally, each user rates 16 items that are randomly selected, yielding 4,640 unbiased ratings for model evaluation. The user-item pairs with rating greater than 4 are seen as positive, and others are viewed as negative.
- **KuaiRec** is a public large-scale industrial dataset, which consists of 4,676,570 video watching ratio records from 1,411 users for 3,327 videos. The user-item pairs with ratings less than two are viewed as negative, and otherwise are viewed as positive. The records less than two are viewed as negative feedback, and otherwise are viewed as negative feedback.

A.2. Discussion on Complexity

Concerns may arise regarding the computational overhead introduced by PURL, given that it incorporates a linear programming procedure—the solving of the PT problem—in each iteration of model training. These concerns are particularly pertinent in the context of large batch sizes commonly employed in training recommender models, as this could render the PT problem large-scale and computationally intensive. To address these concerns, we empirically evaluate the time required for solving the PT problem under various settings, as depicted in Figure 6. We summarize two key observations as follows.

- The computational complexity of solving the PT problem is not prohibitive in our specific application. For instance, even when matching as many as 4096 unlabeled samples, the problem can be solved within 0.1 seconds. This relatively low computational burden is primarily due to the scarcity of positively labeled samples. In contrast to standard optimal transport (OT) scenarios where two distributions have a similar number of samples, the PT problem in our case involves matching n_u unlabeled samples with a significantly smaller number of n_p positive samples. Given that $n_p \ll n_u$, the computational cost is substantially lower than in traditional OT applications.
- The computational complexity is also influenced by the value of the mass weight w . Specifically, as w decreases, more unlabeled samples are compelled to engage in matching due to the mass-preserving constraint, thereby increasing the computational cost. When $w = 1$, the PT problem reduces to the canonical Kantorovich problem. However, given that the class prior is generally small in ImplicitRec applications, candidate values for w can be set to relatively large values, which further reduces the complexity of solving PT in PURL.

A.3. Hyperparameter Sensitivity

We perform a comprehensive sensitivity analysis to examine the influence of four critical hyperparameters, namely batch size, learning rate, and embedding dimension, on the performance of the PURL model using the coat and KuaiRec datasets. The results of these experiments are illustrated in Figure 8 and 7, respectively.

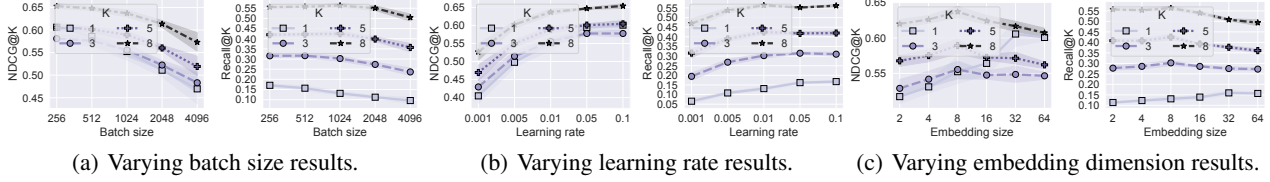


Figure 7: Parameter sensitivity studies of the PURL framework on Coat. Different colors indicate ranking metrics given different numbers of top candidates (K).

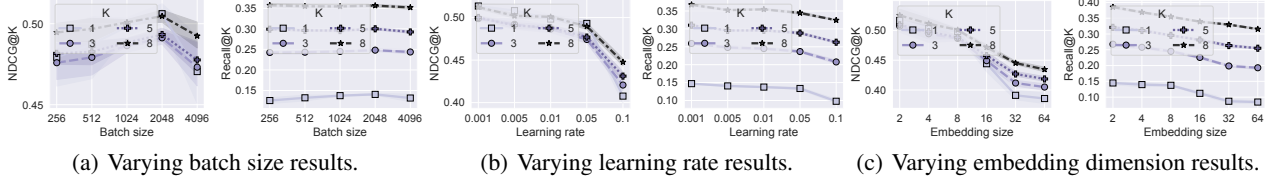


Figure 8: Parameter sensitivity studies of the PURL framework on KuaiRec. Different colors indicate ranking metrics given different numbers of top candidates (K).

Initially, we examine the effects of varying batch sizes on two distinct datasets. Interestingly, as the batch size increases, metrics such as NDCG@K and Recall@K generally show a decreasing trend. However, the KuaiRec dataset exhibits an initial increase followed by a decrease in these performance measures. This fluctuation can potentially be attributed to the interplay between the inherent characteristics of the PPT algorithm, which is integral to our model architecture, and the properties of the datasets in question, as well as their batch sizes. Specifically, smaller batch sizes may lead the PPT algorithm to experience generalization issues, adversely impacting the model’s overall performance. Conversely, larger batch sizes could introduce an over-smoothing problem, thereby reducing performance. These observations are further corroborated by data from the Coat and KuaiRec datasets in the batch size intervals of [256, 4096] and [2048, 4096] respectively. As for the interval [256, 2048] in the KuaiRec dataset, this behavior can be explained by the dataset’s size and inherent properties. Given the large size of the KuaiRec dataset, which primarily focuses on video clips, there is likely a significant level of noise. Furthermore, the dataset’s user base, representing a variety of socio-economic strata, tends to rapidly dismiss recommended items. Consequently, smaller batch sizes may make the model more susceptible to noise.

Subsequently, we tuned the learning rate and observe divergent trends in performance across the Coat and KuaiRec datasets. Specifically, an increase in the learning rate leads to an improvement in performance metrics on the Coat dataset, while causing a decline on the KuaiRec dataset. This differential behavior can largely be attributed to the contrasting sizes of the two datasets. The KuaiRec dataset is considerably larger than the Coat dataset, and this disparity likely influences the performance outcomes. In larger datasets like KuaiRec, the presence of noise becomes more prevalent, which may adversely affect model performance when higher learning rates are employed.

Additionally, we investigate the influence of varying the embedding dimension, which ranges from 2 to 64. We observe that the model performance initially improves, only to subsequently decline. This fluctuating pattern can be linked to the inherent properties of the PPT algorithm integrated into our model. PPT computations are executed in the embedding space, and an overly restricted dimensionality may lack the capacity to accurately capture the data’s complex manifold, leading to a degradation in model performance. On the other hand, an excessively large embedding dimension risks introducing redundancy and noise, thereby undermining the effectiveness of the optimal transport problem solution, which in turn may not reflect the true data manifold. For the KuaiRec dataset, a monotonically decreasing performance trend is evident. This observation lends further credence to our assertion that the KuaiRec dataset likely contains significant noise. Such noise interferes with the accurate representation of the true data manifold within the PPT algorithm, thereby adversely affecting the model performance.

In a broad range of hyperparameter settings, the PURL model consistently outperforms a majority of baseline models, further substantiating its superior performance characteristics. Specifically, even under varying conditions such as different learning rates, batch sizes, and embedding dimensions, the PURL model still excels or at least matches other commonly-used

Table 4: Recommendation performances in terms of NDCG@k on Yahoo! R3, Coat and KuaiRec.

Dataset	Yahoo! R3			Coat			KuaiRec		
Metrics	NDGC@1	NDGC@3	NDGC@5	NDGC@1	NDGC@3	NDGC@5	NDGC@1	NDGC@3	NDGC@5
RMF	0.632 \pm 0.002	0.667 \pm 0.003	0.707 \pm 0.002	0.370 \pm 0.022	0.394 \pm 0.021	0.433 \pm 0.021	0.209 \pm 0.006	0.211 \pm 0.006	0.215 \pm 0.005
CRMF	0.633 \pm 0.003	0.667 \pm 0.002	0.706 \pm 0.003	0.377 \pm 0.019	0.400 \pm 0.016	0.439 \pm 0.019	0.211 \pm 0.007	0.217 \pm 0.006	0.222 \pm 0.005
WMF	0.679 \pm 0.030	0.710 \pm 0.029	0.745 \pm 0.028	0.402 \pm 0.025	0.427 \pm 0.018	0.465 \pm 0.020	0.246 \pm 0.049	0.256 \pm 0.042	0.270 \pm 0.043
UBPR	0.757 \pm 0.003	0.795 \pm 0.001	0.826 \pm 0.002	0.459 \pm 0.028	0.493 \pm 0.021	0.530 \pm 0.014	0.255 \pm 0.010	0.272 \pm 0.010	0.290 \pm 0.011
CUBPR	0.756 \pm 0.003	0.795 \pm 0.002	0.826 \pm 0.002	0.460 \pm 0.030	0.493 \pm 0.020	0.531 \pm 0.015	0.257 \pm 0.007	0.273 \pm 0.009	0.292 \pm 0.010
UPL	0.760 \pm 0.004	0.798 \pm 0.002	0.829 \pm 0.002	0.508 \pm 0.028	0.511 \pm 0.012	0.540 \pm 0.010	0.281 \pm 0.034	0.297 \pm 0.034	0.315 \pm 0.036
ORACLE	0.654 \pm 0.003	0.693 \pm 0.003	0.734 \pm 0.003	0.425 \pm 0.026	0.457 \pm 0.019	0.496 \pm 0.017	0.212 \pm 0.011	0.231 \pm 0.008	0.251 \pm 0.008
PURL	0.787 \pm 0.004	0.817 \pm 0.003	0.845 \pm 0.003	0.579 \pm 0.035	0.567 \pm 0.019	0.597 \pm 0.015	0.489 \pm 0.013	0.480 \pm 0.009	0.483 \pm 0.007

Note: The results are reported in mean \pm std with 5 runs. The best and second best metrics are bolded and underlined, respectively. "*" marks the metrics that PURL surpasses the best baseline with p-value < 0.05 over paired samples t-test.

recommendation algorithms in multiple evaluation metrics. This further confirms the robustness and versatility of the PURL model in practical recommendation system applications. Moreover, by conducting more fine-grained hyperparameter tuning, the PURL model has the potential to achieve even better performance than reported in the main text. This implies that, after more precise hyperparameter search and validation, the PURL model can not only maintain its advantage over baseline models but also possibly reach higher performance levels in specific application scenarios or datasets. This further highlights the superior and adaptive nature of the PURL model, making it worth further exploration in real-world applications.

A.4. Supplementary Numerical Examples

To validate the efficacy of PPT in class prior estimation, we include a case study using a numerically simulated dataset with predefined (and known) class priors. In the main text, we only present the matching strategy on the simulated case wherein $\kappa_p = 0.5$. Nevertheless, it is essential to consider other simulation conditions to back up the utility of the proposed approach.

To achieve this, we extend our numerical verification by considering a range of class priors $\kappa_p = 0.05, 0.1, 0.3, 0.5$. We visualize the resulting transport strategies for different mass weights w . The key insights are consistent with those presented in the main text. The transport strategies are optimized when $w = 1/\kappa_p$, at which point the transport cost reaches its minimum, as illustrated in Figure 4. These additional numerical scenarios further corroborate the effectiveness of the Progressive Proximal Transport (PPT) method in accurately estimating the class prior, thereby reinforcing the validity of our PURL framework.

A.5. Performance with different backbone

In the main text, we implement the recommendation model g as MF, since it is a widely used model to evaluate the efficacy of learning objectives in recommendation system (Li et al., 2023c;a), especially in recent ImplicitRec researches (Ren et al., 2023; Lee et al., 2022; Zhu et al., 2020; Saito et al., 2020). Nevertheless, we agree that it is interesting to investigate the performance of different ImplicitRec estimators in different backbones to test their generality. Therefore, we select NCF as our new backbone and summarize the experimental results in Table 4 and Table 5, which showcase trends similar to the results on MF.

B. Theoretical Justification

B.1. Formal Analysis on the Surrogate Risk

Theorem 1. Given that the ideal risk of the recommender g defined over all samples is $R_{\text{ideal}}(g)$, the surrogate risk $\hat{R}_{\text{purl}}(g)$ as defined in (6) is an unbiased estimator of $R_{\text{ideal}}(g)$:

$$\mathbb{E}_{\mathcal{X}} [\hat{R}_{\text{purl}}(g)] = R_{\text{ideal}}(g).$$

Proof. Let \mathcal{X} be the empirical distribution of samples for training, with positive samples \mathcal{X}_p and unlabeled samples \mathcal{X}_u . According to the formulation of PURL in (6), the expectations regarding different training dataset can be formulated as

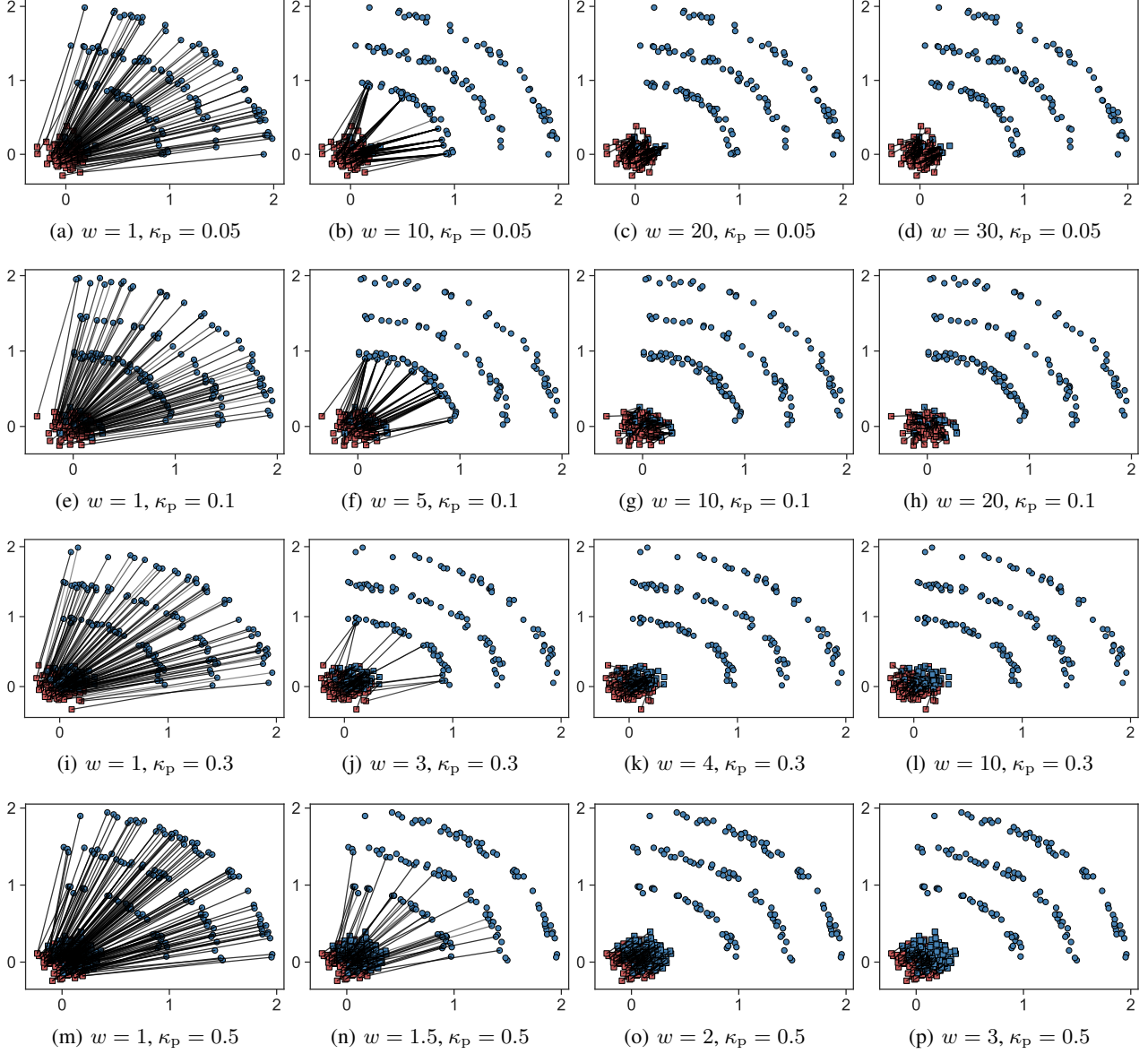


Figure 9: Overview of the PT transport plan given different weights of mass w and ground truth positive priors κ_p . The ground-truth positive and negative feedbacks are denoted by circular and rectangular markers, respectively. The labeled and unlabeled intersections are indicated in red and blue, respectively. The best transport strategies, which minimize transport costs for different values of κ_p , are illustrated in subfigures (c, g, k, o), characterized by $1/w \approx \kappa_p$.

Table 5: Recommendation performances in terms of Recall@k on Yahoo! R3, Coat and KuaiRec.

Dataset	Yahoo! R3			Coat			KuaiRec		
Metrics	Recall@1	Recall@3	Recall@5	Recall@1	Recall@3	Recall@5	Recall@1	Recall@3	Recall@5
RMF	0.042 \pm 0.002	0.133 \pm 0.004	0.226 \pm 0.003	0.054 \pm 0.013	0.158 \pm 0.022	0.270 \pm 0.026	0.009 \pm 0.001	0.025 \pm 0.005	0.042 \pm 0.006
CRMF	0.042 \pm 0.002	0.132 \pm 0.003	0.226 \pm 0.004	0.056 \pm 0.011	0.167 \pm 0.016	0.277 \pm 0.024	0.009 \pm 0.002	0.030 \pm 0.004	0.049 \pm 0.005
WMF	0.074 \pm 0.021	0.175 \pm 0.030	0.260 \pm 0.029	0.066 \pm 0.013	0.192 \pm 0.019	0.306 \pm 0.025	0.023 \pm 0.020	0.066 \pm 0.033	0.108 \pm 0.044
UBPR	0.126 \pm 0.002	0.265 \pm 0.001	0.344 \pm 0.002	0.100 \pm 0.017	0.254 \pm 0.020	0.373 \pm 0.013	0.026 \pm 0.004	0.078 \pm 0.011	0.128 \pm 0.015
CUBPR	0.126 \pm 0.002	0.265 \pm 0.002	0.343 \pm 0.003	0.100 \pm 0.018	0.254 \pm 0.020	0.374 \pm 0.012	0.026 \pm 0.003	0.078 \pm 0.010	0.130 \pm 0.014
UPL	<u>0.129\pm0.004</u>	<u>0.267\pm0.003</u>	<u>0.345\pm0.002</u>	<u>0.117\pm0.012</u>	<u>0.261\pm0.011</u>	<u>0.367\pm0.010</u>	<u>0.033\pm0.012</u>	<u>0.096\pm0.028</u>	<u>0.153\pm0.039</u>
ORACLE	0.056 \pm 0.002	0.163 \pm 0.003	0.260 \pm 0.003	0.078 \pm 0.011	0.221 \pm 0.019	0.341 \pm 0.021	0.014 \pm 0.004	0.054 \pm 0.006	0.102 \pm 0.008
PURL	0.148\pm0.003	0.282\pm0.004	0.356\pm0.003	0.149\pm0.023	0.308\pm0.015	0.423\pm0.016	0.133\pm0.007	0.245\pm0.005	0.296\pm0.003

Note: The results are reported in mean \pm std with 5 runs. The best and second best metrics are bolded and underlined, respectively. ”**” marks the metrics that PURL surpasses the best baseline with p-value < 0.05 over paired samples t-test.

follow:

$$\begin{aligned} \mathbb{E}_{\mathcal{X}} [\hat{R}_{\text{purl}}(g)] &= \mathbb{E}_{\mathcal{X}_p} \left[\frac{\kappa_p}{n_p} \sum_{i=1}^{n_p} \tilde{\ell}(g(x_i^p)) \right] \\ &\quad + \mathbb{E}_{\mathcal{X}_u} \left[\frac{1}{n_u} \sum_{i=1}^{n_u} \ell(g(x_i^u), -1) \right] \end{aligned} \quad (8)$$

For the first term, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{X}_p} \left[\frac{\kappa_p}{n_p} \sum_{i=1}^{n_p} \tilde{\ell}(g(x_i^p)) \right] &= \frac{\kappa_p}{n_p} \sum_{i=1}^{n_p} \mathbb{E}_{\mathcal{X}_p} [\tilde{\ell}(g(x_i^p))] \\ &= \frac{\kappa_p}{n_p} \sum_{i=1}^{n_p} \mathbb{E}_{x_i^p} [\tilde{\ell}(g(x_i^p))] \\ &= \frac{\kappa_p}{n_p} \sum_{i=1}^{n_p} \mathbb{E}_{p_p} [\tilde{\ell}(g(x))] \\ &= \kappa_p \mathbb{E}_{p_p} [\tilde{\ell}(g(x^p))] \end{aligned} \quad (9)$$

For the second term, similarly we have

$$\mathbb{E}_{\mathcal{X}_u} \left[\frac{1}{n_u} \sum_{i=1}^{n_u} \ell(g(x_i^u), -1) \right] = \mathbb{E}_{p_u} [\ell(g(x), -1)]. \quad (10)$$

Combining (8)-(10), we have

$$\begin{aligned} \mathbb{E}_{\mathcal{X}} [\hat{R}_{\text{purl}}(g)] &= \kappa_p \mathbb{E}_{p_p} [\tilde{\ell}(g(x))] + \mathbb{E}_{p_u} [\ell(g(x), -1)] \\ &= \kappa_p \mathbb{E}_{p_p} [\ell(g(x), +1)] \\ &\quad + (1 - \kappa_p) \mathbb{E}_{p_u} [\ell(g(x), -1)] \\ &= R_{\text{ideal}}(g). \end{aligned} \quad (11)$$

□

Lemma 1. Let \mathcal{G} be a family of recommender models. For any $\delta > 0$, with probability at least $1 - \delta$, [Niu et al. \(2016\)](#) demonstrate:

$$\sup_{g \in \mathcal{G}} |\hat{R}_{\text{purl}}(g) - R(g)| \leq 4\kappa_p L_\ell \mathfrak{R}_p + 2L_\ell \mathfrak{R}_u + 2\kappa_p \sqrt{\frac{\ln(4/\delta)}{2n_p}} + \sqrt{\frac{\ln(4/\delta)}{2n_u}}.$$

where \mathfrak{R}_p and \mathfrak{R}_u are the Rademacher complexities of \mathcal{G} given n_p positive samples and n_u unlabeled samples, respectively. L_ℓ is the Lipschitz constant of ℓ with respect to g .

Theorem 2 (Variance reduction). *The variance of R_{purI} is smaller than that of the ideal loss R_{ideal} given $\kappa_p \leq 0.5$.*

Proof. The variance of the ideal loss can be expressed as

$$\begin{aligned} \text{Var}(\hat{R}_{\text{ideal}}(g)) &= \left(\frac{\kappa_p^2}{n_p} \text{Var}(\ell(g(x^p), +1)) \right) + \left(\frac{\kappa_n^2}{n_n} \text{Var}(\ell(g(x^n), -1)) \right) \\ &= \frac{\kappa_p^2}{n_p} \sigma_p^2 + \frac{\kappa_n^2}{n_n} \sigma_n^2, \end{aligned}$$

where $\sigma_p^2 = \text{Var}(\ell(g(x^p), +1))$, $\sigma_n^2 = \text{Var}(\ell(g(x^n), -1))$. The variance of $\hat{R}_{\text{purI}}(g)$ is:

$$\begin{aligned} \text{Var}(\hat{R}_{\text{purI}}(g)) &= \left(\frac{\kappa_p^2}{n_p} \text{Var}(\tilde{\ell}(g(x^p))) \right) + \left(\frac{1}{n_u} \text{Var}(\ell(g(x^u), -1)) \right) \\ &= \left(\frac{\kappa_p^2}{n_p} \text{Var}(\ell(g(x), +1)) + \frac{\kappa_p^2}{n_p} \text{Var}(\ell(g(x), -1)) \right) \\ &\quad + \left(\frac{1}{n_u} \text{Var}(\ell(g(x^u), -1)) \right) \\ &= \frac{\kappa_p^2}{n_p} (\sigma_p^2 + \sigma_n^2) + \frac{1}{n_u} \sigma_n^2, \end{aligned}$$

where the covariance between the two terms is zero, which immediately holds if positive and unlabeled samples are independently drawn.

To show that $\text{Var}(\hat{R}_{\text{purI}}(g)) \leq \text{Var}(\hat{R}_{\text{ideal}}(g))$, it is equivalent to show that

$$\frac{\kappa_p^2}{n_p} (\sigma_p^2 + \sigma_n^2) + \frac{1}{n_u} \sigma_n^2 \leq \frac{\kappa_p^2}{n_p} \sigma_p^2 + \frac{\kappa_n^2}{n_n} \sigma_n^2.$$

That is:

$$\frac{\kappa_p^2}{n_p} + \frac{1}{n_u} \leq \frac{\kappa_n^2}{n_n}.$$

Since $n_u \gg n_p, n_n$, $1/n_u \rightarrow 0$. Moreover, since $n_p/n_n = \kappa_p/\kappa_n$, the above inequality holds if:

$$\kappa_p \leq \kappa_n,$$

which holds if $\kappa_p \leq 0.5$, a condition naturally holds in most recommendation scenarios where the positive samples are sparse and much smaller than positive samples. □

Theorem 3. *Let \mathcal{G} be a family of recommender models; $g^* = \arg \min_{g \in \mathcal{G}} R_{\text{ideal}}(g)$ be the ground-truth optimal recommender acquired by minimizing the ideal risk (1), $g^\dagger = \arg \min_{g \in \mathcal{G}} \hat{R}_{\text{purI}}(g)$ be the recommender acquired by minimizing (6). For any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$R(g^\dagger) - R(g^*) \leq 8\kappa_p L_\ell \mathfrak{R}_p + 4L_\ell \mathfrak{R}_u + \kappa_p \sqrt{\frac{8 \ln(4/\delta)}{n_p}} + \sqrt{\frac{2 \ln(4/\delta)}{n_u}},$$

where \mathfrak{R}_p and \mathfrak{R}_u are the Rademacher complexities of \mathcal{G} given n_p positive samples and n_u unlabeled samples, respectively. L_ℓ is the Lipschitz constant of ℓ with respect to g .

Proof. Based on Lemma 1, we have

$$\begin{aligned}
 R(g^\dagger) - R(g^*) &= (\widehat{R}_{\text{pur1}}(g^\dagger) - \widehat{R}_{\text{pur1}}(g^*)) \\
 &\quad + (R(g^\dagger) - \widehat{R}_{\text{pur1}}(g^\dagger)) + (\widehat{R}_{\text{pur1}}(g^*) - R(g^*)) \\
 &\leq 0 + (R(g^\dagger) - \widehat{R}_{\text{pur1}}(g^\dagger)) + (\widehat{R}_{\text{pur1}}(g^*) - R(g^*)) \\
 &\leq 0 + 2 \sup_{g \in \mathcal{G}} |\widehat{R}_{\text{pur1}}(g) - R(g)| \\
 &\leq 8\kappa_p L_\ell \mathfrak{R}_p + 4L_\ell \mathfrak{R}_u + \kappa_p \sqrt{\frac{8 \ln(4/\delta)}{n_p}} + \sqrt{\frac{2 \ln(4/\delta)}{n_u}},
 \end{aligned}$$

□

Lemma 2. For any $\ell(\hat{y}, y)$ that is convex in \hat{y} , $\ell(\hat{y}, y) - \ell(\hat{y}, -y)$ is convex in \hat{y} if and only $\ell(\hat{y}, y) - \ell(\hat{y}, -y)$ is linear in \hat{y} .

Proof. The lemma comes from the Theorem 4.1 by Sugiyama et al. (2022). We provide a straight-forward proof here. Since $\ell(\hat{y}, y)$ is convex, we have:

$$\ell(\hat{y}_1 + \hat{y}_2, y) \leq \ell(\hat{y}_1, y) + \ell(\hat{y}_2, y)$$

To make $\ell(\hat{y}, y) - \ell(\hat{y}, -y)$ convex, the term $-\ell(\hat{y}, -y)$ should be convex. Therefore, we have:

$$\ell(\hat{y}_1 + \hat{y}_2, y) \geq \ell(\hat{y}_1, y) + \ell(\hat{y}_2, y)$$

Combining the two inequalities, we conclude that to make $\ell(\hat{y}, y) - \ell(\hat{y}, -y)$ convex, the sufficient and necessary condition is: $\ell(\hat{y}_1 + \hat{y}_2, y) = \ell(\hat{y}_1, y) + \ell(\hat{y}_2, y)$, which indicates that $\ell(\hat{y}, y) - \ell(\hat{y}, -y)$ is linear in \hat{y} . □

Theorem 4. Consider a recommender g where the output score s is transformed by a sigmoid function such that: $g(x) = \text{sigmoid}(s(x))$. Consider the logistic error measure, the surrogate risk $R_{\text{pur1}}(g)$ is convex with respect to the score s .

Proof. The surrogate risk $R_{\text{pur1}}(g)$ consists of two terms: $\hat{\ell}(g(x))$ and $\ell(g(x), +1)$. Under the logistic error measure, we have:

$$\begin{aligned}
 \ell(g(x), +1) &= -\ln(g(x)), \\
 \ell(g(x), -1) &= -\ln(1 - g(x)).
 \end{aligned}$$

Since $g(x) = \text{sigmoid}(s)$, the error measure can be rewritten as:

$$\ell(s, y) = -\ln\left(\frac{1}{1 + e^{-ys}}\right)$$

First, we examine the convexity of the term $\ell(s, 1)$. The second derivative is:

$$\frac{d^2 \ell(s, +1)}{ds^2} = \frac{e^s}{(e^s + 1)^2} > 0 \quad (12)$$

which means that $\ell(s, +1)$ is convex with respect to s .

Next, we analyze the term $\hat{\ell}(s)$, which can be expressed as:

$$\begin{aligned}
 \ell(s, +1) - \ell(s, -1) &= -\ln\left(\frac{1}{1 + e^{-s}}\right) + \ln\left(\frac{1}{1 + e^s}\right) \\
 &= -\ln\left(\frac{1}{1 + e^{-s}}\right) + \ln\left(\frac{e^{-s}}{1 + e^{-s}}\right) \\
 &= -s.
 \end{aligned} \quad (13)$$

Since $\hat{\ell}(s)$ is linear with respect to s , it is also convex according to Lemma 2. Finally, as the sum of two convex functions remains convex, $R_{\text{pur1}}(g)$ is convex with respect to s .

Then, consider a matrix factorization model with \mathbf{U}_x and \mathbf{V}_x being the user embeddings and item embeddings indexed by x , respectively. The model is expressed as:

$$s(x) = \mathbf{U}_x^\top \mathbf{V}_x, g(x) = \text{sigmoid}(s(x)). \quad (14)$$

The score function s is convex in model parameters (\mathbf{U}_x and \mathbf{V}_x), which immediately follows from

$$\begin{aligned} \frac{d^2 s}{d\mathbf{U}_x^2} &= 0, \\ \frac{d^2 s}{d\mathbf{V}_x^2} &= 0. \end{aligned} \quad (15)$$

Therefore, R_{pur1} is convex in \mathbf{U}_x and \mathbf{V}_x , which immediately follows from the composition of convex functions is convex, R_{pur1} is convex in s , and s is convex in model parameters. \square

Lemma 3. Suppose g is a recommender model based on matrix factorization, with the output score s transformed by a sigmoid function, and the surrogate risk $R_{\text{pur1}}(g)$ is convex with respect to the score s . The surrogate risk $R_{\text{pur1}}(g)$ is minimized by the recommender g . Then, the minimization of the surrogate risk $R_{\text{pur1}}(g)$ converges within a finite number of iterations.

Proof. According to Theorem 4, the surrogate risk $R_{\text{pur1}}(g)$ is convex with respect to the score s . Given the recommender g as matrix factorization, $R_{\text{pur1}}(g)$ is convex in the model parameters, which immediately follows from the composition of convex functions. Therefore, the minimization for the convex $R_{\text{pur1}}(g)$ converges within a finite number of iterations. \square

B.2. Solution to Proximal Transport Problem

The PT problem in (7) is a linear programming problem and can be solved with traditional linear programming solvers (Gurobi Optimization, 2020; Hart et al., 2011). Nevertheless, such solutions often overlook the unique structure of optimal transport problems, thereby suffering from relatively large cost. In this section, we demonstrate a reformulation that aligns PT with the well-established Kantorovich problem, which permits solution to PT problem using specialized solvers for the Kantorovich problem, such as iterative Bregman projections (Benamou et al., 2015), Sinkhorn iterations (Altschuler et al., 2017; Dvurechensky et al., 2018), and the network-simplex method (Flamary et al., 2021), with superior convergence rate and efficiency (Bonneel et al., 2011; Ling and Okada, 2007; Pele and Werman, 2009).

To reformulate the PT problem (7) as a Kantorovich problem, we introduce a slack variable $\mathbf{s} \in \mathbb{R}_+^m$, and rephrase the inequality constraint $\boldsymbol{\pi}^\top \mathbf{1}_n \leq w * \mathbf{b}$ in (7) as an equality constraint:

$$\boldsymbol{\pi}^\top \mathbf{1}_n \leq w * \mathbf{b} \Rightarrow \boldsymbol{\pi}^\top \mathbf{1}_n + \mathbf{s} = w * \mathbf{b}, \quad (16)$$

which yields the equality constraints of \mathbf{b} as follow:

$$\begin{cases} \mathbf{1}_n^\top \boldsymbol{\pi} \mathbf{1}_m + \mathbf{s}^\top \mathbf{1}_m = (w * \mathbf{b})^\top \mathbf{1}_m \\ 1 + \mathbf{s}^\top \mathbf{1}_m = w * \|\mathbf{b}\|_1 \end{cases}. \quad (17)$$

Combining (17) and (7), we reformulate the PT problem (7) as:

$$\begin{aligned} \boldsymbol{\pi}^*(\alpha, \beta; w) &:= \arg \min_{\boldsymbol{\pi} \in \Pi(\alpha, \beta; w)} \langle \mathbf{C}, \boldsymbol{\pi} \rangle, \\ \Pi(\alpha, \beta; w) &:= \left\{ \begin{aligned} &\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} : \boldsymbol{\pi} \mathbf{1}_m = \mathbf{a}, \mathbf{1}_n^\top \boldsymbol{\pi} \mathbf{1}_m = 1, \\ &\mathbf{s} \in \mathbb{R}_+^m : \boldsymbol{\pi}^\top \mathbf{1}_n + \mathbf{s} = w * \mathbf{b}, \mathbf{s}^\top \mathbf{1}_m = w * \|\mathbf{b}\|_1 - 1 \end{aligned} \right\}. \end{aligned}$$

To substantiate the structural identity between the constraint set $\Pi(\alpha, \beta; w)$ in PT and that in the Kantorovich problem (Xu et al., 2020), we introduce augmented matrices and vectors as follows:

$$\tilde{\boldsymbol{\pi}} = \begin{bmatrix} \boldsymbol{\pi} & \mathbf{0}_{n \times 1} \\ \mathbf{s}^\top & \mathbf{0}_{1 \times 1} \end{bmatrix}, \quad \tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ w * \|\mathbf{b}\|_1 - 1 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{b}} = \begin{bmatrix} w * \mathbf{b} \\ \mathbf{0}_{1 \times 1} \end{bmatrix}.$$

On the basis, we can reformulate $\Pi(\alpha, \beta; w)$ as

$$\tilde{\Pi}(\tilde{\alpha}, \tilde{\beta}; w) := \left\{ \tilde{\pi} \in \mathbb{R}_+^{\tilde{n} \times \tilde{m}} : \tilde{\pi} \mathbf{1}_{\tilde{m}} = \tilde{\mathbf{a}}, \tilde{\pi}^\top \mathbf{1}_{\tilde{n}} = \tilde{\mathbf{b}} \right\} \quad (18)$$

which precisely mirrors the constraint set in the Kantorovich problem. Furthermore, we extend the transport cost matrix \mathbf{C} to $\tilde{\mathbf{C}}$ to align with this formulation:

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \xi \mathbf{1}_n \\ \xi \mathbf{1}_m & 2\xi + A \end{bmatrix}. \quad (19)$$

which incorporates a bounded scalar ξ and a constant $A > \max(C_{ij})$. [Xu et al. \(2020\)](#) shows that the transport cost in the Kantorovich problem can be rewritten as $\langle \tilde{\mathbf{C}}, \tilde{\pi} \rangle$. Therefore, the PT problem can be formulated as

$$\tilde{\pi}^*(\tilde{\alpha}, \tilde{\beta}; w) = \arg \min_{\tilde{\pi} \in \tilde{\Pi}(\tilde{\alpha}, \tilde{\beta}; w)} \langle \tilde{\mathbf{C}}, \tilde{\pi} \rangle. \quad (20)$$

In summary, through these augmentations and reformulations, encapsulated in (20), we seamlessly recast the PT problem as a Kantorovich problem. This equivalence enables us to leverage existing optimal transport solvers for solving the PT problem, thereby offering computational advantages and methodological coherence.