Joint User and Item Prototype Alignment for Cross-Platform Recommendation

Kehan Chen[1], Jiaxing Bai[2], Yishan Ji[3], Chen Lin[1,4](✉), and Ying Wang[2](✉)

[1] School of Informatics, Xiamen University, Xiamen, China
kehanchen@stu.xmu.edu.cn, chenlin@xmu.edu.cn
[2] School of Aerospace Engineering, Xiamen University, Xiamen, China
jiaxingbai@stu.xmu.edu.cn, wangying@xmu.edu.cn
[3] Maynooth International Engineering College, Fuzhou University, Fuzhou, China
832203109@fzu.edu.cn
[4] National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen, China

Abstract. Cross-platform recommendation (CPR) attracts increasing attention as a solution to data sparsity by leveraging non-overlapping user/item information from different platforms with similar services. Existing CPR methods face two main challenges: (1) the divergence in item IDs and content and (2) the divergence in user preference distributions across platforms. To tackle these challenges, we propose a CPR framework based on **U**ser and **I**tem **P**rototype **A**lignment (UIPA). For item representations, UIPA constructs a unified item semantic space via a large language model (LLM) and aligns item prototypes across platforms. For user representations, UIPA aligns homogeneous groups through user prototype alignment and enhances platform unique groups with a pseudo-prototype generation strategy. UIPA is a plug-and-play framework, and experimental results on three pairs of datasets demonstrate that UIPA significantly improves the performance of different backbones on both source and target platforms.

Keywords: Cross-platform recommendation · Prototype alignment · Pseudo-prototype.

1 Introduction

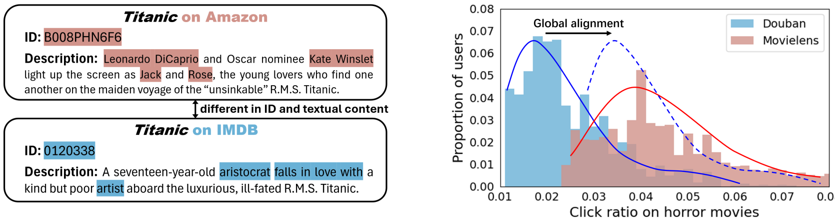
Recommendation Systems (RSs) are crucial for addressing information overload, achieving success in fields like e-commerce [14,17], advertising [1,6], and social networking [15,23]. However, data sparsity remains an issue due to limited feedback. To mitigate this, Cross-Platform Recommendation (CPR) and Cross-Domain Recommendation (CDR) leverage multi-platform or cross-domain information and are gaining importance in academia and industry [21].

This paper focuses on CPR, which differs from CDR in two key aspects: (1) CPR provides recommendations across platforms, while CDR operates within a

K. Chen and J. Bai contribute equally to this paper.

single platform; (2) CPR targets non-overlapping users or items, whereas CDR relies on overlapping sets [21]. Therefore, CPR has broader applications, enhancing niche platforms and reducing privacy risks in cross-platform data transfers.

In the literature, extensive research on CDR typically employs overlapping user/item information across domains as a bridge for cross-domain transfer [18,24]. However, commercial privacy policies and data sensitivity make it impractical to share user and item data across platforms, rendering CDR methods unsuitable for CPR tasks due to non-overlapping users and items. CPR studies typically transfer textual content [12,13,20] or align global user preference distributions [10,22,25]. However, these methods face limitations from cross-platform divergence and bias, leaving two challenges under-explored.



(a) The ID and content divergence of movie *Titanic* on Amazon and IMDB

(b) The divergent user preference distributions: the bin width of the X-axis is set to 0.002.

Fig. 1. The Divergence of users and items across platforms

Challenge 1: Divergence in item ID and textual content. Items exhibit platform-dependent characteristics, including IDs and content. **(1) ID divergence.** Different platforms assign unique IDs to identical items, such as Titanic on Amazon and IMDB (Figure 1(a)). Since ID embeddings dominate collaborative filtering RSs, this inconsistency impedes cross-platform feedback and effective ID embedding learning. **(2) Content divergence.** To address ID inconsistency, existing methods rely on item textual content for cross-platform transfer [4,12,13,20]. However, content often varies in language, style, and detail across platforms due to differing market focuses. For instance, Amazon emphasizes actors and characters, whereas IMDB focuses on the plot (Figure 1(a)). Such content divergence complicates creating a unified semantic space, causing negative transfer effects in content-based methods.

Challenge 2: Divergence in the distribution of user preferences. Figure 1(b) illustrates user click histograms for horror movies, revealing preference divergence between platforms. Current CPR methods [10,22,25] align distribution curves to reduce gaps (i.e., the blue and red lines in Figure 1(b)). However, Movielens users prefer horror movies more than Douban users, causing two alignment errors. **(1) Misalignment of homogeneous user groups.** Overlapping bars in Figure 1(b) indicate homogeneous groups, e.g., users with a 0.04 click ratio. These groups differ in size across platforms, with about 5% of Movielens users and only 1% of Douban users. Aligning distributions inflates or deflates these groups inaccurately. **(2) Misalignment of unique user groups.** Non-overlapping bars highlight platform-specific biases, such as Movielens users' strong preference for horror movies (i.e., click ratio $\geqslant 0.06$), absent on Douban. Aligning distributions overlooks these unique groups, limiting their benefit from cross-platform recommendations.

To address these challenges, we propose a general cross-platform recommendation framework based on **U**ser and **I**tem **P**rototype **A**lignment (UIPA). (1) From the item perspective, UIPA learns item prototypes (i.e., central representations of item clusters) and aligns them across platforms to bridge distinct ID spaces. Furthermore, it leverages a Large Language Model (LLM) to generate content, minimizing linguistic and semantic gaps. (2) From the user perspective, UIPA aligns user prototypes that capture the key characteristics of homogeneous user groups, preserving group sizes correctly. Additionally, it introduces pseudo prototypes to replace missing user groups on one platform, enabling effective information transfer to counterpart unique groups on the other platform.

UIPA is a plug-and-play framework. We apply UIPA to both matrix factorization (MF) model [9] and LightGCN [8] and conduct experiments on three pairs of real datasets. It achieves significant improvements, with average gains of 13.88% on MF and 8.01% on LightGCN, outperforming the latest CPR frameworks GWCDR [10] and SRTrans [12] by at least 4.01% across all datasets.

In summary, the contributions of this paper are as follows:

- We propose UIPA, a model-agnostic cross-platform recommendation framework that simultaneously enhances both source and target platforms, unlike prior CPR models focused solely on target platform improvement.
- UIPA introduces novel alignment methods, including item prototype alignment with content generation to mitigate item divergence and user prototype alignment with pseudo-prototype generation to address preference bias.
- Extensive experiments on three real-world dataset pairs validate UIPA 's effectiveness across two backbone recommendation models.

2 Related work

Cross-Domain Recommendation. Cross-domain recommendation (CDR) addresses data sparsity and cold-start issues by leveraging cross-domain knowledge to improve recommendation performance in the target domain [21]. Existing approaches extract cross-domain knowledge through overlapping information. For instance, HCCDR [18] models personalized preferences using overlapping users and their clustering. COAST [24] transfers user interest invariance based on partial user overlap. NATR [7] applies attention mechanisms to transfer item-side information between domains. M3Rec [2] builds cross-domain heterogeneous graphs with overlapping items to explore intra- and inter-domain item similarities. However, these methods depend on overlapping information, which is often unavailable in real-world applications, limiting their practical use.

Cross-Platform Recommendation. To address the limitations of CDR's reliance on overlapping information, Cross-Platform Recommendation (CPR) has gained significant attention. CPR models are primarily divided into two approaches: transferring global user interactions and transferring textual content.

Transferring global user interactions aims to align or aggregate user behaviors across platforms. For example, DA-CDR [22] employs dual adversarial learning to train an encoder for extracting shared user and item features, enabling alignment. ALCDR [25] utilizes a transportation matrix to aggregate user representations based on correlation coefficients. GWCDR [10] achieves cross-platform alignment by matching representation distributions through Gromov-Wasserstein learning. Transferring textual content utilizes textual information as a bridge for cross-platform transfer. For example, TDAR [20] and CFAA [13] align the latent spaces by using textual content as links. Furthermore, SR-Trans [12] and SCDGN [11] enhance the performance of cross-platform recommendation by clustering item textual content semantically and propagating knowledge on graphs.

3 Methodology

3.1 Overall Framework

This paper addresses the cross-platform recommendation (CPR) problem involving two platforms: the source platform $\langle S \rangle$ and the target platform $\langle T \rangle$. Each platform has a user set, item set, and interaction matrix denoted as $\mathcal{U}^i, \mathcal{V}^i, \mathcal{R}^i$, $(i \in \langle S \rangle, \langle T \rangle)$. For interactions, $R_{u^i,v^i} = 1$ if user u^i interacts with item v^i, otherwise $R_{u^i,v^i} = 0$. The goal is to propose a plug-and-play framework that simultaneously improves recommendations on both platforms. For simplicity, we assume the recommendation models on $\langle S \rangle$ and $\langle T \rangle$ are the same backbone model. Given users $u^i \in \mathcal{U}^i$ and items $v^i \in \mathcal{V}^i$, the backbones $\mathcal{M}^{\langle S \rangle}$ and $\mathcal{M}^{\langle T \rangle}$ learn L-dimensional ID embeddings, denoted as $\mathbf{u}^{\langle S \rangle}, \mathbf{u}^{\langle T \rangle}$ for users and $\mathbf{v}^{\langle S \rangle}, \mathbf{v}^{\langle T \rangle}$ for items.

The workflow of our proposed model, UIPA, is as follows: (1) **Pre-training Phase.** We independently train backbones $\mathcal{M}^{\langle S \rangle}$ and $\mathcal{M}^{\langle T \rangle}$ on their respective platforms until validation performance stabilizes. This step generates initial embeddings $\mathbf{u}^{\langle S \rangle}$, $\mathbf{u}^{\langle T \rangle}$, $\mathbf{v}^{\langle S \rangle}$, and $\mathbf{v}^{\langle T \rangle}$. (2) **Enhancement Phase.** These embeddings are input into UIPA, which enhances the representations and continues updating $\mathcal{M}^{\langle S \rangle}$ and $\mathcal{M}^{\langle T \rangle}$, producing final embeddings $\widetilde{\mathbf{u}}^i$ and $\widetilde{\mathbf{v}}^i$.

As illustrated in Figure 2, UIPA incorporates two core alignment components: (1) **Alignment from Item Perspective** (Section 3.2). A large language model (LLM) generates textual content, which is transformed into unified text embeddings to enhance item ID embeddings. Additionally, item prototypes are constructed and aligned across platforms to bridge items from different ID universes. (2) **Alignment from User Perspective** (Section 3.3). Homogeneous user groups are aligned via user prototype alignment to transfer preferences across platforms without falsely altering group sizes. For unique user groups on each platform, a pseudo-prototype generation strategy incorporates cross-platform information to enrich user representations.

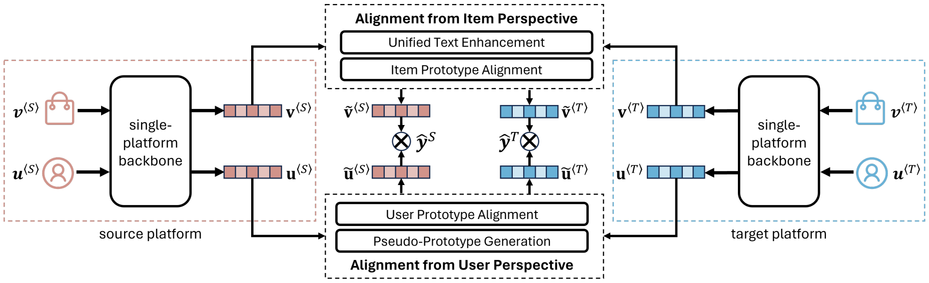

Alignment from Item Perspective
Unified Text Enhancement
Item Prototype Alignment
$\boldsymbol{v}^{\langle S\rangle}$
$\boldsymbol{u}^{\langle S\rangle}$
single-platform backbone
$\mathbf{v}^{\langle S\rangle}$
$\mathbf{u}^{\langle S\rangle}$
$\tilde{\mathbf{v}}^{\langle S\rangle}$
$\hat{\boldsymbol{y}}^{S}$
$\tilde{\mathbf{u}}^{\langle S\rangle}$
$\tilde{\mathbf{v}}^{\langle T\rangle}$
$\hat{\boldsymbol{y}}^{T}$
$\tilde{\mathbf{u}}^{\langle T\rangle}$
$\mathbf{v}^{\langle T\rangle}$
$\mathbf{u}^{\langle T\rangle}$
single-platform backbone
$\boldsymbol{v}^{\langle T\rangle}$
$\boldsymbol{u}^{\langle T\rangle}$
source platform
User Prototype Alignment
Pseudo-Prototype Generation
Alignment from User Perspective
target platform

Fig. 2. Overall framework of UIPA

3.2 Alignment from Item Perspective

Unified Text Enhancement. Different platforms provide rich textual content, such as item titles and descriptions, which have been used in previous studies [11,12,13,20] to supplement IDs. However, cross-platform divergences, such as language differences and market-specific emphasis, limit their direct use in CPR. Inspired by the success of LLMs, we employ the LLM Baichuan 2-13B [19] to generate unified textual content for CPR.

Specifically, item titles, as unique identifiers, are input into the LLM. We design task-specific demonstrations for different item types to ensure the generated text includes essential information. For example, for movies, the LLM generates text covering the plot, cast, directors, advantages, and disadvantages. Additionally, we adopt a one-shot prompt to ensure a uniform format. Through the above steps, the generated text embeds rich information from the LLM's knowledge base into a unified semantic space, addressing cross-platform divergence.

Secondly, to encode the LLM-generated text, we use pre-trained BERT [5], which leverages extensive knowledge from Wikipedia for semantic extraction and understanding. We input the LLM-generated text into BERT and obtain the [CLS] token embedding $\mathbf{v}^{i,CLS}, i \in \langle S\rangle, \langle T\rangle$, which represents the meaning of the entire paragraph. Then, we utilize a linear layer to resize $\mathbf{v}^{i,CLS}$ to the same dimensions as $\mathbf{v}$, and finally obtain textual embedding $\mathbf{v}^{i,text}, i \in \langle S\rangle, \langle T\rangle$:

$$\mathbf{v}^{i,text} = \mathbf{v}^{i,CLS}\mathbf{W}^{T} + \mathbf{b}, i \in \{\langle S\rangle, \langle T\rangle\}, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathbf{v}^{i,CLS}| \times L}$ is the weight and $\mathbf{b} \in \mathbb{R}^{L}$ is the additive bias. Notably, we can precompute embeddings $\mathbf{v}^{i,text}$ and store them in the cache for subsequent reuse, thus greatly reducing the time complexity during training.

The unified embedding $\mathbf{v}^{i,text}, i \in \langle S\rangle, \langle T\rangle$ enhances item ID embedding $\mathbf{v}^{i}, i \in \langle S\rangle, \langle T\rangle$ from backbones. The enhanced embedding $\hat{\mathbf{v}}^{i}, i \in \langle S\rangle, \langle T\rangle$ is calculated as:

$$\hat{\mathbf{v}}^{i} = (1-\alpha)\mathbf{v}^{i,text} + \alpha\mathbf{v}^{i}, i \in \{\langle S\rangle, \langle T\rangle\}, \tag{2}$$

where α is a hyper-parameter balancing the weights of the two embeddings.

Item Prototype Alignment. On different platforms, the same item often has different IDs, making it infeasible to map items by their IDs directly. Previous studies [2,16] suggest that items with similar attributes attract the same users or user groups. Therefore, it is more effective to connect clusters of items sharing similar attributes.

Defining prototypes. Intuitively, a prototype represents the attributes of an item cluster. Formally, the prototype of cluster j is an embedding vector $\mathbf{cv}_j \in \mathbb{R}^L$, symbolizing a typical characteristic in item cluster j. For platforms $\langle S \rangle$ and $\langle T \rangle$, we both create N prototypes, defined as $\mathbf{cv}_j^{\langle S \rangle}, \mathbf{cv}_j^{\langle T \rangle}, j \leq N$.

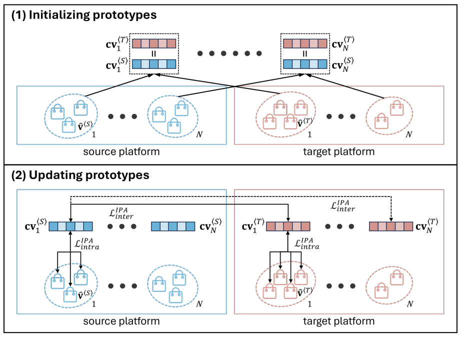

(1) Initializing prototypes
$\mathbf{cv}_1^{\langle T \rangle}$
$\mathbf{cv}_1^{\langle S \rangle}$
$\mathbf{cv}_N^{\langle T \rangle}$
$\mathbf{cv}_N^{\langle S \rangle}$
$\hat{\mathbf{v}}^{\langle S \rangle}$
1
N
$\hat{\mathbf{v}}^{\langle T \rangle}$
1
N
source platform
target platform
(2) Updating prototypes
$\mathcal{L}_{inter}^{IPA}$
$\mathcal{L}_{inter}^{IPA}$
$\mathbf{cv}_1^{\langle S \rangle}$
$\mathbf{cv}_N^{\langle S \rangle}$
$\mathbf{cv}_1^{\langle T \rangle}$
$\mathbf{cv}_N^{\langle T \rangle}$
$\mathcal{L}_{intra}^{IPA}$
$\mathcal{L}_{intra}^{IPA}$
$\hat{\mathbf{v}}^{\langle S \rangle}$
1
N
$\hat{\mathbf{v}}^{\langle T \rangle}$
1
N
source platform
target platform

Fig. 3. Illustration of item prototype alignment

Initializing prototypes. Prototypes are initialized by co-clustering items from $\langle S \rangle$ and $\langle T \rangle$. Item embeddings from the unified text enhancement module, $\hat{\mathbf{v}}^i, i \in \langle S \rangle, \langle T \rangle$, are combined into a set and clustered using the KMedoids algorithm. These cluster centers serve as the initial prototypes for both $\langle S \rangle$ and $\langle T \rangle$, ensuring that the starting points for alignment are consistent across platforms.

Updating prototypes. During the update, we maintain two replicas, $\mathbf{cv}_j^i, i \in \langle S \rangle, \langle T \rangle$, for the same cluster j, referred to as *counterpart prototypes*. Prototypes are updated through intra-platform and inter-platform alignment. In intra-platform alignment, prototypes are updated by minimizing intra-cluster distances. For each item embedding $\mathbf{v}^i, i \in \langle S \rangle, \langle T \rangle$, the closest prototype $\mathbf{cv}_j^i$ is selected based on L2 distance. Each item embedding is adjusted closer to its prototype, ensuring cluster consistency. The intra-platform alignment loss is defined as:

$$\mathcal{L}_{intra}^{IPA} = -\sum_{v^i \in \mathcal{V}^i} (||sg(\mathbf{cv}_j^i) - \hat{\mathbf{v}}^i||^2 + ||\mathbf{cv}_j^i - sg(\hat{\mathbf{v}}^i)||^2), i \in \{\langle S \rangle, \langle T \rangle\}, \tag{3}$$

where $sg(\cdot)$ is the stop-gradient operator, which is defined as the identity during forward computation, and its partial derivative is zero.

In inter-platform alignment, we align counterpart prototypes closer and separate prototypes of different clusters to enable information transfer between source and target platforms while preserving platform-specific dependencies. The inter-platform alignment loss is expressed as:

$$\begin{aligned} \mathcal{L}_{inter}^{IPA} = &-\frac{1}{N}\sum_{j=1}^{N} \log \frac{\exp(sim(\mathbf{cv}_j^{\langle S \rangle}, sg(\mathbf{cv}_j^{\langle T \rangle}))/\tau)}{\sum_{j'=1}^{N} \exp(sim(\mathbf{cv}_j^{\langle S \rangle}, sg(\mathbf{cv}_{j'}^{\langle T \rangle}))/\tau)} \\ &-\frac{1}{N}\sum_{j=1}^{N} \log \frac{\exp(sim(\mathbf{cv}_j^{\langle T \rangle}, sg(\mathbf{cv}_j^{\langle S \rangle}))/\tau)}{\sum_{j'=1}^{N} \exp(sim(\mathbf{cv}_j^{\langle T \rangle}, sg(\mathbf{cv}_{j'}^{\langle S \rangle}))/\tau)}, \end{aligned} \tag{4}$$

where $\mathbf{cv}_j^{\langle S\rangle}$ and $\mathbf{cv}_j^{\langle T\rangle}$ denote the embeddings of corresponding item prototypes from the source and target platforms, τ is the temperature parameter, $sim(\cdot,\cdot)$ represents cosine similarity, and $sg(\cdot)$ is the stop-gradient operator.

The final item prototype alignment loss is formulated by:

$$\mathcal{L}^{IPA} = \mathcal{L}^{IPA}_{intra} + \mathcal{L}^{IPA}_{inter}. \tag{5}$$

3.3 Alignment from User Perspective

User Prototype Alignment. Similar to an item prototype, a user prototype represents a specific user type characterized by similar user preferences. We adopt a similar strategy for user prototype alignment. First, we initialize the user prototypes $\mathbf{cu}_j^{\langle S\rangle}, \mathbf{cu}_j^{\langle T\rangle}, j \leq N$ by the clustering center of users. Then, we update prototypes and compute $\mathcal{L}^{UPA}_{intra}$ and $\mathcal{L}^{UPA}_{inter}$ as follows:

$$\mathcal{L}^{UPA}_{intra} = -\sum_{u^i \in \mathcal{U}^i} (||sg(\mathbf{cu}_j^i) - \mathbf{u}^i||^2 + ||\mathbf{cu}_j^i - sg(\mathbf{u}^i)||^2), i \in \{\langle S\rangle, \langle T\rangle\}, \tag{6}$$

$$\begin{aligned}\mathcal{L}^{UPA}_{inter} = &-\frac{1}{N}\sum_{j=1}^{N}\log\frac{\exp(sim(\mathbf{cu}_j^{\langle S\rangle}, sg(\mathbf{cu}_j^{\langle T\rangle}))/\tau)}{\sum_{j'=1}^{N}\exp(sim(\mathbf{cu}_j^{\langle S\rangle}, sg(\mathbf{cu}_{j'}^{\langle T\rangle}))/\tau)} \\ &-\frac{1}{N}\sum_{j=1}^{N}\log\frac{\exp(sim(\mathbf{cu}_j^{\langle T\rangle}, sg(\mathbf{cu}_j^{\langle S\rangle}))/\tau)}{\sum_{j'=1}^{N}\exp(sim(\mathbf{cu}_j^{\langle T\rangle}, sg(\mathbf{cu}_{j'}^{\langle S\rangle}))/\tau)},\end{aligned} \tag{7}$$

where $\mathbf{cu}_j^i, i \in \{\langle S\rangle, \langle T\rangle\}$ is the corresponding prototype to $\mathbf{u}^i, i \in \{\langle S\rangle, \langle T\rangle\}$. $\mathbf{cu}_j^{\langle S\rangle}$ and $\mathbf{cu}_j^{\langle T\rangle}$ are the embeddings of a pair of counterpart item prototypes on the source and target platforms. Next, the user prototype alignment loss is formulated by:

$$\mathcal{L}^{UPA} = \mathcal{L}^{UPA}_{intra} + \mathcal{L}^{UPA}_{inter}. \tag{8}$$

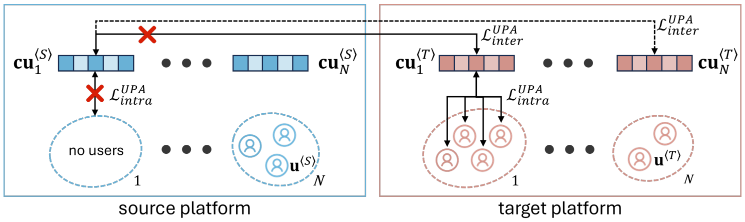

$\mathbf{cu}_1^{\langle S\rangle}$
$\mathbf{cu}_N^{\langle S\rangle}$
$\mathcal{L}^{UPA}_{intra}$
no users
1
$\mathbf{u}^{\langle S\rangle}$
N
source platform
$\mathcal{L}^{UPA}_{inter}$
$\mathcal{L}^{UPA}_{inter}$
$\mathbf{cu}_1^{\langle T\rangle}$
$\mathbf{cu}_N^{\langle T\rangle}$
$\mathcal{L}^{UPA}_{intra}$
1
$\mathbf{u}^{\langle T\rangle}$
N
target platform

Fig. 4. An illustrative example that the platform $\langle T\rangle$'s unique prototype $\mathbf{cu}_1^{\langle T\rangle}$ can not fully exploit the information in platform $\langle S\rangle$

Pseudo-Prototype Generation. Due to the divergence in user preference distribution, certain unique user groups exist on one platform but not another. For instance, Figure 4 illustrates a unique user group $\mathbf{cu}_1^i, i \in \langle S\rangle, \langle T\rangle$ present on platform $\langle T\rangle$ but absent on $\langle S\rangle$. During the computation of $||\mathbf{cu}_1^{\langle S\rangle} - sg(\mathbf{u}^{\langle S\rangle})||^2$ in Eq. 6, the lack of corresponding $\mathbf{u}^{\langle S\rangle}$ prevents $\mathbf{cu}_1^{\langle S\rangle}$ from being updated with $\langle S \rangle$ user data. Similarly, in calculating $sim(\mathbf{cu}_1^{\langle T \rangle}, sg(\mathbf{cu}_1^{\langle S \rangle}))$ in Eq. 7, the missing user information for $\mathbf{cu}_1^{\langle S \rangle}$ hinders the transfer of user data from $\langle S \rangle$ to $\langle T \rangle$. This example demonstrates that user information on $\langle S \rangle$ is not fully utilized to optimize $\langle T \rangle$.

To address this problem, we propose generating a pseudo-prototype $\mathbf{cp}_1^{\langle S \rangle}$ to replace the real prototype $\mathbf{cu}_1^{\langle S \rangle}$ by leveraging users on $\langle S \rangle$. To effectively link the platforms, the users chosen to generate $\mathbf{cp}_1^{\langle S \rangle}$ on $\langle S \rangle$ should be similar to those on $\langle T \rangle$, establishing a meaningful connection. However, as the pseudo-prototype is derived from similar users, it may lack precision. To ensure positive contributions to the transfer process, $\mathbf{cu}_1^{\langle S \rangle}$ is replaced with $\mathbf{cp}_1^{\langle S \rangle}$ only if the BPR loss based on the pseudo-prototype decreases.

Algorithm 1: Pseudo-Prototype Generation

```
Input: prototypes $\mathbf{cu}_k^i, k \le N, i \in \{\langle S \rangle, \langle T \rangle\}$,
user embedding sets $\mathbf{U}^i, i \in \{\langle S \rangle, \langle T \rangle\}$
Output: prototypes $\mathbf{cu}_k^i, k \le N, i \in \{\langle S \rangle, \langle T \rangle\}$
$\mathbf{P}_k^i \leftarrow \varnothing$, $\mathbf{Q}_k^i \leftarrow \varnothing$, $k \le N, i \in \{\langle S \rangle, \langle T \rangle\}$;
foreach $\mathbf{u}^i \in \mathbf{U}^i, i \in \{\langle S \rangle, \langle T \rangle\}$ do
    $k \leftarrow \arg\min_k(\sqrt{\sum_{l=1}^{L} \|\mathbf{cu}_{kl}^i - \mathbf{u}_l^i\|_2^2})$;
    $\mathbf{P}_k^i \leftarrow \mathbf{P}_k^i \cup \{\mathbf{u}^i\}$;
end
foreach $\mathbf{cu}_k^i, k \le N, i \in \{\langle S \rangle, \langle T \rangle\}$ do
    if $|\mathbf{P}_k^i| = 0$ then
        foreach $\mathbf{u}^j \in \mathbf{P}_k^j, j \ne i$ do
            $\mathbf{u}^i \leftarrow \arg\min_{\mathbf{u}^i}(\sqrt{\sum_{l=1}^{L} \|\mathbf{u}_l^i - \mathbf{u}_l^j\|_2^2})$;
            $\mathbf{Q}_k^i \leftarrow \mathbf{Q}_k^i \cup \{\mathbf{u}^i\}$;
        end
        $\mathbf{cp}_k^i \leftarrow \frac{1}{|\mathbf{Q}_k^i|} \sum_{\mathbf{u}^i \in \mathbf{Q}_k^i} (\mathbf{u}^i)$;
        $\mathcal{L}_{old}^{BPR} \leftarrow \sum_{\mathbf{u}^j \in \mathbf{P}_k^j} \ln \sigma(\widetilde{\mathbf{u}}^j \widetilde{\mathbf{v}}^j - \widetilde{\mathbf{u}}^j \widetilde{\mathbf{n}}^j)$;
        Update $\mathbf{P}_k^j$ by Eq. 6 - 7 and obtain $\mathbf{P}'^j_k$;
        $\mathcal{L}_{new}^{BPR} \leftarrow \sum_{\mathbf{u}'^{\mathbf{j}} \in \mathbf{P}'^j_k} \ln \sigma(\widetilde{\mathbf{u}}'^j \widetilde{\mathbf{v}}'^j - \widetilde{\mathbf{u}}'^j \widetilde{\mathbf{n}}'^j)$;
        if $\mathcal{L}_{new}^{BPR} < \mathcal{L}_{old}^{BPR}$ then
            $\mathbf{cu}_k^i \leftarrow \mathbf{cp}_k^i$;
        end
    end
end
```

We generate the pseudo-prototype $\mathbf{cp}_k^i$ and decide whether to replace $\mathbf{cu}_k^i$ based on Algorithm 1, following four steps: **Step 1:** For each user embedding $\mathbf{u}^i$, identify the closest prototype $\mathbf{cu}_k^i$ in L2 distance and add the user embedding to its corresponding set $\mathbf{P}_k^i$ (Lines 1-5). **Step 2:** For any prototype $\mathbf{cu}_k^i$ with an empty set $\mathbf{P}_k^i$, execute the subsequent steps (Lines 6-7). **Step 3:** For user embeddings $\mathbf{u}^j \in \mathbf{P}_k^j$ on platform j (i.e., the other platform relative to i), find their closest counterparts $\mathbf{u}^i$ on platform i. Then, average these embeddings to create $\mathbf{cp}_k^i$ (Lines 8-12). **Step 4:** Determine whether to replace $\mathbf{cu}_k^i$ with $\mathbf{cp}_k^i$ by evaluating the change in BPR loss (Lines 13-18).

3.4 Optimization

Through the enhancement of the prototypes in the item prototype alignment module and the user prototype alignment module, we obtain the final item embedding $\widetilde{\mathbf{v}}^i, i \in \{\langle S\rangle, \langle T\rangle\}$ and user embedding $\widetilde{\mathbf{u}}^i, i \in \{\langle S\rangle, \langle T\rangle\}$:

$$\widetilde{\mathbf{v}}^i = (1-\beta)\hat{\mathbf{v}}^i + \beta \mathbf{cv}_j^i, i \in \{\langle S\rangle, \langle T\rangle\}, j \leq N, \tag{9}$$

$$\widetilde{\mathbf{u}}^i = (1-\beta)\mathbf{u}^i + \beta \mathbf{cu}_j^i, i \in \{\langle S\rangle, \langle T\rangle\}, j \leq N, \tag{10}$$

where β is the hyper-parameter. $\hat{\mathbf{v}}^i$ is the enhanced item embedding obtained by Eq. 2. $\mathbf{cv}_j^i$ and $\mathbf{cu}_j^i$ are the corresponding prototypes to $\hat{\mathbf{v}}^i$ and $\mathbf{u}^i$, respectively.

The original loss function for single platform recommendation systems is usually the BPR Loss $\mathcal{L}^{BPR}$,

$$\mathcal{L}^{BPR} = -\sum_{(u^i,v^i,n^i)\in\mathcal{B}^i} \ln \sigma(\widetilde{\mathbf{u}}^i\widetilde{\mathbf{v}}^i - \widetilde{\mathbf{u}}^i\widetilde{\mathbf{n}}^i), i \in \{\langle S\rangle, \langle T\rangle\}, \tag{11}$$

where $(u^i, v^i, n^i) \in \mathcal{B}^i$ represents a sampled triple, with user u^i interacting with item v^i (i.e., $r_{u^i,v^i} = 1$), and item n^i being a negative sample (i.e., $r_{u^i,n^i} = 0$), for $i \in \langle S\rangle, \langle T\rangle$. σ denotes the sigmoid function.

Finally, the overall loss includes the BPR loss $\mathcal{L}^{BPR}$, the item prototype alignment loss $\mathcal{L}^{IPA}$ and user prototype alignment loss $\mathcal{L}^{UPA}$:

$$\mathcal{L} = \mathcal{L}^{BPR} + \lambda(\mathcal{L}^{IPA} + \mathcal{L}^{UPA}), \tag{12}$$

where λ is the weight coefficient that balances loss terms.

4 Experiment

In this section, We focus on the following research questions (RQs):

- **RQ1:** Can UIPA improve the performance of different RS backbones?
- **RQ2:** How does each component affect UIPA's performance?
- **RQ3:** How sensitive is UIPA to its hyper-parameters?
- **RQ4:** How is the performance of UIPA on the sparser datasets?

4.1 Experimental Setup

Datasets. We adopt three pairs of datasets to evaluate the effectiveness of UIPA, which are *MovieLens-1m* and *Douban Movies*, *MovieTwettings* and *Douban Movies*, *Douban Books* and *BookCrossing*. These three pairs of datasets are publicly available and widely used by existing studies [13,25]. At the same time, these datasets reflect different characteristics of users in different countries, which is difficult for CPR.

Table 1. Statistics of datasets

Dataset	#Users	#Items	#Interactions	Sparsity	Language
MovieLens-1m	5,638	1,916	220,312	97.96%	English
MovieTwettings	10,142	2,538	149,495	99.42%	English
Douban Movies	2,502	4,462	231,303	97.93%	Chinese
BookCrossing	4,154	3,992	59,451	99.64%	English
Douban Books	1,409	2,064	42,078	98.55%	Chinese

Inspired by prior studies [10,13], we binarize the ratings to 0 and 1 and apply 5-core and 10-core settings for the users and items in each dataset. Specifically, for denser movie datasets, ratings $\geq$ 5 are set as 1, and all others as 0. For sparser book datasets, ratings ≥ 4 are set as 1, and others as 0. Users with at least 5 interactions and items with at least 10 interactions are retained, ensuring a balance between capturing user preferences and maintaining sparsity. Dataset statistics are presented in Table 1, with an 8:1:1 split for training, validation, and testing.

Evaluation Metrics. We use standard top-N recommendation metrics, including Hit Rate@k (HR@k), NDCG@k, Recall@k, and Precision@k, for evaluation. Specifically, we choose $k = 10$ for the evaluation metrics in this paper.

Implementation Details. Our framework is implemented in Pytorch, with datasets and codes publicly available online[1]. User and item embeddings are set to size 64, and Adam optimizer is used for all methods. Backbones across frameworks share the same architecture.

Hyper-parameters for UIPA are tuned using grid search, with early stopping based on R@10 on the validation set to ensure efficiency and avoid overfitting. Learning rates are set to 0.005 for movie datasets and 0.01 for book datasets in MF- and LightGCN-based frameworks. As for all baselines, we utilize the optimal parameters mentioned in their respective papers.

Backbones. The proposed UIPA is a plug-and-play CPR framework and can be applied to collaborative filtering backbones. To facilitate comparison, we follow existing CPR studies [10,12] and use two standard backbones: (1) Matrix Factorization (MF) [9], which decomposes the user-item interaction matrix to user and item factors, and (2) LightGCN [8], which aggregates user-item interaction by a simplified graph convolutional network.

Baselines. We compare UIPA with state-of-the-art plug-and-play CPR frameworks: (1) SRTrans [12], which clusters items semantically and extracts inter-platform relational knowledge, and (2) GWCDR [10], which aligns representation distributions between platforms using Gromov-Wasserstein distance. Additionally, we evaluate against the latest CPR models: (1) SCDGN [11], which builds a cross-platform bipartite graph by clustering items semantically and incorporating user interactions, and (2) UniCDR [3], which transfers shared platform information using masking mechanisms and aggregators.

4.2 Main Results (RQ1)

We conduct experiments on three real-world dataset pairs. For UIPA, we first train the backbones separately on the two platforms until performance stabilizes. Then, UIPA is applied to update the backbones on both source and target platforms simultaneously. Thus, for each dataset pair, UIPA is applied once, and

[1] https://github.com/XMUDM/UIPA

Table 2. Overall performance of cross-platform recommendation. The best improvements are highlighted in bold. %Improv. represents the percentage of improvement of UIPA to the best result in corresponding competitors.

Method		*MovieLens-1m→Douban Movies*				*Douban Movies→MovieLens-1m*			
		HR@10	NDCG@10	Recall@10	Precision@10	HR@10	NDCG@10	Recall@10	Precision@10
CPR models	SCDGN	0.0297	0.0394	0.0394	0.0288	0.1149	0.1064	0.1442	0.0499
	UniCDR	0.0317	0.0150	0.0471	0.0307	0.1024	0.0489	0.1450	0.0445
CPR plug-and-play frameworks	MF	0.0312	0.0402	0.0403	0.0303	0.1340	0.1253	0.1673	0.0582
	+ SRTrans	0.0194	0.0263	0.0275	0.0188	0.1145	0.1014	0.1375	0.0498
	+ GWCDR	0.0324	0.0409	0.0372	0.0314	0.1315	0.1181	0.1561	0.0571
	+ **UIPA**	**0.0374**	**0.0508**	**0.0485**	**0.0362**	**0.1557**	**0.1453**	**0.1960**	**0.0677**
	%Improv.	15.14%	24.26%	20.39%	15.14%	16.17%	15.95%	17.15%	16.17%
	LightGCN	0.0463	0.0644	0.0621	0.0448	0.1729	0.1630	0.2113	0.0751
	+ SRTrans	0.0481	0.0655	0.0602	0.0466	0.1646	0.1528	0.2009	0.0716
	+ GWCDR	0.0452	0.0610	0.0568	0.0437	0.1730	0.1621	0.2125	0.0752
	+ **UIPA**	**0.0510**	**0.0699**	**0.0670**	**0.0494**	**0.1801**	**0.1694**	**0.2233**	**0.0783**
	%Improv.	6.01%	6.71%	7.76%	6.01%	4.10%	3.93%	5.07%	4.10%

Method		*MovieTwettings→Douban Movies*				*Douban Movies→MovieTwettings*			
		HR@10	NDCG@10	Recall@10	Precision@10	HR@10	NDCG@10	Recall@10	Precision@10
CPR models	SCDGN	0.0302	0.0382	0.0398	0.0293	0.1130	0.0773	0.1328	0.0213
	UniCDR	0.0310	0.0147	0.0477	0.0300	0.1166	0.0568	0.1399	0.0220
CPR plug-and-play frameworks	MF	0.0312	0.0402	0.0403	0.0303	0.1318	0.0884	0.1493	0.0249
	+ SRTrans	0.0196	0.0251	0.0261	0.0190	0.0657	0.0428	0.0705	0.0124
	+ GWCDR	0.0316	0.0399	0.0365	0.0306	0.1044	0.0673	0.1138	0.0197
	+ **UIPA**	**0.0365**	**0.0493**	**0.0473**	**0.0353**	**0.1395**	**0.0955**	**0.1585**	**0.0263**
	%Improv.	15.40%	22.66%	17.47%	15.40%	5.79%	8.03%	6.14%	5.79%
	LightGCN	0.0463	0.0644	0.0621	0.0448	0.1487	0.1001	0.1655	0.0281
	+ SRTrans	0.0483	0.0650	0.0613	0.0467	0.1385	0.0933	0.1572	0.0261
	+ GWCDR	0.0455	0.0614	0.0574	0.0441	0.1492	0.1025	0.1688	0.0281
	+ **UIPA**	**0.0519**	**0.0693**	**0.0668**	**0.0502**	**0.1603**	**0.1076**	**0.1797**	**0.0302**
	%Improv.	7.53%	6.57%	7.58%	7.53%	7.46%	5.00%	6.47%	7.46%

Method		*BookCrossing→Douban Books*				*Douban Books→BookCrossing*			
		HR@10	NDCG@10	Recall@10	Precision@10	HR@10	NDCG@10	Recall@10	Precision@10
CPR models	SCDGN	0.0532	0.0435	0.0654	0.0182	0.0413	0.0279	0.0513	0.0076
	UniCDR	0.0459	0.0228	0.0622	0.0157	0.0481	0.0243	0.0580	0.0089
CPR plug-and-play frameworks	MF	0.0563	0.0459	0.0663	0.0192	0.0762	0.0560	0.0874	0.0141
	+ SRTrans	0.0345	0.0285	0.0380	0.0118	0.0394	0.0261	0.0455	0.0073
	+ GWCDR	0.0557	0.0423	0.0558	0.0190	0.0609	0.0449	0.0678	0.0112
	+ **UIPA**	**0.0648**	**0.0546**	**0.0779**	**0.0221**	**0.0794**	**0.0571**	**0.0918**	**0.0147**
	%Improv.	15.13%	19.05%	17.49%	15.13%	4.28%	1.96%	4.95%	4.28%
	LightGCN	0.0711	0.0591	0.0809	0.0243	0.0962	0.0681	0.1073	0.0178
	+ SRTrans	0.0729	0.0572	0.0783	0.0249	0.0791	0.0560	0.0907	0.0146
	+ GWCDR	0.0731	0.0581	0.0818	0.0250	0.0914	0.0660	0.1023	0.0168
	+ **UIPA**	**0.0804**	**0.0675**	**0.0940**	**0.0275**	**0.0990**	**0.0687**	**0.1108**	**0.0183**
	%Improv.	9.94%	14.26%	15.03%	9.94%	2.85%	0.89%	3.30%	2.85%

results are reported for both source and target datasets. In contrast, competitors only improve the target dataset by transferring from the source dataset, requiring two separate applications for each dataset pair. From the performance results illustrated in Table 2, we have the following observations.

(1) **UIPA significantly improves both MF and LightGCN backbones.** For the MF backbone, UIPA achieves an average improvement of 12.95% in HR@10, 15.67% in NDCG@10, 13.93% in Recall@10, and 12.95% in Precision@10. For the LightGCN backbone, UIPA achieves an average improvement of 8.38%, 7.11%, 8.19%, and 8.38%. Despite LightGCN being a robust model with superior performance over MF, UIPA's improvements on it demonstrate its ability to transfer cross-platform information and enhance single-platform backbones effectively.

(2) **UIPA is simultaneously effective for both the source and target platforms on the three pairs of datasets.** For the three pairs of datasets, UIPA achieves an average improvement of 8.85%-12.52% in HR@10, 9.04%-13.71% in NDCG@10, 9.93%-12.74% in Recall@10, and 8.85%-12.52% in Precision@10. Importantly, unlike other CPR methods focusing solely on the target platform, UIPA improves performance on both platforms. This dual-platform effectiveness not only enhances performance but also reduces training time.

(3) **UIPA consistently outperforms all baselines.** Across all datasets, UIPA achieves stable and significant improvements, surpassing the best-performing baseline, GWCDR, by 12.42%, 15.30%, 17.91%, and 12.43% in HR@10, NDCG@10, Recall@10, and Precision@10, respectively. Other CPR methods exhibit performance instability due to divergence in item space and user preferences, leading to negative transfer. By mitigating this divergence, UIPA achieves both stability and superior performance.

In summary, UIPA effectively addresses cross-platform divergence, facilitating robust knowledge transfer and achieving the best performance.

4.3 Ablation Study (RQ2)

To demonstrate the effectiveness of each component in UIPA, we conduct two ablation studies on the *Douban Movies* and *MovieLens-1m* datasets: module ablation and content generation ablation.

Table 3. Performance of different components of UIPA

Method		*MovieLens-1m→Douban Movies*				*Douban Movies→MovieLens-1m*			
		HR@10	NDCG@10	Recall@10	Precision@10	HR@10	NDCG@10	Recall@10	Precision@10
MF	Vanilla	0.0312	0.0402	0.0403	0.0303	0.1340	0.1253	0.1673	0.0582
	+ UTE	0.0357	0.0467	0.0473	0.0345	0.1460	0.1356	0.1822	0.0635
	+ UTE & IPA	0.0359	0.0484	0.0476	0.0348	0.1485	0.1379	0.1853	0.0645
	+ UTE & IPA & UPA	0.0365	0.0502	0.0482	0.0353	0.1535	0.1441	0.1939	0.0667
	+ **UTE & IPA & UPA & PPG**	**0.0374**	**0.0508**	**0.0485**	**0.0362**	**0.1557**	**0.1453**	**0.1960**	**0.0677**
LightGCN	Vanilla	0.0463	0.0644	0.0621	0.0448	0.1729	0.1630	0.2113	0.0751
	+ UTE	0.0483	0.0655	0.0636	0.0468	0.1768	0.1677	0.2182	0.0768
	+ UTE & IPA	0.0492	0.0669	0.0646	0.0476	0.1782	0.1680	0.2206	0.0774
	+ UTE & IPA & UPA	0.0506	0.0690	0.0659	0.0490	0.1788	0.1684	0.2207	0.0777
	+ **UTE & IPA & UPA & PPG**	**0.0510**	**0.0699**	**0.0670**	**0.0494**	**0.1801**	**0.1694**	**0.2233**	**0.0783**

Module ablation. This study evaluates the contribution of each module in UIPA. We sequentially integrate four modules into the backbones and compare their performance: (1) Unified Text Enhancement (UTE), which leverages LLM and BERT to generate unified text embeddings, (2) Item Prototype Alignment (IPA), which aligns items at the cluster level, (3) User Prototype Alignment (UPA), which extracts user preferences and aligns homogeneous user groups, (4) Pseudo-Prototype Generation (PPG), which transfers cross-platform information for unique user groups. From the results in Table 3, we have the following observations.

(1) **Each module is effective for performance improvement.** Adding the UTE, IPA, UPA, and PPG modules sequentially to the backbones leads to average improvements of 7.52%, 1.41%, 2.25%, and 1.20% across all met-

rics, respectively. This highlights that each module in UIPA effectively transfers enhancement information, positively impacting performance.

(2) **The Unified Text Enhancement (UTE) module provides the most improvement.** Among the modules, UTE achieves the highest improvement, with an average of 7.52%. The possible reason is that the LLM-generated text eliminates cross-platform text differences and supplements items with rich semantic information, allowing the model to better understand item characteristics and relevance. While UTE provides the most significant improvement, the other modules contribute to further improvement, showing that UIPA effectively leverages cross-platform enhancements.

Table 4. Performance of UIPA with different types of text

Method		*MovieLens-1m→Douban Movies*				*Douban Movies→MovieLens-1m*			
		HR@10	NDCG@10	Recall@10	Precision@10	HR@10	NDCG@10	Recall@10	Precision@10
MF	+ $\mathrm{UIPA}_{\mathrm{ID}}$	0.0329	0.0435	0.0438	0.0318	0.1493	0.1409	0.1885	0.0649
	+ $\mathrm{UIPA}_{\mathrm{title}}$	0.0345	0.0450	0.0430	0.0334	0.1526	0.1429	0.1898	0.0663
	+ $\mathrm{UIPA}_{\mathrm{plot}}$	0.0343	0.0457	0.0433	0.0333	0.1524	0.1432	0.1920	0.0662
	+ $\mathbf{UIPA}_{\mathbf{LLM}}$	**0.0374**	**0.0508**	**0.0485**	**0.0362**	**0.1557**	**0.1453**	**0.1960**	**0.0677**
LightGCN	+ $\mathrm{UIPA}_{\mathrm{ID}}$	0.0479	0.0650	0.0623	0.0464	0.1750	0.1661	0.2173	0.0761
	+ $\mathrm{UIPA}_{\mathrm{title}}$	0.0492	0.0689	0.0650	0.0476	0.1771	0.1671	0.2183	0.0769
	+ $\mathrm{UIPA}_{\mathrm{plot}}$	0.0503	0.0681	0.0667	0.0487	0.1776	0.1691	0.2210	0.0772
	+ $\mathbf{UIPA}_{\mathbf{LLM}}$	**0.0510**	**0.0699**	**0.0670**	**0.0494**	**0.1801**	**0.1694**	**0.2233**	**0.0783**

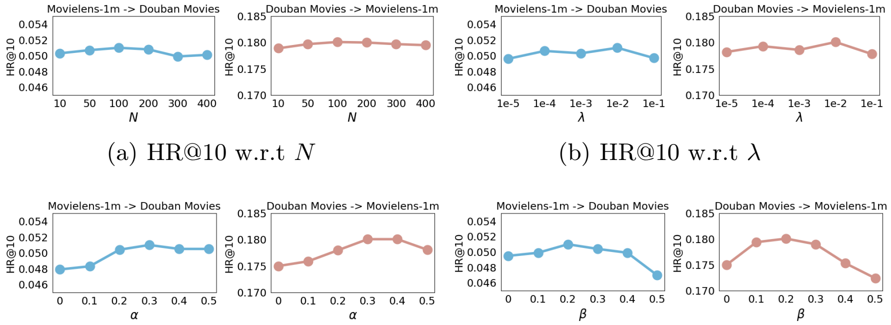
Content generation ablation. This study explores the superiority of the textual content generated by LLM over other types. We incorporate UIPA with four types of content: (1) $\mathrm{UIPA}_{\mathrm{ID}}$ which is UIPA without textual content, (2) $\mathrm{UIPA}_{\mathrm{title}}$ which is UIPA with movie titles, (3) $\mathrm{UIPA}_{\mathrm{title}}$ which is UIPA with movie plots, (4) $\mathrm{UIPA}_{\mathrm{LLM}}$ which is UIPA with text generated by LLM.

From the results in Table 4, we observe that **the textual content generated by LLM outperforms other types.** Compared to $\mathrm{UIPA}_{\mathrm{ID}}$, UIPA with any textual content achieves better performance, demonstrating the benefit of semantic information. However, longer content does not always lead to more remarkable improvement: while plots are longer than titles, $\mathrm{UIPA}_{\mathrm{plot}}$ does not consistently outperform $\mathrm{UIPA}_{\mathrm{title}}$ due to potential semantic divergence. Conversely, $\mathrm{UIPA}_{\mathrm{LLM}}$ achieves the best and most stable results, effectively mitigating cross-platform divergence.

4.4 Hyper-Parameter Analysis (RQ3)

To analyze hyper-parameters' impact on UIPA, we apply LightGCN+UIPA to *Douban Movies* and *MovieLens-1m* and observe changes in HR@10. Four hyper-parameters are examined: the number of prototypes N, λ in Eq.12, α in Eq.2, and β in Eq.9-10. Specifically, we vary N in $\{10, 50, 100, 200, 300, 400\}$, λ in $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$, α, β in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The results are illustrated in Figure 5 and we have the following observations.

(1) **UIPA is robust to the number of prototypes.** For $N = 10 \sim 400$, UIPA consistently improves HR@10 compared to LightGCN (0.0463 on *Douban*



(a) HR@10 w.r.t N

(b) HR@10 w.r.t λ

(c) HR@10 w.r.t α

(d) HR@10 w.r.t β

Fig. 5. HR@10 of UIPA under different hyper-parameters

Movies and 0.1729 on *MovieLens-1m*), with optimal performance at $N = 100$. Performance declines slightly when the number of prototypes is too small or too large. With too few prototypes, each covers too many users or items, thus losing group-specific commonalities. In contrast, too many prototypes form tiny clusters, failing to capture shared characteristics. These findings highlight that prototypes effectively model user preferences and item features.

(2) **UIPA is robust to the loss weight.** For $\lambda = 1e-5 \sim 1e-1$, the HR@10 of UIPA varies from 0.0496 to 0.0510 on *Douban Movies* and from 0.1778 to 0.1801 on *MovieLens-1m*. Compared with LightGCN's HR@10, UIPA with all values of λ can improve at least 9.0% on *Douban Movies* and 2.5% on *MovieLens-1m*. The optimal performance is achieved at $\lambda = 1e-2$.

(3) **Appropriately introducing unified text embeddings and prototypes to item embeddings is effective for performance.** For α, the optimal performance is achieved at 0.3, as excessive text information (high α) reduces captured ID interactions. For β, the best performance is achieved at 0.2, as excessive prototype information (high β) causes embeddings within a cluster to share excessive common attributes, reducing specificity.

4.5 Performance on Sparser Datasets (RQ4)

Given that long-tail distributions are prevalent in real-world recommender systems, performance on datasets with sparse interactions is critical. To evaluate UIPA under varying sparsity levels, we randomly remove 0%-70% of interactions from datasets and apply LightGCN and LightGCN+UIPA.

As shown in Figure 6, recommendation performance decreases with increasing sparsity due to limited interactions, which hinder understanding user interests and item similarities. However, UIPA exhibits better performance improvements in sparser datasets, demonstrating its ability to leverage cross-platform information to alleviate the impact of data sparsity.

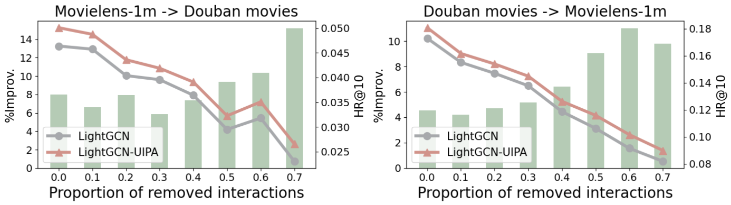
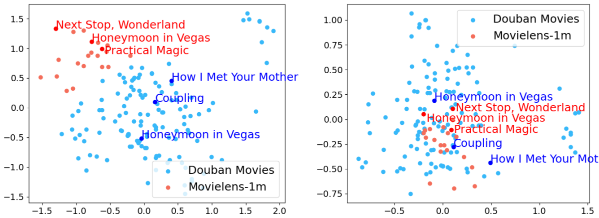


Fig. 6. Performance of LightGCN+UIPA and LightGCN on datasets at varying sparsity levels: Green bars represent improvements of LightGCN+UIPA over LightGCN.

4.6 Visualization



(a) LightGCN (b) LightGCN+UIPA

Fig. 7. PCA visualization of item embeddings with the same genre

To visualize the effect of UIPA, we use PCA to plot item embeddings for a specific prototype. As shown in Figure 7(a), without UIPA, embeddings of the same movie (e.g., "Honeymoon in Vegas") are distant, and movies of the same genre (e.g., comedy and romance) across datasets are also separated. In contrast, Figure 7(b) shows that with UIPA, embeddings of the same movie are closer, and embeddings of movies in the same genre across datasets are clustered in feature space. These results indicate that the prototypes in UIPA effectively capture item characteristics on different platforms, linking similar items and promoting cross-platform transfer.

5 Conclusion

In this paper, we identify two under-explored challenges in the cross-platform recommendation (CPR): the divergence in the item space and the divergence in the distribution of user preference. We propose a CPR framework based on **U**ser and **I**tem **P**rototype **A**lignment (UIPA) to tackle these challenges from the perspectives of both items and users. Moreover, UIPA is a plug-and-play framework that can be applied to single-platform collaborative filtering backbones and simultaneously enhance performance on both platforms. Extensive experiments on three pairs of real datasets show that UIPA provides a stable and significant improvement, outperforming SOTA CPR frameworks and models.

Acknowledgments. Chen Lin and Ying Wang are the corresponding authors. This work is supported by the Natural Science Foundation of China (No. 62173282, No. 62372390) and Fuzhou Inter-institutional Science and Technology Cooperation Project (2024-Y-018).

References

1. Bai, X., et al.: Improving text-based similar product recommendation for dynamic product advertising at yahoo. In: CIKM. pp. 2883–2892 (2022)
2. Cao, J., Cong, X., Liu, T., Wang, B.: Item similarity mining for multi-market recommendation. In: SIGIR. pp. 2249–2254 (2022)
3. Cao, J., Li, S., Yu, B., Guo, X., Liu, T., Wang, B.: Towards universal cross-domain recommendation. In: WSDM. pp. 78–86 (2023)
4. Choi, Y., et al.: Review-based domain disentanglement without duplicate users or contexts for cross-domain recommendation. In: CIKM. pp. 293–303 (2022)
5. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1). pp. 4171–4186 (2019)
6. Fan, Y., Si, N., Zhang, K.: Calibration matters: Tackling maximization bias in large-scale advertising recommendation systems. In: ICLR (2023)
7. Gao, C., et al.: Cross-domain recommendation with bridge-item embeddings. ACM Trans. Knowl. Discov. Data **16**(1), 2:1–2:23 (2022)
8. He, X., et al.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR. pp. 639–648 (2020)
9. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
10. Li, X., et al.: Gromov-wasserstein guided representation learning for cross-domain recommendation. In: CIKM. pp. 1199–1208 (2022)
11. Li, Z., et al.: Debiasing graph transfer learning via item semantic clustering for cross-domain recommendations. In: IEEE Big Data. pp. 762–769 (2022)
12. Li, Z., et al.: Semantic relation transfer for non-overlapped cross-domain recommendations. In: PAKDD. pp. 271–283 (2023)
13. Liu, W., et al.: Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In: WWW. pp. 1181–1190 (2022)
14. Liu, Y., et al.: U-NEED: A fine-grained dataset for user needs-centric e-commerce conversational recommendation. In: SIGIR. pp. 2723–2732 (2023)
15. Quan, Y., Ding, J., Gao, C., Yi, L., Jin, D., Li, Y.: Robust preference-guided denoising for graph based social recommendation. In: WWW. pp. 1097–1108 (2023)
16. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW. pp. 285–295 (2001)
17. Wang, Z., et al.: An industrial framework for personalized serendipitous recommendation in e-commerce. In: RecSys. pp. 1015–1018 (2023)
18. Xu, J., et al.: Heterogeneous and clustering-enhanced personalized preference transfer for cross-domain recommendation. Inf. Fusion **99**, 101892 (2023)
19. Yang, A., et al.: Baichuan 2: Open large-scale language models. arXiv Preprint (2023), https://arxiv.org/abs/2309.10305
20. Yu, W., Lin, X., Ge, J., Ou, W., Qin, Z.: Semi-supervised collaborative filtering by text-enhanced domain adaptation. In: KDD. pp. 2136–2144 (2020)
21. Zang, T., et al.: A survey on cross-domain recommendation: Taxonomies, methods, and future directions. ACM Trans. Inf. Syst. **41**(2), 42:1–42:39 (2023)
22. Zhang, Q., et al.: A deep dual adversarial network for cross-domain recommendation. IEEE Trans. Knowl. Data Eng. **35**(4), 3266–3278 (2023)
23. Zhang, X., Xu, S., Lin, W., Wang, S.: Constrained social community recommendation. In: KDD. pp. 5586–5596 (2023)
24. Zhao, C., Zhao, H., He, M., Zhang, J., Fan, J.: Cross-domain recommendation via user interest alignment. In: WWW. pp. 887–896 (2023)
25. Zhao, Y., et al.: Beyond the overlapping users: Cross-domain recommendation via adaptive anchor link learning. In: SIGIR. pp. 1488–1497 (2023)