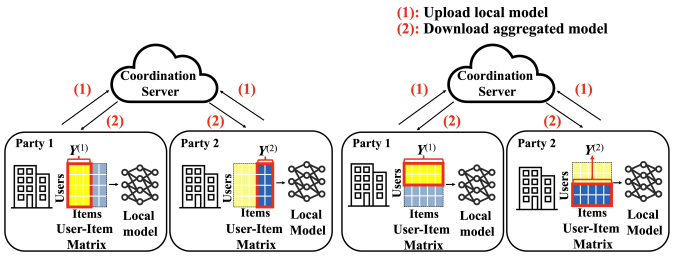
# Differentially Private Implicit Matrix Factorization

Xun Ran†, Qingqing Ye†, Xin Huang‡, Jianliang Xu‡, Haibo Hu†,
†The Hong Kong Polytechnic University, ‡Hong Kong Baptist University
*qi-xun.ran@connect.polyu.hk; qqing.ye@polyu.edu.hk; {xinhuang, xujl}@comp.hkbu.edu.hk; haibo.hu@polyu.edu.hk*

***Abstract*—Implicit Matrix Factorization (IMF) refers to Matrix Factorization (MF) based on users' implicit data (i.e., clients' actions or inactions). It serves as the backbone of many recommender systems for handling implicit feedback, such as webpage visits or bookmarks. Since these methods require a large amount of user data to provide accurate recommendations, data privacy has become a significant concern. Although Differential Privacy (DP) has been widely applied to MF to protect explicit data, the resulting utility loss makes it challenging to incorporate DP into IMF. In this study, we design a differentially private IMF, named DPIMF, using objective perturbation. To enhance utility, we redesign the loss function and adopt an importance sampling to reduce the noise scale in IMF. We provide formal utility guarantees for the proposed schemes and theoretically analyze the conditions for ensuring the optimal utility enhancement. Experimental results on three benchmark datasets validate our theoretical conclusions, demonstrating that the proposed schemes achieve a better trade-off between privacy and recommendation accuracy compared to state-of-the-art methods.**

## I. Introduction

Cross-silo collaborative recommendation is a key problem with extensive real-world applications [1], [2]. In this setting, the user-item interaction matrix is distributed across multiple parties. As shown in Figure 1, the partitioning typically follows two patterns [3]: parties share a common item set but hold disjoint user subsets (e.g., regional branches of a news platform serve different local users), or parties share a common user set but hold disjoint item subsets (e.g., different apps on the same device collect user behavior on distinct types of content). In both cases, the goal is to collaboratively learn from the full interaction matrix without any party exposing its raw data to others or to the coordination server [4], [5].




(b) Item set partitioning (a) User set partitioning

Fig. 1: Cross-silo recommendation settings.

Matrix factorization (MF) naturally separates user and item embeddings, aligning well with cross-silo data partitioning and ensures scalability and effectiveness [4], [6], [7], [8]. MF predicts a target user's preference for items based on the historical behavior data of all users. Many recommender systems

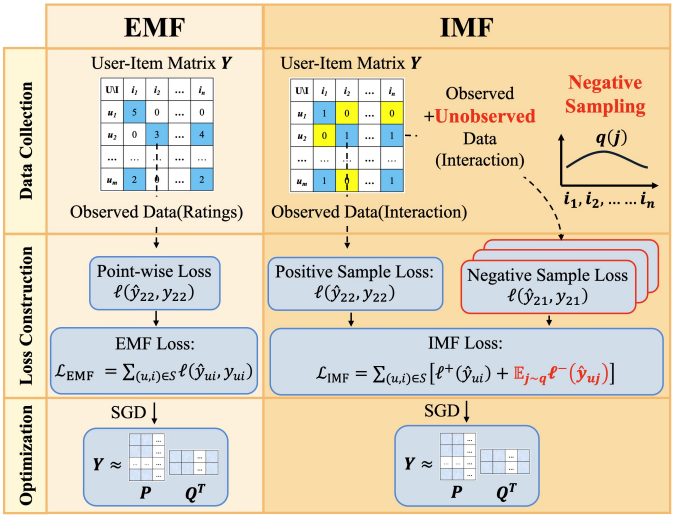



Fig. 2: Comparison of EMF and IMF

rely on users' explicit preferences, which are represented as numerical ratings (e.g., 1 to 5 stars in Netflix [9]). In more real-world applications, implicit feedback, which is represented as a sequence of bits 1 indicating a user's interaction with a set of items (e.g., purchasing goods or browsing webpages), has gained more attention in many scenarios due to its prevalence and ease of collection [10], [11], [12].

Although raw data is never directly shared, exchanging model updates can still leak sensitive information [13], [14]. To mitigate this, differential privacy (DP) is adopted to protect all shared information. However, existing studies primarily focus on explicit feedback-based MF (EMF) [15], [16], [17], [4], and they cannot be extended to implicit feedback-based MF (IMF) due to the fundamental differences in data structure and learning formulation.

Figure 2 shows the differences between EMF and IMF tasks. In EMF, user feedback is real-valued and fully observed, expressed by numerical ratings, allowing the learning objective to be constructed as a regression problem with objective function $\mathcal{L}_{EMF}$, where $y_{ui} \in \mathbb{R}$ is an observed explicit rating on item $i$ by user $u$, and $\ell$ is a standard pointwise loss. Each training instance contributes independently, and changes to any single data point affect only its corresponding term in the loss.

In contrast, IMF data is one-class, with only positive interactions observed [10]. This necessitates *negative sampling* to form the loss $\mathcal{L}_{IMF}$ where $\ell^+$ and $\ell^-$ denote the loss terms for positive and sampled negative items, and $q(j)$ is the sampling distribution. Unlike EMF, each data point in IMF

may have direct influence on the loss and indirect influence through its involvement in the sampling space, leading to *interdependencies*.

From a differential privacy perspective, such interdependence complicates the analysis of *sensitivity*—the maximum effect that changing a single data point can have on the output. Sensitivity directly determines the scale of noise needed to achieve DP guarantees. Compared to EMF, IMF typically has *higher and harder-to-control sensitivity*, which poses significant challenges for adopting standard DP mechanisms.

Several existing approaches attempt to mitigate the sensitivity issue by introducing DP at different stages of the learning process [15], [18], [19], [20], [4]. A common strategy is to perturb either the input (e.g., interaction data) or the gradient updates during training. *Input perturbation* methods (e.g., via randomized response [21] or sketching mechanisms [22]) inject noise directly into the user-item interaction matrix [19], [20]. However, implicit feedback data is inherently high-dimensional and extremely sparse. Adding noise at the input level often degrades data utility significantly. Moreover, these methods are typically designed for local DP (LDP) in cross-device settings and are less suitable for cross-silo scenarios. *Gradient perturbation* methods add calibrated noise to gradients [18], [4], [23]. Although this aligns with standard DP-SGD frameworks, the interdependent nature of IMF's loss—particularly the reliance on negative sampling—leads to accumulated noise across iterations. This results in degraded convergence and poor model quality.

In this work, we study IMF-based approaches that preserves $\varepsilon$-DP, namely DPIMF. We first present an objective perturbation-based method, which adds a one-off noise into the loss function and thus avoids excessive noise and accumulated error introduced by input and gradient perturbation-based methods. However, designing such a method while balancing model utility and privacy presents several challenges. First, in our setting, the privacy mechanism must mask a user's entire behaviors, including observed explicit feedback (in bit 1) and unobserved implicit feedback (in bit 0 or 1 potentially). Even more challenging, due to the extreme sparsity of data in recommender systems, a user's entire behavior set is far larger than the observed ones. Consequently, the noise scale of the DP mechanism can become unacceptably high. The excessive noise can also alter the properties of the loss function, potentially leading to unbounded solutions and degrading the utility of recommendations.

Resolving these issues paves the way for practical privacy-preserving recommender systems. To achieve this goal, we propose three strategies upon the following observations. First, the two loss terms of implicit feedback, defined over observed and unobserved data, can offset each other. This motivates us to redesign the negative sampling and the loss function of IMF to minimize DP sensitivity without significantly sacrificing model utility. Then, the objective perturbation is incorporated into the loss formulation, resulting in improved privacy-utility trade-offs while preserving the structure of implicit feedback learning.

Second, the noisy loss of IMF can be bounded if and only if its coefficient matrix is symmetric and positive definite. Therefore, we propose spectral regularization to ensure that the matrix satisfies these properties while maintaining the DP guarantee.

Third, we can enable privacy amplification, which reduces the injected noise without additional privacy loss. Due to the skewed distribution of implicit data, traditional uniform sampling can lead to the loss of informative points, negatively impacting model training effectiveness. To mitigate this problem, we design an importance sampling-based method. The privacy amplification achieved by importance sampling has been theoretically demonstrated to be effective.

The contributions of our work are summarized as follows:

1) To the best of our knowledge, DPIMF is the first work to study differentially private matrix factorization for implicit feedback in a cross-silo setting. To reduce the high sensitivity and mitigate utility loss, we redesign the loss function to suppress the sensitivity effectively, and implement spectral regularization to preserve the characteristics of loss terms.
2) By integrating importance sampling into DP-enabled MF, DPIMF achieves effective privacy amplification. By coupling sampling probabilities with the individual privacy loss and/or a suitable measure of "informativeness", we simultaneously improve both the privacy and accuracy in the subsampling mechanism.
3) By incorporating each utility-enhancing technique, we provide utility guarantees for DPIMF, offering an in-depth understanding of its theoretical properties. We further prove that the proposed schemes preserve a pure $\varepsilon$-DP.
4) Through extensive evaluation on real-world datasets, DPIMF achieves the best recommendation accuracy compared to state-of-the-art solutions. The results validate our theoretical findings and demonstrate the effectiveness of our approach in improving model utility.

The remainder of this paper is organized as follows. In Section II, we introduce preliminaries and the studied problem. Sections III and IV present the proposed schemes in detail and Section V provides the theoretical analysis of the proposed schemes. The experimental results are presented and analyzed in Section VI. The related work is discussed in Section VII. Section VIII concludes the paper.

## II. Preliminaries and Problem Formulation

### A. Implicit Matrix Factorization

IMF aims to predict a user's preference for unseen items based on binary implicit feedback (1 for observed, 0 for unobserved). The data is typically sparse, and unobserved entries may still reflect positive preference. Matrix Factorization (MF) is a standard approach for this task [24], [7].

Let $\boldsymbol{U}$ and $\boldsymbol{I}$ be the sets of users and items in a recommender system. The implicit data can be expressed as a matrix $\boldsymbol{Y} \in \mathbb{R}^{|\boldsymbol{U}|\times|\boldsymbol{I}|}$, where its entry $y_{ui} \in \{1, 0\}$ ($u \in [1, |\boldsymbol{U}|]$, $i \in [1, |\boldsymbol{I}|]$) records whether the user $u$ has interacted with

the item $i$ or not[1]. MF-based models factorize this binary matrix $\boldsymbol{Y}$ into the user and item profile matrices $\boldsymbol{P} \in \mathbb{R}^{d\times|\boldsymbol{U}|}$, $\boldsymbol{Q} \in \mathbb{R}^{d\times|\boldsymbol{I}|}$, where $d \ll \min(|\boldsymbol{U}|, |\boldsymbol{I}|)$ is the latent dimension. The interacting probability of user $u$ on item $i$ is estimated by $\boldsymbol{p}_u^T \boldsymbol{q}_i$, where $\boldsymbol{p}_u \in \mathbb{R}^d$ and $\boldsymbol{q}_i \in \mathbb{R}^d$ are column vectors (of $\boldsymbol{P}$ and $\boldsymbol{Q}$) that represent the user and item profiles, respectively.

To learn $\boldsymbol{P}$ and $\boldsymbol{Q}$, the state-of-the-art methods [25], [7], [6] suggest minimizing the following loss function:

$$L(\boldsymbol{P}, \boldsymbol{Q}) = \sum_{(u,i)\in S} (\boldsymbol{p}_u^T \boldsymbol{q}_i - 1)^2 + \alpha_0 \sum_{(u,i)\in\Omega} (\boldsymbol{p}_u^T \boldsymbol{q}_i)^2 \quad (1)$$

Here, $\Omega$ is the universe of all user-item pairs (i.e., $|\Omega| = |\boldsymbol{U}| \times |\boldsymbol{I}|$), and $S$ includes the pairs of only positive data. The first term of $L(\boldsymbol{P}, \boldsymbol{Q})$ measures the distance between predictions for positive data in $S$ and the ground-truth label (i.e., 1). The second term is defined over $\Omega$ and the trade-off between the two terms is controlled by $\alpha_0$. To prevent overfitting, a regularization term $R(\boldsymbol{P}, \boldsymbol{Q})$ is introduced to penalize the energy of $\boldsymbol{P}$ and $\boldsymbol{Q}$: $L(\boldsymbol{P}, \boldsymbol{Q}) \leftarrow L(\boldsymbol{P}, \boldsymbol{Q}) + R(\boldsymbol{P}, \boldsymbol{Q})$, and $R(\boldsymbol{P}, \boldsymbol{Q})$ is defined as

$$R(\boldsymbol{P}, \boldsymbol{Q}) = \sum\nolimits_{u\in\boldsymbol{U}} \rho_u \|\boldsymbol{p}_u\|_2^2 + \sum\nolimits_{i\in I} \rho_i \|\boldsymbol{q}_i\|_2^2, \quad (2)$$

where $\rho_u = \lambda(|\boldsymbol{I}_u| + \alpha_0|\boldsymbol{I}|)$, $\rho_i = \lambda(|\boldsymbol{U}_i| + \alpha_0|\boldsymbol{U}|)$, $\boldsymbol{I}_u = \{i : (u,i) \in S\}$, $\boldsymbol{U}_i = \{u : (u,i) \in S\}$. Eq. (2) is known as the frequency-based regularizer, which imposes heavier regularization on frequent items and users, and its strength is tuned by $\lambda$ in $\rho_u$. To minimize $L(\boldsymbol{P}, \boldsymbol{Q})$, one of the most commonly used algorithms is alternating least squares (ALS) algorithm which optimizes $\boldsymbol{P}$ and $\boldsymbol{Q}$ alternatively [15].

### B. Differential Privacy

Differential Privacy (DP) ensures that outputs reveal little about any individual, regardless of auxiliary knowledge [26]. Formally:

*Definition 1:* ($\varepsilon$-differential privacy [26].) An algorithm $\mathcal{A}$ satisfies $\varepsilon$-differential privacy ($\varepsilon$-DP), where $\varepsilon \geq 0$, if and only if for $\forall\ O \subseteq \textit{Range}(\mathcal{A})$ and any neighboring datasets $D$ and $D'$, it satisfies

$$\Pr[\mathcal{A}(D) \in O] \leq \exp(\varepsilon) \Pr[\mathcal{A}(D') \in O], \quad (3)$$

where $\exp(\cdot)$ denotes the exponential function and $\textit{Range}(\mathcal{A})$ denotes the set of all possible outputs of the algorithm $\mathcal{A}$.

The privacy budget $\varepsilon$ controls the level of privacy: smaller $\varepsilon$ implies stronger protection. In MF, neighbors differ by one rating; in IMF, by one user's full behavior row, increasing sensitivity.

*Definition 2:* (Sensitivity [26].) For a function $f : D \to \mathbb{R}$, the sensitivity of $f$, denoted as $\Delta_f$, is defined as:

$$\Delta_f = \max_{D,D'} \|f(D) - f(D')\|_1, \quad (4)$$

where $D$ and $D'$ are neighboring datasets.

To enforce DP, the Laplace mechanism adds noise scaled to sensitivity:

[1]For EMF-based methods, the difference is that $\boldsymbol{Y}$ is not binary but contains numerical rating scores. All subsequent MF optimizations are changed accordingly.

*Definition 3:* (Laplace mechanism [26].) Given a function $f : D \to \mathbb{R}^d$, the following mechanism $\mathcal{A}$ satisfies $\varepsilon$-DP:

$$\mathcal{A}(D) = f(D) + \textit{Lap}(\Delta_f/\varepsilon)^d, \quad (5)$$

where $\textit{Lap}(b)$ denotes a random variable drawn from a Laplace distribution with zero mean and scale $b$, $\Delta_f$ is the sensitivity of $f$ and $d$ represents the dimension of $f$.

In practical applications of DP, an algorithm is often a combination of a series of operations over the disjoint subsets of a dataset [26]. For this case, the overall privacy protection level the algorithm provides can be quantified through the parallel theorem.

*Theorem 1:* (Parallel composition [27].) Let $\mathcal{A}_1$, $\mathcal{A}_2$, $\cdots$, $\mathcal{A}_k$ be $k$ algorithms that satisfy $\varepsilon_1$-DP, $\varepsilon_2$-DP, $\cdots$, $\varepsilon_k$-DP, respectively. $D_1$, $D_2$ $\cdots$ $D_k$ are $k$ disjoint partitions of a dataset $D$, where $D_1 \cup D_2, \cdots, \cup D_k = D$. Then publishing $\mathcal{A}_1(D_1), \mathcal{A}_2(D_2), \cdots, \mathcal{A}_k(D_k)$ satisfies $\max_{i\in\{1,\cdots,k\}} \{\varepsilon_i\}$-DP.

In IMF, modifying all behaviors of a user leads to higher sensitivity than explicit-feedback MF, which only hides interaction values; as a result, more noise is needed to satisfy $\varepsilon$-DP, reducing accuracy.

### C. Problem Formulation

Our study follows a cross-silo setting in prevalent recommender systems [4], [16], [17], in which multiple parties collaboratively train recommendation models without directly sharing their data. All parties share a common subset of users or items, but each possesses distinct user behavior data. A typical application involves two companies, such as an e-book platform and a local retailer, share some common users but each has different user-behavior information. They want to collaboratively train a recommendation model that learns from both users' behaviors on digital book purchases and physical items bought in the local store. Overlapping items, such as books available in both digital and physical formats, allow the model to leverage feedback from users in both contexts for those shared items.

Figure 3 illustrates this process using an example involving two parties. The users of party $k$ ($k \in \{1, 2\}$), denoted as $\boldsymbol{U}^{(k)}$, provide their implicit feedback on item set $\boldsymbol{I}^{(k)}$ to the trusted recommender $k$ (Step (1)). The recommender then trains a local IMF model to derive user and item profile matrices, i.e., $\boldsymbol{P}_{U^{(k)}}$ and $\boldsymbol{Q}_{I^{(k)}}$ (Step (2)), and uploads the matrices to the coordination server (Step (3)). Note that two parties share a common user set $\hat{\boldsymbol{U}}$ and item set $\hat{\boldsymbol{I}}$, highlighted in blue and yellow in the user-item matrix. Therefore, according to Eq. (7) as below, the server aggregates the reported user and item profile matrices over the common user set $\hat{\boldsymbol{U}}$ and item set $\hat{\boldsymbol{I}}$, and then sends the results (i.e., $\boldsymbol{P}^*_{\hat{U}}$ or $\boldsymbol{Q}^*_{\hat{I}}$) back to each party to refine local user and item profile matrices for the next iteration (Step (4)). As for the server-side aggregation, we adopt a general framework that supports various aggregation strategies. Let $\{S_1, \cdots, S_K\}$ be a partition of the set of user-item pairs of all observed data. For each party $k$, the local loss

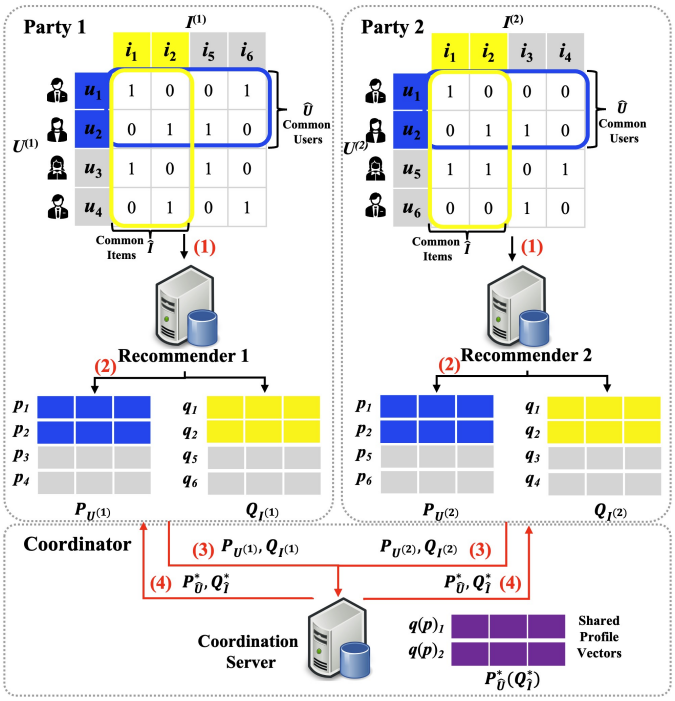


Fig. 3: Scenario of DPIMF.

function is

$$L(\boldsymbol{P}_{U^{(k)}}, \boldsymbol{Q}_{I^{(k)}}) = \sum_{(u,i)\in S_k} (\boldsymbol{p}_u^T \boldsymbol{q}_i - 1)^2 + \alpha_0 \sum_{(u,i)\in\Omega} (\boldsymbol{p}_u^T \boldsymbol{q}_i)^2 + R(\boldsymbol{P}_{U^{(k)}}, \boldsymbol{Q}_{I^{(k)}}) \quad (6)$$

where $U^{(k)} = \{u \mid (u,i) \in S_k\}$, $I^{(k)} = \{i \mid (u,i) \in S_k\}$. The global loss function can be written as:

$$L(\boldsymbol{P}, \boldsymbol{Q}) = \sum_k w_k L(\boldsymbol{P}_{U^{(k)}}, \boldsymbol{Q}_{I^{(k)}}) \quad (7)$$

where $w_k$ is the aggregation weight assigned to party $k$.

Regarding the threat model, we assume the coordination server and all recommenders are honest-but-curious, i.e., they follow the protocol we have designed but may attempt to infer user data by analyzing the communication transcript between the recommenders and the coordination server (represented by the red arrows in Figure 3).

Based on the threat model and the loss function, learning and publishing user or item profile matrices is the building block of a privacy-preserving recommender system in the considered scenario [28], [16], [29]. Following this line, the goal of this study is to design a scheme for learning and publishing the private profile vectors, satisfying the rigorous privacy setting ($\varepsilon$-DP) while preserving model utility to the greatest extent possible.

## III. A Strawman Solution for DPIMF

In this section, we first introduce a paradigm for solving differentially private implicit matrix factorization (DPIMF) problem in a cross-silo setting. Then a strawman implementation using an objective perturbation mechanism is developed. Recall that the user and the item profile matrices, $\boldsymbol{P}$ and $\boldsymbol{Q}$, are symmetrically evaluated in the loss function of Eq. (1). A

**Algorithm 1:** A Paradigm for DPIMF (Solving Private Item Profile Matrix)

**Input :** $\{S_1, S_2, \ldots, S_K\}$owned by $K$ different parties; the total iteration $T$.
**Output:** User profile matrix $\boldsymbol{P}$; Private item profile matrix $\bar{\boldsymbol{Q}}$

1 **// Phase 1: Initialization**
2 Server initializes item profile matrix $\boldsymbol{Q}$;
3 Each party $k \in \{1, 2, \ldots, K\}$ initializes local profile matrix $\boldsymbol{P}_{U^{(k)}}$;
4 **// Phase 2: Collaborative Learning**
5 **for** $t \in \{1, 2, \ldots, T\}$ **do**
6 **// Local Update** (in each party)
7 **for** *each item* $i \in \boldsymbol{I}$ *(in parallel)* **do**
8 Loss Construction: $L(q_i)$;
9 Objective Perturbation: $\bar{L}(q_i) \leftarrow L(q_i)$;
10 Private Learning: $\bar{\boldsymbol{q}}_i = \arg\min_{\boldsymbol{q}_i : \|\boldsymbol{q}_i\|_2 \le \sqrt{1/\lambda}} \bar{L}(\boldsymbol{q}_i)$.
11 **// Global Update**
12 Server aggregates and shares the private item profile matrices $\{\bar{\boldsymbol{Q}}_{I^{(1)}}, \cdots, \bar{\boldsymbol{Q}}_{I^{(K)}}\}$ to all parties.
13 **// Phase 3: Local Fine-tuning**
14 Each party $k \in \{1, 2, \ldots, K\}$ refines $\boldsymbol{P}_{U^{(k)}}$ locally.

private item profile matrix $\bar{\boldsymbol{Q}}$ and user profile matrix $\bar{\boldsymbol{P}}$ can be solved by minimizing the loss, i.e.,

$$\bar{\boldsymbol{P}}, \bar{\boldsymbol{Q}} = \arg\min L(\boldsymbol{P}, \boldsymbol{Q})$$

where the optimization processes for both matrices are identical. For clarity and readability, we introduce our method with a focus on solving $\bar{\boldsymbol{Q}}$, while $\bar{\boldsymbol{P}}$ can be solved in a similar manner.

As outlined in Algorithm 1, the paradigm for solving DPIMF consists of three phases. In **Phase 1 (Initialization)**, local and global profile vectors are initialized locally by each party and the server, respectively (Lines 1-3). In **Phase 2 (Collaborative Learning)**, all parties collaborate with the server to learn the shared profile matrix (Lines 4-12), which includes two processes, namely **Local Updates** and **Global Updates**. The local updates involve three key steps: Loss Construction, Objective Perturbation, and Private Learning, which will be detailed later. Once the process of local updates complies with DP, the subsequent global updates maintain DP compliance thanks to the post-processing theorem. In **Phase 3 (Local Fine-tuning)**, each party refines its local profile matrix using the global profile matrix (Lines 13-14).

Now let's elaborate on the implementation of the key steps in local update by each client.

**Loss Construction**: Given that the loss evaluation of each item is independent, we construct the loss function for each item in parallel. This approach not only facilitates the application of the parallel theorem but also accelerates the overall algorithm. For a specific item $i$, the loss function for its profile $\boldsymbol{q}_i$ is derived from Eq. (1) as

$$L^{\mathrm{O}}(\boldsymbol{q}_i) = \sum_{u\in\boldsymbol{U}_i} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i - 1\right)^2 + \alpha_0 \sum_{u\in\boldsymbol{U}} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i\right)^2 + \rho_i \|\boldsymbol{q}_i\|_2^2 \quad (8)$$

where $\boldsymbol{U}$ denotes the universe of users involved in the recommender system, and $\boldsymbol{U}_i$ denotes the set of users who interact with item $i$ (where the interaction is labelled as 1 in the user-item matrix). This set is sensitive in nature.

Since the sophisticated form of $L^{\mathrm{O}}(\boldsymbol{q}_i)$ complicates the

sensitivity analysis, we expand $L^{\mathrm{O}}(\boldsymbol{q}_i)$ as a polynomial. By omitting the constant term, Eq. (8) can be rewritten as

$$l(\boldsymbol{q}_i) = \boldsymbol{q}_i^T(\boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0 \boldsymbol{G}_{\boldsymbol{U}})\boldsymbol{q}_i - \boldsymbol{q}_i^T(2\boldsymbol{g}_{\boldsymbol{U}_i}) + \rho_i \|\boldsymbol{q}_i\|_2^2, \quad (9)$$

where $\boldsymbol{G}_{\boldsymbol{U}_i} = \sum_{u \in \boldsymbol{U}_i}(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)$, $\boldsymbol{G}_{\boldsymbol{U}} = \sum_{u \in \boldsymbol{U}}(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)$, $\boldsymbol{g}_{\boldsymbol{U}_i} = \sum_{u \in \boldsymbol{U}_i} \boldsymbol{p}_u$, and $\boldsymbol{p}_u \otimes \boldsymbol{p}_u \in \mathbb{R}^{d \times d}$ is the outer product.

**Objective Perturbation**: By expanding the objective function, we can analyze its sensitivity, which determines the scale of DP noise. Note that the change of implicit matrix (either $1 \to 0$ or $0 \to 1$) only affects $\boldsymbol{U}_i$ in Eq. (9). Without loss of generality, let $\boldsymbol{U}_i'$ denote the set obtained by adding a user $v$ to $\boldsymbol{U}_i$. The sensitivity between $\boldsymbol{U}_i$ and $\boldsymbol{U}_i'$ satisfies

$$\|l_{\boldsymbol{U}_i}(q_i) - l_{\boldsymbol{U}_i'}(q_i)\|_1 \leq \boldsymbol{q}_i^T \|A(\boldsymbol{U}_i) - A(\boldsymbol{U}_i')\|_1 \boldsymbol{q}_i$$

where $\|A(\boldsymbol{U}_i) - A(\boldsymbol{U}_i')\|_1$ measures the difference between the summations of coefficients, denoted as $A(\boldsymbol{U}_i)$ and $A(\boldsymbol{U}_i')$. We can derive the sensitivity based on the upper bound of this difference:

$$\|A(\boldsymbol{U}_i) - A(\boldsymbol{U}_i')\|_1 \leq \max_{u \in \boldsymbol{U}} (2\|\boldsymbol{p}_u\|_1 + \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} + 1) \quad (10)$$

Here, $\|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}}$ denotes $\sum_{1 \leq j,l \leq d} |(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)_{jl}|$. Thus, the sensitivity on the corresponding coefficients of $l(\boldsymbol{q}_i)$ is given by

$$\Delta = \max_{u \in \boldsymbol{U}} (2\|\boldsymbol{p}_u\|_1 + \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} + 1). \quad (11)$$

To ensure $\varepsilon$-DP, Laplacian noises are sampled and added to $l(\boldsymbol{q}_i)$. The perturbed loss thus becomes

$$\bar{l}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T (\boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0 \boldsymbol{G}_{\boldsymbol{U}} + \boldsymbol{B}) \boldsymbol{q}_i - \boldsymbol{q}_i^T (2\boldsymbol{g}_{U_i} + \boldsymbol{b}) + (\rho_i + \lambda\eta) \|\boldsymbol{q}_i\|_2^2, \quad (12)$$

where $\boldsymbol{B} \sim Lap\left(\frac{\Delta}{\varepsilon}\right)^{d \times d}$, $\boldsymbol{b} \sim Lap\left(\frac{\Delta}{\varepsilon}\right)^{d}$ and $\eta \sim Lap\left(\frac{\Delta}{\varepsilon}\right)$.

**Private Learning**: It can be proved that the optimal profile $\boldsymbol{q}_i^* = \arg\min L^{\mathrm{O}}(\boldsymbol{q}_i)$, obtained by minimizing Eq. (8) in the non-private setting, satisfies $\|\boldsymbol{q}_i^*\|_2 \leq \sqrt{1/\lambda}$. For brevity, we defer the proof of this result to the Lemma 1 in appendix.

To satisfy the same constraint and enhance utility, we derive the private profile $\bar{\boldsymbol{q}}_i$ by minimizing $\bar{l}(\boldsymbol{q}_i)$ over the convex set $C = \{\boldsymbol{q}_i \mid \|\boldsymbol{q}_i\|_2 \leq \sqrt{1/\lambda}\}$. Repeating the above steps for all items ($i \in [1, |\boldsymbol{I}|]$), we can obtain the private matrix $\bar{\boldsymbol{Q}}$.

**Remark**. The method outlined above satisfies $\varepsilon$-DP. However, it suffers from high sensitivity. According to Eq. (9), the sensitivity analysis must account for the effects of polynomial terms with different orders, as well as the regularization term. Specifically, by letting $c = \max_{u \in \boldsymbol{U}, j \in \{1,2,\ldots,d\}} |p_{uj}|$, $\Delta$ is bounded by $(2cd + c^2d^2 + 1)$. This bound is dominated by its quadratic term, which can be extremely large when either $c$ or $d$ is large. The high sensitivity comes at an expensive cost of model utility.

## IV. The Full-fledged DPIMF

Although the strawman solution works for DPIMF, it still suffers from utility degradation due to the large noise scale required to protect the vast amount of unobserved data. In this section, we address these issues and propose a full-fledged solution for DPIMF, resulting in more accurate recommendation.

### A. Overview

Our solution reduces the noise scale significantly, which is inspired by several key insights. First, we observe that the noise scale is determined by the sensitivity, which quantifies how much the loss function responds to changes in the implicit data. We aim to investigate whether there is the optimal way to construct the loss function that strikes a better balance between sensitivity and utility. Second, we identify that the issue of unbounded loss arises from the non-positive definite coefficient matrix in the noisy loss. To address it, we can regularize this matrix, ensuring it remains positive definite. Finally, considering the increasing noise scale, we explore a sampling process on the training data to amplify the differential privacy guarantee without sacrificing utility.

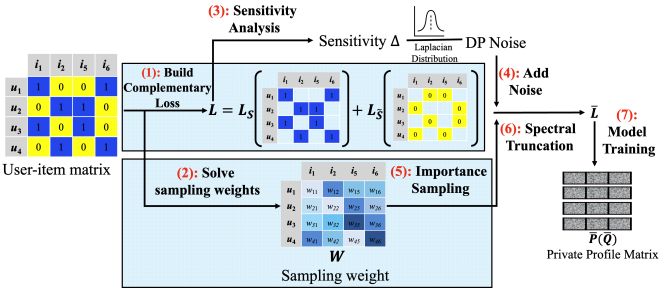


Fig. 4: Overview of the full-fledged DPIMF.

Motivated by these insights, we have developed three strategies over the strawman solution — constructing complementary loss, applying spectral truncation, and exploiting importance sampling. These strategies have shown to effectively reduce the noise scale and improve the utility of the method. An overview of the approach is illustrated in Figure 4. Using implicit feedback, we first construct a complementary loss function defined over both observed and unobserved data (Step (1)). We then derive the sensitivity of the complementary loss (Step (2)). The importance weights of the data are computed and used for importance sampling (Steps (3) and (4)). After sampling, noise drawn from a Laplacian distribution is added to the loss, and spectral truncation is applied as a post-processing step (Steps (5) and (6)). Finally, we obtain a private loss $\bar{L}$, which enables us to compute the private user/item profile matrix (Step (7)).

### B. Complementary Loss

In this section, we present the design of complementary loss, which enhances the utility of DPIMF by limiting DP sensitivity.

Our design of complementary loss is built upon the following observation. The sensitivity analysis on the loss function of Eq. (9) is derived from the original loss function of Eq. (1), which is defined over the positive samples $S$ and the overall samples $\Omega$. Consequently, the effect of changes in implicit rating data only occurs in positive samples in $S$. We find that this effect can be mitigated by including a term defined over the non-positive samples, i.e., $\tilde{S} = \Omega \backslash S$, thereby reducing sensitivity.

Without loss of generality, for a given dataset $D$ and a function $f$, we consider a cumulative query defined as $F(D) = \sum_{d \in D} f(d)$. Both $F(S)$ and $F(\tilde{S})$ will respond to changes in the dataset $S$ (or $\tilde{S}$), and their responses are

always opposite. For instance, when a record $k$ is removed from $S$, it will appear in $\tilde{S}$ because the two sets are mutually exclusive and exhaustive (i.e., $S \cap \tilde{S} = \emptyset, S \cup \tilde{S} = \Omega$). Corresponding to this change, the increment of $F(S)$ is $\Delta F(S) = F(S^{-k}) - F(S^k) = -f(k)$, while the increment of $F(\tilde{S})$ is $\Delta F(\tilde{S}) = F(\tilde{S}^k) - F(\tilde{S}^{-k}) = f(k)$, where $D^k$ and $D^{-k}$ denote the dataset $D$ with and without record $k$ respectively. Notably, $\Delta F(S)$ and $\Delta F(\tilde{S})$ complement each other, and their sum equals zero. Thus, by adding a term over $\tilde{S}$, we can offset the overall impact when implicit rating data changes, leading to reduced sensitivity.

Based on this observation, we redesign the loss function of MF in Eq. (1) as follows:

$$L(\boldsymbol{P},\boldsymbol{Q}) = \sum_{(u,i)\in S} (\boldsymbol{p}_u^T \boldsymbol{q}_i - 1)^2 + \alpha_0 \sum_{(u,i)\in \tilde{S}} (\boldsymbol{p}_u^T \boldsymbol{q}_i)^2 + R(\boldsymbol{P},\boldsymbol{Q}) \quad (13)$$

Compared with Eq. (1), the second term in $L(\boldsymbol{P},\boldsymbol{Q})$ is modified to keep the predictions close to zero only for the non-positive entries of $\tilde{S}$, rather than all entries of $\Omega$. Moreover, the regularizer $R(\boldsymbol{P},\boldsymbol{Q})$ needs to be modified in a similar way:

$$R(\boldsymbol{P},\boldsymbol{Q}) = \lambda \left( \sum\nolimits_{u\in\boldsymbol{U}} \left( |\boldsymbol{I}_u| + \alpha_0 |\tilde{\boldsymbol{I}}_u| \right) \|\boldsymbol{p}_u\|_2^2 + \sum\nolimits_{i\in I} \left( |\boldsymbol{U}_i| + \alpha_0 |\tilde{\boldsymbol{U}}_i| \right) \|\boldsymbol{q}_i\|_2^2 \right), \quad (14)$$

where $\tilde{\boldsymbol{I}}_u := \{i : (u,i) \in \tilde{S}\}$, $\tilde{\boldsymbol{U}}_i := \{u : (u,i) \in \tilde{S}\}$. Armed with the new formulation, the loss function $L(\boldsymbol{q}_i)$ for item $i$ becomes

$$L(\boldsymbol{q}_i) = \underbrace{\sum\nolimits_{u\in\boldsymbol{U}_i} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i - 1\right)^2 + \alpha_0 \sum\nolimits_{u\in\tilde{\boldsymbol{U}}_i} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i\right)^2}_{L_I(\boldsymbol{q}_i)} + \underbrace{\lambda \left( |\boldsymbol{U}_i| + \alpha_0 |\tilde{\boldsymbol{U}}_i| \right) \|\boldsymbol{q}_i\|_2^2}_{R(\boldsymbol{q}_i)}. \quad (15)$$

We will prove in Section V-B that the effect of this change in the loss function is always bounded and can be made arbitrarily small under proper parameter settings.

For the ease of analysis, we divide $L(\boldsymbol{q}_i)$ into two components: $L_I(\boldsymbol{q}_i)$ and $R(\boldsymbol{q}_i)$. The first component $L_I(\boldsymbol{q}_i)$ consists of two quadratic terms defined over the positive and non-positive rating data associated with user sets $\boldsymbol{U}_i$ and $\tilde{\boldsymbol{U}}_i$, respectively. The second component $R(\boldsymbol{q}_i)$ is the regularization term, where its weight is also defined over $\boldsymbol{U}_i$ and $\tilde{\boldsymbol{U}}_i$. Omitting the constants, the expansion of $L_I(\boldsymbol{q}_i)$ is thus

$$l_I(\boldsymbol{q}_i) = \boldsymbol{q}_i^T (\boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0 \boldsymbol{G}_{\tilde{\boldsymbol{U}}_i}) \boldsymbol{q}_i - 2\boldsymbol{q}_i^T \boldsymbol{g}_{\boldsymbol{U}_i}. \quad (16)$$

where $\boldsymbol{G}_{\tilde{\boldsymbol{U}}_i} = \sum_{u\in\tilde{\boldsymbol{U}}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u$. Different from Eq. (11), we treat the polynomial terms with different orders as independent queries. This will allow us to calibrate the noise for different terms based on their own sensitivities. Same as in Section III, we use $\boldsymbol{U}_i'$ to denote the neighboring data obtained by adding a user $v$ to $\boldsymbol{U}_i$. Let $c = \max_{u\in\boldsymbol{U}, j\in\{1,2,\ldots,d\}} |p_{uj}|$, $\Delta$, the first-order term of Eq. (16), same as in Eq. (11), its coefficients satisfy

$$\left\| 2 \sum_{j=1}^{d} \left( \sum_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u - \sum_{u\in\boldsymbol{U}_i'} \boldsymbol{p}_u \right)_j \right\|_1 \le 2cd.$$

Thus, we define the sensitivity of the first-order term as

$$\Delta_1 = 2cd. \quad (17)$$

For the second-order term of Eq. (16), the following bound is hold

$$\begin{aligned} &\left\| \sum\nolimits_{1\le j,l\le d} \left( \sum\nolimits_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \sum\nolimits_{u\in\boldsymbol{U}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum\nolimits_{u\in\tilde{\boldsymbol{U}}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \alpha_0 \sum\nolimits_{u\in\tilde{\boldsymbol{U}}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u \right)_{jl} \right\|_1 \\ &\le \sum\nolimits_{1\le j,l\le d} (1-\alpha_0) |(\boldsymbol{p}_v \otimes \boldsymbol{p}_v)_{jl}| \\ &\le \max_{u\in\boldsymbol{U}} \left( (1-\alpha_0) \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} \right) \le (1-\alpha_0) d^2. \end{aligned}$$

Therefore, the sensitivity of the second-order term is

$$\Delta_2 = (1-\alpha_0) d^2. \quad (18)$$

For the regularization term $R(\boldsymbol{q}_i)$ of Eq. (15), we have

$$\left\| |\boldsymbol{U}_i| - |\boldsymbol{U}_i'| + \alpha_0 |\tilde{\boldsymbol{U}}_i| - \alpha_0 |\tilde{\boldsymbol{U}}_i'| \right\|_1 \le 1 - \alpha_0$$

The sensitivity of $R(\boldsymbol{q}_i)$ is in turn defined as

$$\Delta_3 = 1 - \alpha_0. \quad (19)$$

Compared to the counterparts in Eq. (11), the sensitivities of the second-order term and the regularization term are both reduced by a factor of $(1-\alpha_0)$. This agrees with our intuition mentioned earlier that the effect of the removing/adding one rating value in $\boldsymbol{U}_i$ can be offset by introducing the loss defined over $\tilde{\boldsymbol{U}}_i$. By controlling the parameter $\alpha_0$, the sensitivity can be significantly reduced. For the trade-off between the sensitivity scale and model utility, the setting of $\alpha_0$ will be theoretically analyzed in Section V.

Our second design is built upon the following observation. The coefficients of the polynomial in Eq. (16) are defined over complementary sets $\boldsymbol{U}_i$ and $\tilde{\boldsymbol{U}}_i$, implying the fact that they are independent of each other. Thus, the matrix $(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)$ is symmetric.

From this observation, we can get a tighter bound of the sensitivity $\Delta_2$ as

$$\Delta_2 = \max_{u\in\boldsymbol{U}} \left( (1-\alpha_0) \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}} \right) \le \frac{cd(d\alpha_0 - 1)}{2}, \quad (20)$$

where $\|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}} = \sum_{j\le l} |(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)_{jl}|$ is computed on the values in the upper triangular elements of the matrix. Further, the symmetric noise matrix associated with the sensitivity $\Delta_2$ is

$$\tilde{\boldsymbol{B}} = \text{triu}(\boldsymbol{B}) + \text{tril}_{-1}(\boldsymbol{B}^T), \quad (21)$$

where $\boldsymbol{B}$ is a noise matrix of i.i.d. Laplacian entries, the function $\text{triu}(\cdot)$ outputs a copy of the input matrix with elements below the main diagonal zeroed, and $\text{tril}_{-1}(\cdot)$ outputs a copy of the input with elements above and the main diagonal zeroed. We will prove that the sensitivity and the symmetric noise matrix complies with DP in Section V.

Corresponding to the sensitivities $\Delta_1$, $\Delta_2$ and $\Delta_3$, the privacy budget $\varepsilon$ is divided into three parts as $\varepsilon_1$, $\varepsilon_2$, $\varepsilon_3$, where $\varepsilon_i = \varepsilon\beta_i$ with $\beta_i \in [0,1]$ and $\sum_{i=1}^3 \beta_i = 1$. And the Laplacian noises are sampled and added to $l_I(\boldsymbol{q}_i)$ and $R(\boldsymbol{q}_i)$ as

$$\begin{aligned} \bar{l}_I(\boldsymbol{q}_i) &= \boldsymbol{q}_i^T (\boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0 \boldsymbol{G}_{\tilde{\boldsymbol{U}}_i} + \tilde{\boldsymbol{B}}) \boldsymbol{q}_i - \boldsymbol{q}_i^T (2\boldsymbol{g}_{\boldsymbol{U}_i} + \boldsymbol{b}), \\ \bar{R}(\boldsymbol{q}_i) &= \lambda (|\boldsymbol{U}_i| + \alpha_0 |\tilde{\boldsymbol{U}}_i| + \eta) \|\boldsymbol{q}_i\|_2^2. \end{aligned} \quad (22)$$

Here, $\boldsymbol{b} \sim Lap\left(\frac{\Delta_1}{\varepsilon_1}\right)^d$, $\eta \sim Lap\left(\frac{\Delta_3}{\varepsilon_3}\right)$, and $\tilde{\boldsymbol{B}}$ is generated from $\boldsymbol{B} \sim Lap\left(\frac{\Delta_2}{\varepsilon_2}\right)^{d\times d}$. In particular, when $\alpha_0 = 1$, $\Delta_2 = \Delta_3 = 0$ and no noise needs to be added to the second-order term in $l_I(\boldsymbol{q}_i)$ and $R(\boldsymbol{q}_i)$. In this case, the whole privacy budget $\varepsilon$ is used to generate the noise vector $\boldsymbol{b}$. The effect of this setting will be analyzed in Section V. Combining the above two components, we get the private objective function

$$\bar{l}(\boldsymbol{q}_i) = \bar{l}_I(\boldsymbol{q}_i) + \bar{R}(\boldsymbol{q}_i). \tag{23}$$

Same as in Section III, we minimize $\bar{l}(\boldsymbol{q}_i)$ over the convex set $C = \{\boldsymbol{q}_i \mid \|\boldsymbol{q}_i\|_2 \leq \sqrt{1/\lambda}\}$ to solve the private profile $\bar{\boldsymbol{q}}_i$. Repeating the above steps through all item in $\boldsymbol{I}$, the private matrix $\bar{\boldsymbol{Q}}$ is obtained.

### C. Spectral Truncation

Complementary loss can reduce noise, but since the noise is directly added to the loss function, it may still cause the loss function to be unbounded, leading to invalid solutions. As shown in Eq. (22), our method would transform the objective function into a quadratic polynomial $\bar{l}(\boldsymbol{q}_i)$, after which it injects noise into the coefficients of $\bar{l}(\boldsymbol{q}_i)$ to ensure privacy. Let $\bar{l}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T\boldsymbol{M}\boldsymbol{q}_i + \boldsymbol{q}_i^T\boldsymbol{m} + \tau$, where $\boldsymbol{M} = \boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0\boldsymbol{G}_{\tilde{\boldsymbol{U}}_i} + \tilde{\boldsymbol{B}}$ and $\boldsymbol{m} = 2\boldsymbol{g}_{\boldsymbol{U}_i} + \boldsymbol{b}$, to ensure that $\bar{l}(\boldsymbol{q}_i)$ is bounded after noise injection, it suffices to make $\boldsymbol{M}$ symmetric and positive definite [30].

The symmetry of $\boldsymbol{M}$ is ensured by the addition of symmetric noise. However, ensuring that $\boldsymbol{M}$ is positive definite remains a significant challenge. To the best of our knowledge, no existing method transforms the coefficient matrix of an MF model into a positive definite matrix in a differentially private manner. According to [30], a matrix is not positive definite if and only if at least one eigenvalue of $\boldsymbol{M}$ is non-positive. Building on this insight, we propose the spectral truncation method.

Let $\boldsymbol{M} = \boldsymbol{V}\Lambda\boldsymbol{V}^T$, where $\boldsymbol{V}$ is a $d \times d$ matrix where each row is an eigenvector of $\boldsymbol{M}$, and $\Lambda$ is a diagonal matrix where the $i$-th diagonal element is the eigenvalue $\lambda_i$ of $\boldsymbol{M}$ corresponding to the eigenvector in the $i$-th row of $\boldsymbol{V}$. Accordingly,

$$\bar{l}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T\left(\boldsymbol{V}\Lambda\boldsymbol{V}^T\right)\boldsymbol{q}_i + \boldsymbol{q}_i^T\boldsymbol{m} + \tau.$$

To ensure all the eigenvalues are positive, we replace all negative or zero eigenvalues with a small positive value (typically a small constant $\xi$) as follows:

$$\lambda_i' = \max(\lambda_i, \xi).$$

This ensures the new eigenvalue matrix $\Lambda'$ will contain only positive eigenvalues. After removing these elements, the adjusted matrix $\boldsymbol{M}'$ can be reconstructed:

$$\boldsymbol{M}' = \boldsymbol{V}\Lambda'\boldsymbol{V}^T.$$

This matrix $\boldsymbol{M}'$ is positive definite. The noisy objective function then becomes

$$\bar{l}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T\boldsymbol{M}'\boldsymbol{q}_i + \boldsymbol{q}_i^T\boldsymbol{m} + \tau. \tag{24}$$

In summary, we eliminate non-positive elements from $\Lambda$ to guarantee boundedness of the loss function. As these non-private elements are mostly noise, their removal minimally affects information. Thus, the objective function in Eq. (24) can still yield an accurate profile matrix. Moreover, this removal does not compromise $\varepsilon$-differential privacy, as it relies solely on the differentially private $\boldsymbol{M}$ rather than the input database.

### D. Importance Sampling

In the implicit feedback setting, the number of private objects increases due to the inclusion of unobserved data, resulting in a large DP noise scale. This section introduces an importance sampling method, which allows DPIMF to be trained on only the sampled subset of private objects without sacrificing too much utility. We first present our theoretical contribution on the privacy amplification via importance sampling. Then, we describe how to construct sampling distributions that achieve a given DP guarantee with minimal sample size.

*Importance Sampler:* We begin by introducing the sampling strategy we use. It is a weighted version of the Poisson sampling [31], which we refer to as *Poisson importance sampling*.

*Definition 4 (Importance Sample):* Given the dataset of user-item pairs $\Omega = \boldsymbol{U} \times \boldsymbol{I}$, the procedure Importance Sampler $\mathcal{S}_W$ outputs a subset of the data $\{(w_{ui}, u, i) \mid \gamma_{ui} = 1\}$ by sampling $\gamma_{ui} \sim \textbf{Ber}(1/w_{ui})$, where **Ber** represents the Bernoulli distribution.

Here, the importance sampler $\mathcal{S}_W$ returns a weighted user-item pair. The varying sampling weights lead to different privacy losses for each data. To capture this privacy heterogeneity, we define the function $\phi : [1, \infty) \times \Omega \to \mathbb{R}^+$ to represent the privacy loss profile of each user-item pair. In practice and our experiments, $\phi$ can be computed based on public information or fixed local model parameters without requiring access to the input database. However, an total privacy loss is required since our algorithm works in compliance with the rigorous $\varepsilon$-differential privacy. This total privacy loss can be the upper bound of $\phi$ across all user-item pairs. Based on this notion in place, we have the following results.

*Theorem 2 (Privacy Amplification):* For the importance sampler $\mathcal{S}_W$ and any $\varepsilon$-DP mechanism $\mathcal{M}$, $\widehat{\mathcal{M}} = \mathcal{M} \circ S_q$ satisfies $\varepsilon'$-DP where

$$\varepsilon' = \ln\left(1 + \frac{1}{w_{ui}}\left(e^{\phi_{max}} - 1\right)\right), \tag{25}$$

where $\phi_{max}$ is the maximum of privacy loss profile $\phi$.

*Optimal Importance Sampling:* We now describe how to construct a sampling distribution that achieves a given privacy guarantee with minimal sample size. The motivation for this is two-fold. First, by imposing a overall privacy loss as a constraint, we can ensure that the importance sampling mechanism satisfies $\varepsilon$-DP by design. Second, the sample size of private objects is a primary indicator of the noise scale and the efficiency of the mechanism. Minimizing the expected sample size subject to a given $\varepsilon'$-DP constraint can be described as the following optimization problem.

*Problem 1 (Privacy-utility optimal sampling):* For a privacy loss profile $\phi$: $[1, \infty) \times \Omega \to \mathbb{R}_{\geq 0}$, a target privacy guarantee

$\varepsilon'$, and a data set of user-item pairs $\Omega$, we define the privacy-optimal sampling problem as

$$\underset{\boldsymbol{W}\in\mathbb{R}^{|\boldsymbol{U}|\times|\boldsymbol{I}|}}{\arg\min} \quad \sum\nolimits_{(u,i)\in\Omega} \frac{1}{w_{ui}} \tag{26a}$$

$$\text{s.t.} \quad \ln\left(1+\frac{e^{\phi(w_{ui},u,i)}-1}{w_{ui}}\right) \le \varepsilon', \quad \forall u,i, \tag{26b}$$

$$w_{ui} \ge 1, \qquad \forall u,i. \tag{26c}$$

The constraint in Eq. (42b) captures the requirement that $\phi$ should be bounded by $\varepsilon'$ for all $(u,i)\in\Omega$, and the constraint in Eq. (42c) ensures that $1/w_{ui}$ is a probability. This problem is guaranteed to have a unique solution if $\phi$ is convex. This problem can be solved by the off-the-shelf methods for convex problem. We defer the proof and the algorithm for solving this problem to the section E and section F of the appendix.

### *E. Implementation of Full-fledged DPIMF*

Based on the above three strategies, we show the implementation details of the full-fledged DPIMF. We first provide the process for solving the private item profile matrix, as shown in Algorithm 1. Then we show how to extend the algorithm to solve the private user profile matrix.

It starts by initializing the profile matrix and sensitivities (Lines 1-2). Here, we uniformly sample random values between 0 and 1 by following the approach used in existing literature [15]. To reduce the computational overhead, we pre-calculate $\boldsymbol{G}_U$, which will be reused in the iterative process (Line 3). Next, the privacy profile is initialized to compute the importance weight by solving an optimization problem that is detailed in Section IV-D (Lines 5-6). As described in Section III, we compute the coefficients of the loss for adding noise. And the data points are weighted according to $\boldsymbol{W}$ (Line 9). Next, Laplacian noise is sampled and added to the coefficients. Spectral truncation is then applied to the noisy coefficient matrix to construct the final private loss (Lines 10-13). The optimal private profile vector is solved within a convex set (Line 14).

Alogrithm 2 is for solving item profile matrix. Now we discuss how to extend it to user profile matrix. When solving a user's profile vector $\boldsymbol{p}_u$, we focus on the items that $u$ has interacted with, denoted as $\boldsymbol{I}_u$. In this case, the input $\boldsymbol{P}$ and $\{\boldsymbol{U}_i \mid i\in\boldsymbol{I}\}$ should be replaced by $\boldsymbol{Q}$ and $\{\boldsymbol{I}_u \mid u\in\boldsymbol{U}\}$, respectively. The importance sampling steps (Lines 5-6) remain unchanged since $\boldsymbol{W}$ is solved by an independent problem. To build the private loss of $\boldsymbol{p}_u$, we change all $\boldsymbol{U}_i$ and $\boldsymbol{p}_u$ in the algorithm (Lines 9-13) to $\boldsymbol{I}_u$ and $\boldsymbol{q}_i$, respectively. Then we can derive $\bar{\boldsymbol{p}}_u$ using the same solution as for $\bar{\boldsymbol{q}}_i$ (Line 14).

## V. Theoretic Analyses

In this section, we establish privacy and utility guarantees of the proposed DPIMF methods.

### *A. Privacy Analysis*

*Theorem 3:* DPIMF satisfies $\varepsilon$-DP.

*Proof:* We provide a simplified proof here, with the full proof available in the section A of appendix. Based on the sensitivity analysis in Equations 17, 20, and 19, it is straightforward to show that solving $\bar{\boldsymbol{q}}_i$ for the private loss $\bar{l}(\boldsymbol{q}_i)$ in Equation (23) for an arbitrary item $i$ satisfies $\varepsilon$-DP. Furthermore, in the local system, since the data for each item and user is disjoint, Theorem 1 (i.e., the parallel composition property of DP) implies that solving the entire profile matrix also satisfies $\varepsilon$-DP. ■

**Algorithm 2:** Full-fledged DPIMF (Solving Private Item Profile Matrix)

**Input :** The local user profile matrix $\boldsymbol{P}$, the user set $\boldsymbol{U}$, the item set $\boldsymbol{I}$, the set of users interacted with each item $\{\boldsymbol{U}_i \mid i\in\boldsymbol{I}\}$, $\alpha_0$, the privacy budgets $\varepsilon_1,\varepsilon_2,\varepsilon_3$.
**Output:** $\bar{\boldsymbol{Q}}$

1 Randomly initialize $\bar{\boldsymbol{Q}}$ such that each element $q_{ij}$ is randomly drawn from $[0,1]$.
2 Compute sensitivities: $\Delta_1$, $\Delta_2$, $\Delta_3$ via Eq. (17), Eq. (20) and Eq. (19).
3 Pre-compute: $\boldsymbol{G}_{\boldsymbol{U}}=\sum_{u\in\boldsymbol{U}} \boldsymbol{p}_u\otimes\boldsymbol{p}_u$.
4 **// Solve the weights for importance sampling**.
5 Initialize privacy profile $\phi$
6 Solve the optimal $\boldsymbol{W}$ by minimizing Eq. (41).
7 **for** *each* $i\in\boldsymbol{I}$ **do**
8 **// Build the private complementary loss with $\boldsymbol{W}$.**
9 Compute the coefficients of complementary loss:
$\boldsymbol{G}_{\boldsymbol{U}_i}=\sum_{u\in\boldsymbol{U}_i} w_{ui}(\boldsymbol{p}_u\otimes\boldsymbol{p}_u)$,
$\boldsymbol{A}_1=2\sum_{u\in\boldsymbol{U}_i} w_{ui}\boldsymbol{p}_u$,
$\boldsymbol{A}_2=\boldsymbol{G}_{\boldsymbol{U}_i}+\alpha_0(\boldsymbol{G}_{\boldsymbol{U}}-\boldsymbol{G}_{\boldsymbol{U}_i})+\lambda(|\boldsymbol{U}_i|+\alpha_0|\tilde{\boldsymbol{U}}_i|)\boldsymbol{E}$.
10 Sample $\boldsymbol{b}\sim Lap\left(\frac{\Delta_1}{\varepsilon_1}\right)^d$,
$\boldsymbol{B}\sim Lap\left(\frac{\Delta_2}{\varepsilon_2}\right)^{d\times d}$, $\eta\sim Lap\left(\frac{\Delta_3}{\varepsilon_3}\right)$
11 Compute symmetric noise: $\tilde{\boldsymbol{B}}=\text{triu}(\boldsymbol{B})+\text{tril}_{-1}(\boldsymbol{B}^T)$.
12 Perform Spectral Truncation on $(\boldsymbol{A}_2+\lambda\eta\boldsymbol{E}+\boldsymbol{B})$, then get the positive definite matrix $\boldsymbol{M}'$
13 Derive the private loss:
$\bar{L}(\boldsymbol{q}_i)=\boldsymbol{q}_i^T\boldsymbol{M}'\boldsymbol{q}_i-\boldsymbol{q}_i^T(\boldsymbol{A}_1+\boldsymbol{b})$.
14 Solve $\bar{\boldsymbol{q}}_i=\underset{\boldsymbol{q}_i:\|\boldsymbol{q}_i\|_2\le\sqrt{1/\lambda}}{\text{argmin}}\bar{L}(\boldsymbol{q}_i)$.
15 **return** $\bar{\boldsymbol{Q}}$.

### *B. Utility Analysis*

The utility guarantee of the proposed DPIMF methods are theoretically analyzed in this section. To bound the sensitivity and improve utility, we modify the loss function of Eq. (8) to the form in Eq. (15). The effect of such modification is bounded, and this fact is shown by the theorem below.

*Theorem 4:* Let $N=\sqrt{1/\lambda}, c=\max_{u\in\boldsymbol{U},j\in\{1,2,\ldots,d\}}|p_{uj}|$. For the the redesigned loss function $L(\boldsymbol{q}_i)$ of Eq. (15), and the original loss function $L^{\text{O}}(\boldsymbol{q}_i)$ of Eq. (8), let $\boldsymbol{q}_i^*=\arg\min L(\boldsymbol{q}_i)$, $\boldsymbol{q}_i'=\arg\min L^{\text{O}}(\boldsymbol{q}_i)$, then we have:

$$L(\boldsymbol{q}_i^*)-L^{\text{O}}(\boldsymbol{q}_i')\le\alpha_0|\boldsymbol{U}_i|N^2(\lambda+2dc^2). \tag{27}$$

*Proof:* Please refer to section 4 of the appendix. ■

Theorem 4 implies that the minimizer $\boldsymbol{q}_i^*$ of the redesigned loss function is always bounded to the minimizer of the original loss, and they can be arbitrarily close to each other under proper setting of parameters $\alpha_0$, $\lambda$ and $d$. The empirical results in Section VI further corroborate that the impact of the modification on utility is negligible compared to the impact of the perturbation introduced by DP.

To inspect the effectiveness of each technique proposed in Section IV, we denote $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ and $\mathcal{M}_4$ as four strategies we proposed to inspect each method's contribution for utility improvement. In particular, the strategy $\mathcal{M}_1$ represents the scheme proposed in Section IV where the private loss is defined as Eq. (13). The strategies $\mathcal{M}_2, \mathcal{M}_3$ and $\mathcal{M}_4$ represent the utility-enhanced strategies proposed in Section IV. The difference among them lies only in the perturbation techniques. In $\mathcal{M}_2$, $\Delta_2$ is computed as Eq. (18), and a noise matrix of i.i.d. entries is added to the second-order term in the loss. In $\mathcal{M}_3$, $\Delta_2$ is computed as Eq. (20), and a symmetric noise matrix is sampled for perturbation. The method $\mathcal{M}_4$ represents our assumed optimal algorithm. It implements $\mathcal{M}_3$ with the presumed optimal settings, specifically setting $\alpha_0$ to 1 and utilizing importance sampling after spectral truncation. For these four different strategies, we have the following theorem.

*Theorem 5:* Let $N = \sqrt{1/\lambda}$, $\varepsilon_1 = \beta_1\varepsilon, \varepsilon_2 = \beta_2\varepsilon, c = \max_{u,j} |p_{uj}|$ and $\mathcal{S}_W$ be the importance sampler solved in Eq. (41). Given the non-private loss $L(\boldsymbol{q}_i)$ of Eq. (15) and its minimizer $\boldsymbol{q}_i^*$. Let $\bar{\boldsymbol{q}}_i^{(1)}$, $\bar{\boldsymbol{q}}_i^{(2)}$, $\bar{\boldsymbol{q}}_i^{(3)}$, be the minimizer of the private loss $\bar{l}(\boldsymbol{q}_i)$ of Eq. (23) with strategies $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$, respectively. Let $\bar{l}_{\mathcal{S}_W}(\boldsymbol{q}_i)$ be the loss with importance sampling, and $\bar{\boldsymbol{q}}_i^{(4)}$ be its minimizer. Then, the utility of the profile vectors satisfies

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(1)}) - L(\boldsymbol{q}_i^*)] \leq \frac{\sqrt{2}cdN\left[(N + \frac{2\sqrt{d}}{d})(d+1)^2\right]}{\varepsilon}, \quad (28)$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(2)}) - L(\boldsymbol{q}_i^*)] \leq \frac{\sqrt{2}cdN\left[\frac{Nd^2(1-\alpha_0)}{\beta_2} + \frac{4\sqrt{d}}{\beta_1}\right]}{\varepsilon}, \quad (29)$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(3)}) - L(\boldsymbol{q}_i^*)] \leq \frac{\sqrt{2}cdN\left[\frac{Nd(1+d)(1-\alpha_0)}{2\beta_2} + \frac{4\sqrt{d}}{\beta_1}\right]}{\varepsilon}, \quad (30)$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(4)}) - L(\boldsymbol{q}_i^*)] \leq \frac{4\sqrt{2d}cdN}{\varepsilon}. \quad (31)$$

*Proof:* Please refer to the section C of the appendix. ■

The above theorem bounds the empirical risk of the four strategies for the proposed differentially private IMF. A higher bound indicates a larger loss in the utility of the strategy. Denote the above error bounds for $\bar{\boldsymbol{q}}_i^{(1)}, \bar{\boldsymbol{q}}_i^{(2)}, \bar{\boldsymbol{q}}_i^{(3)}$ and $\bar{\boldsymbol{q}}_i^{(4)}$ as $\gamma_1, \gamma_2, \gamma_3, \gamma_4$. The relationship among the these bounds are concluded in the following theorem as a corollary of Theorem 5.

*Corollary 1:* Given $\beta_1, \beta_2 \in [0, 1]$, $\gamma_1 > \gamma_2$ when $\alpha_0 \geq 1 - \beta_2$ and $d \geq \sqrt{1/\beta_1} - 1$. For all $\alpha_0, \beta_1, \beta_2 \in [0, 1]$ and $d \in \mathbb{N}^+$, we have $\gamma_2 \geq \gamma_3 \geq \gamma_4$ and $\gamma_1 > \gamma_4$.

*Proof:* Please refer to section D of the appendix. ■

The Corollary 1 proves that $\mathcal{M}_2$ enjoys better utility than $\mathcal{M}_1$ with a proper setting of $\alpha_0$. It demonstrates the effectiveness of our modification to the loss function for enhancing utility. The condition $\alpha_0$ needs to meet to guarantee the improvement can be understood intuitively. For example, a large $\alpha_0$ is required to limit the noise when a small proportion of privacy budget is allocated to the second-order term. Moreover, the improvement from $\mathcal{M}_2$ to $\mathcal{M}_3$ is attributed to the technique of symmetric noise matrix. Moreover, Corollary 1 presents the optimal utility guarantee achieved is when $\alpha_0$ is set to 1. The vanished sensitivities prevent the second-order term and regularization term from perturbed, leading to a great improvement in utility. Moreover, unlike uniform sampling, importance sampling gives an unbiased estimate of the loss, thus ensuring that the improvements still hold after sampling. The above theoretical findings will be empirically validated by the experimental results presented in Section VI.

## VI. Experimental Evaluation

In this section, we first introduce our experiment setup in Section VI-A, and then present the experimental results and our analysis in Section VI-B.

### A. Experiment Setup

*1) Datasets.:* In the experiment, three real-world datasets, i.e., 10M MovieLens [1], YahooMusic [2] and Amazon [3] are used to evaluate the performance of our methods. **10M MovieLens** (denoted as ML-10M in the sequel) was collected by GroupLens through their experimental movie recommendation system and all ratings assigned by the users to movies. **YahooMusic** consists of data supplied by users during normal interaction with YahooMusic services. **Amazon** records the data in Amazon Instant Video (an online shopping website). The details about these datasets are shown in Table I.

According to our problem setting in Section II-C, the experiments will be conducted in two scenarios. (1) Different parties share the same items but have different user sets, requiring them to exchange item profile matrices. (2) Different parties share the same users but have different item sets, in which the user profile matrices will be exchanged. For the former scenario, we split the datasets into $s$ disjoint sub-datasets by users, such that the ratings of each user can appear in only one dataset. For the latter, we split the datasets by items in the same way. Each sub-dataset simulates the local dataset for a party.

TABLE I: The datasets used in our experiments.

| Property | ML-10M | YahooMusic | Amazon |
|---|---|---|---|
| Users | 71567 | 8089 | 5130 |
| Items | 10681 | 1000 | 1685 |
| Density | 4.3% | 3.4% | 0.7% |
| Avg. #ratings per user | 97 | 33 | 12 |
| Avg. #ratings per item | 40.5 | 270 | 36 |

*2) Evaluation Indicator:* We adopt the widely used leave-one-out evaluation, where the latest interaction of each user is held out for prediction. We train all models on the remaining data and generate ranked recommendation list. For recommendation effectiveness, we mainly consider two metrics, namely Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). HR measures whether the test item appears in the target user's recommendation list, and NDCG measures the ranked position of the hit. HR is defined as follows:

[1] 1.http://www.grouplens.org.

[2] 2.http://research.yahoo.com/AcademicRelations.

[3] 3.https://jmcauley.ucsd.edu/data/amazon/

$$\text{HR@k} = \frac{\sum_{u \in \boldsymbol{U}} |I_p(u)@k \cap I_a(u)|}{\sum_{u \in \boldsymbol{U}} |I_a(u)|}, \tag{32}$$

where $I_p(u)@k$ is the set of top k items in the ranked recommendation list. The symbol $I_a(u)$ represents the set of items the user $u$ has interacted with. NDCG is given by

$$\text{NDCG@k} = \frac{1}{|U|} \sum_{u \in U} \frac{\text{DCG}_\text{u}@\text{k}}{\text{IDCG}_\text{u}@\text{k}}, \tag{33}$$

$$\text{DCG}_\text{u}@\text{k} = \sum_{\text{idx}=1}^{k} \frac{2^{rel_{\text{idx}}} - 1}{\log_2(\text{idx}+1)}, \tag{34}$$

where $rel_{\text{idx}} \in \{0, 1\}$ indicates whether there is an interaction between user $u$ and the idx-th item in the ranked list, and $\text{IDCG}_\text{u}@\text{k}$ is the ideal $\text{DCG}_\text{u}@\text{k}$ computed on the recommendation list sorted by $rel_{\text{idx}}$ in descending order. The larger values of HR and NDCG, the better recommendation quality.

*3) Competitors:* We compare our schemes with the ground truth results (i.e., recommender system without privacy protection) and some state-of-the-art differentially private collaborative filtering techniques for implicit data, listed as follows. leftmargin=*

- Ground Truth (GT) [7]: This method uses alternating least squares to solve the IMF problem without considering user privacy. It optimizes Eq. (1) by alternately fixing $\boldsymbol{P}$ or $\boldsymbol{Q}$ and solves the other. Since iALS is one of the most commonly used methods to solve IMF problem, we regard this as the ground truth for comparison.
- DPMF [16]: Differentially Private Matrix Factorization for explicit feedback uses an objective perturbation mechanism.
- DPLCF [19]: This method protects users' implicit data by introducing random flipping technique based on nearest neighbors. Then the server publishes a sanitized similarity matrix, and each user generates recommendation results locally using that similarity matrix.
- LDPICF [20]: Similar to DPLCF, this method is also based on nearest neighbors. It uses random flipping method to perturb user data and reconstruct the relationship between item pairs using two frequency estimation techniques.
- DPIMF$_{\text{str}}$: The strawman DPIMF proposed in Section III;
- DPIMF$_{\text{com}}$: The complementary loss-based DPIMF proposed in Section IV without adding symmetric noise;
- DPIMF$_{\text{sym}}$: The complementary loss-based method proposed in Section IV with symmetric noise;
- DPIMF$_{\text{opt}}$: DPIMF$_{\text{sym}}$ with optimal setting (i.e., $\alpha_0 = 1$).
- DPIMF$_{\text{opt}}$-IS: DPIMF$_{\text{opt}}$ with importance sampling.

*4) Parameter Setting:* For the setting of hyper-parameters, the number of sub-datasets $s$ is set to 10. For all MF-based methods, the total and local iterations of model learning are set to 100 and 20, respectively. This means each party queries the local dataset 2000 times during the federated learning. The parameters of MF-based methods (i.e., including DPMF and our DPIMF solutions) are empirically set according to optimal results of hyperparameter searching (where we repeat each combination of possible hyperparameter values 10 times across all datasets) of different datasets involved. Specifically, the regularization parameters $\lambda$ is set to $0.07$ for ML-10M, and $1.0$ for YahooMusic and Amazon, as ML-10M is denser and thus requires less regularization to fit its richer interactions. The number of latent factor $d$ is set to 20, 16 and 8 for ML-10M, YahooMusic and Amazon dataset, respectively. For two neighbor-based methods (i.e., DPLCF and LDPICF), we set the number of neighbors to 100 according to the best results of parameter tuning.

### B. Experimental Results and Analysis

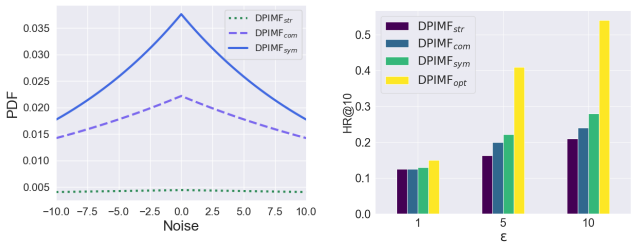
*1) The Effect of Complementary Loss:* In this section, we evaluate the effectiveness of the proposed complementary loss in Section IV by comparing the recommendation performance of DPIMF$_{\text{str}}$, DPIMF$_{\text{com}}$, DPIMF$_{\text{sym}}$ and DPIMF$_{\text{opt}}$.

We first assess the effectiveness of complementary loss in limiting the magnitude of noises added to the loss function. We record the distributions of the noises in the second-order coefficients $\tilde{\boldsymbol{B}}$ (or $\boldsymbol{B}$) in DPIMF$_{\text{str}}$, DPIMF$_{\text{com}}$, DPIMF$_{\text{sym}}$, respectively. For DPIMF$_{\text{com}}$, DPIMF$_{\text{sym}}$, we set $\alpha_0 = 0.8$, $\varepsilon = 0.1, \beta_1 = 0.1, \beta_2 = 0.8$ and $\beta_3 = 0.1$. The distributions in ML-10M, YahooMusic and Amazon datasets are reported in Figures 5(a), (c) and (e). It is clearly shown in the figure that the noise distribution of DPIMF$_{\text{sym}}$ is sharper than other two schemes, while the curve of DPIMF$_{\text{str}}$ is the flattest. Such shapes convey that the variance of the noise in DPIMF$_{\text{sym}}$ is the lowest. And the variance of the noise in DPIMF$_{\text{str}}$ is higher than that in DPIMF$_{\text{com}}$. The differences of noise distributions indicate that the magnitude of the noise added to DPIMF$_{\text{sym}}$ is lower than the other two schemes, and in turn it enjoys an improvement in recommendation accuracy.

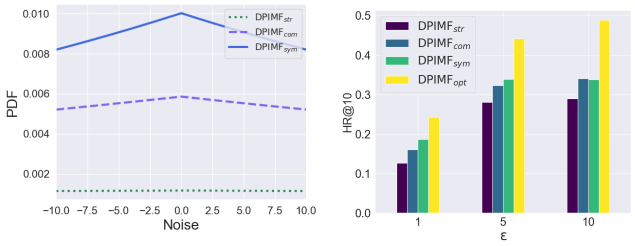
In order to validate the above claim, we evaluate the HRs of the recommendation lists generated by the three schemes on ML-10M, YahooMusic and Amazon datasets, respectively. For DPIMF$_{\text{com}}$ and DPIMF$_{\text{sym}}$, $\alpha_0$ is set to $0.8$. With varying privacy budget $\varepsilon$ from 1 to 10, the test results are reported in Figure 5 (b), (d) and (f). Overall, the utility of the recommendations improves as the privacy budget increases. This is consistent with the property of DP.

It is obvious that DPIMF$_{\text{str}}$ suffers the largest utility loss. When $\varepsilon > 1$, the other three schemes consistently outperform DPIMF$_{\text{str}}$ in HR across all datasets. For example, on the Amazon dataset, the HR value of DPIMF$_{\text{str}}$ is $7\%, 13\%, 31\%$ lower than than of DPIMF$_{\text{com}}$, DPIMF$_{\text{sym}}$ and DPIMF$_{\text{opt}}$, respectively. This validates the claim that the scheme injected noise with largest variance performs worst. The improvements also demonstrate the effectiveness of the offset technique for boosting recommendation accuracy.

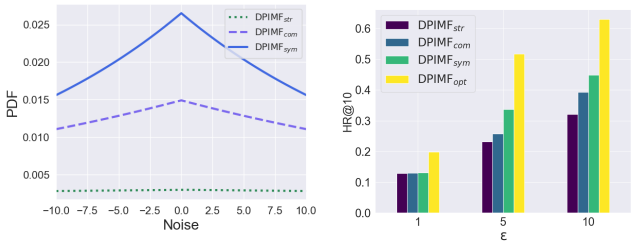
The figures also show that DPIMF$_{\text{sym}}$ is more accurate than DPIMF$_{\text{com}}$ in most cases across all datasets. For example, in Figures 5 (b), (d) and (f), we observe that within the privacy budget range of $[1, 10]$, the HRs of DPIMF$_{\text{sym}}$ is on average $2.2\%$ higher than DPIMF$_{\text{com}}$ on ML-10M. The results validate the theoretical conclusion that DPIMF$_{\text{sym}}$ reduces unnecessary noises by using the symmetric strategy. Summing up the upper triangular entries rather than the full entries in the coefficient




ML-10M, noise distribution ML-10M, HR@10




YahooMusic, noise distribution YahooMusic, HR@10




Amazon, noise distribution Amazon, HR@10

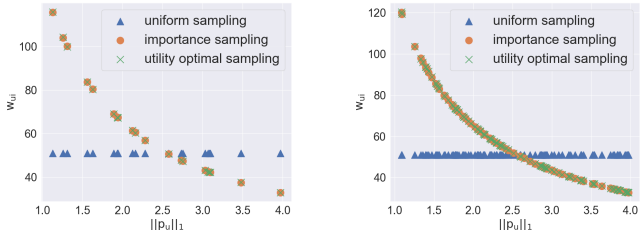
Fig. 5: Performance of DPIMF methods on noise distribution and HR@10.

matrix brings DPIMF$_{sym}$ a lower sensitivity, which in turn enhances the utility of recommendations. For the performance on the YahooMusic and Amazon datasets shown in Figures 5 (d) and (f), DPIMF$_{com}$ outperforms DPIMF$_{sym}$ in some cases (e.g., the HR on Amazon when $\varepsilon = 5$). This is because the noise variances of the two methods are both relatively large on YahooMusic and Amazon, as shown in Figures 5(c) and (e).

Note that DPIMF$_{opt}$ always performs the best, achieving significantly higher HRs under different privacy budgets across all datasets. On Amazon dataset with $\varepsilon = 10$, the HR of DPIMF$_{opt}$ reaches $0.64$, which is $32\%, 25\%$ and $19\%$ higher than that of DPIMF$_{str}$, DPIMF$_{com}$ and DPIMF$_{sym}$, respectively. The results are consistent with the conclusions of utility analysis. By setting $\alpha_0$ to 1, the effect of the change in data is completely offset. The ramification of this setting is that the second-order coefficients are free from perturbation. It not only reduces the injected noise, but also saves the privacy budget. More privacy budget is utilized to inject less noise into the objective function. As a consequence, the utility of recommendation increases dramatically, which explains the superiority of DPIMF$_{opt}$ shown in Figures 5 (b), (d) and (f).

*2) The Effect of Importance Sampling:* To study the effect of importance sampling, we compare the importance sampling to uniform sampling in terms of privacy and utility at a fixed sample size. To test the effect of noise, we define $\phi$ as the perturbed term of first-order coefficients in the private loss, i.e., $\boldsymbol{\mathcal{A}}_1(\mathcal{D}) = 2\sum_{u\in\boldsymbol{U}_i} w_{ui}\boldsymbol{p}_u + \boldsymbol{b}$. When the data $\mathcal{D} = \boldsymbol{U}_i$ is sampled by Poisson importance sampler $\mathcal{S}_W$, we can compute the variance of the mechanism $\boldsymbol{\mathcal{A}}\circ\mathcal{S}_W$ on different dimension.

Let $m$ be the sample size, we compare our importance sampling with two strategies, the uniform sampling, and utility-optimal sampling the solution of minimizing the variance of $\boldsymbol{\mathcal{A}}_1(\mathcal{D})$.We test 2000 users' data from ML-10M where the number of latent factor is set to 20. We visualize the importance weights $w_{ui}$ for each sampling strategy. For this, we fix a target sample size at $m \in \{20, 100\}$ and compute the weights $w_{ui}$ for each strategy within the sample size. The results are shown in Figure 6. As shown in the Figures, weights solved by our importance sampling and the utility-optimal weights are very close. The results indicate that privacy and utility are closely aligned objectives when using importance sampling, and uniform sampling is significantly suboptimal, particularly in the case of IMF. The overall improvement on model utility will be demonstrated in the following results.




(a) m=20 (b) m=100

Fig. 6: Comparison of sampling strategies.

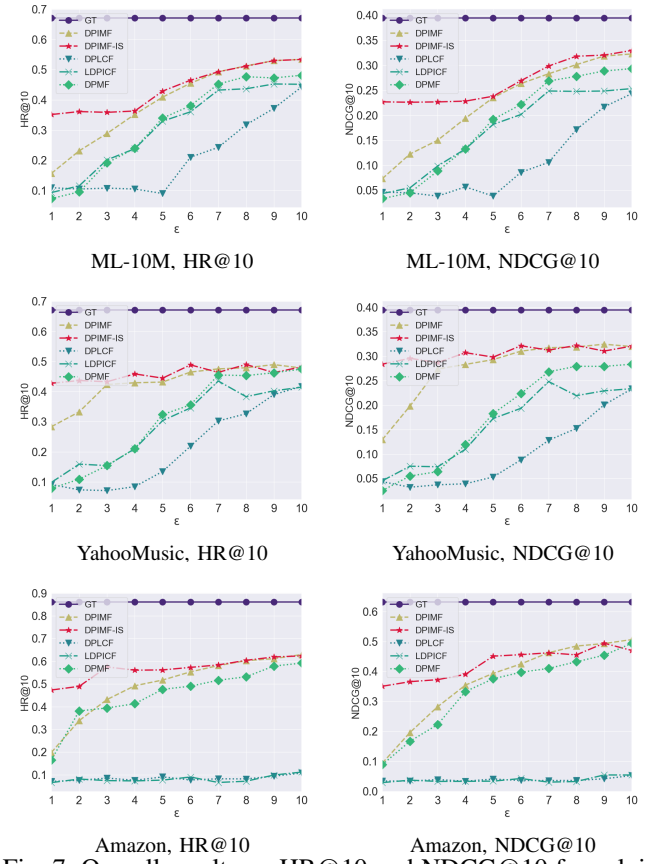
*3) Comparing with other schemes:* In order to demonstrate the effectiveness of our methods, we compare the recommendation quality of our DPIMP methods (i.e., DPIMF$_{opt}$ and DPIMF$_{opt}$-IS, which are referred to as DPIMF and DPIMF-IS in the sequel for simplicity) with the non-private scheme (i.e., GT) and other differentially private methods for the IMF problem. With varying privacy budgets from 1 to 10, the HRs and NDCGs of the competitors on ML-10M, YahooMusic and Amazon datasets are reported in Figure 7.

Overall, the performance curves of all private schemes converge to the ground truth (i.e., GT) with the increasing privacy budget $\varepsilon$. This is consistent with the feature of DP that utility will be improved when weakening privacy assurance. The recommendation accuracy of DPIMF is always better than that of DPLCF and LDPICF on all datasets. Furthermore, DPIMF-IS consistently outperforms the other private methods over all privacy levels. On ML-10M, as shown in Figures 7(a) and (b), the HR of DPIMF-IS is on average $18.3\%$ and $8.3\%$ higher than DPLCF and LDPICF. In NDCG, DPIMF-IS is $12.1\%$ and $5.4\%$ higher than DPLCF and LDPICF. The improvements become more significant on YahooMusic as shown in Figures 7(c) and (d). When $\varepsilon = 3$, the HR and NDCG of DPIMF-IS are $26\%, 20\%$ higher than DPLCF, and $35\%, 22\%$ higher than LDPICF, respectively. Furthermore, this accuracy gap even reaches $50\%$ around both in HR and NDCG on Amazon dataset.

The higher utility of DPIMF methods over DPLCF and LDPICF first validate the better generalization ability of matrix factorization-based methods than KNN-based methods. Second, the DPIMF methods not only benefit the expressiveness of matrix factorization models, but also ensure the model utility with privacy guarantee. By carefully designing the loss function, the sensitivity and the added noise, our DPIMF methods effectively limit the perturbation error introduced into the model. In particular, comparing with DPIMF, DPIMF-IS achieves a more significant utility improvement under stricter privacy requirements (e.g., $\varepsilon < 5$). This aligns with the privacy amplification conclusion proposed in Section V-B, as the effect of privacy amplification becomes more pronounced when the overall privacy budget is smaller, leading to reduced noise and better utility.

Moreover, the KNN-based methods suffer from the difficulty in finding proper neighbors in data with high sparsity level. This is demonstrated in Figures 7(e) and (f). The figures tell that the HRs and NDCGs of DPLCF and LDPICF are just around $0.1$ on Amazon, the sparsest dataset of the three. The results demonstrate that the DPLCF and LDPICF are far from practical on extremely sparse datasets. On the contrary, the HR and NDCG of DPIMF on the same dataset increase steadily when increasing the privacy budget. When $\varepsilon = 10$, the HR and NDCG of DPIMF exceed $0.6$ and $0.5$, respectively. The utility loss is reduced to around $0.2$. This result verifies the advantage of DPIMF in dealing with highly sparse datasets.

To verify the effectiveness of DPIMF in the scenario of learning private user profile matrix, we conducted the tests on sub-datasets split by items using the same hyper-parameters settings. The results are shown in Figure 8. Compared to the case of learning item profiles, on the ML-10M dataset, all algorithms showed significant improvement, where DPIMF-IS approaches the ground truth more quickly when $\varepsilon > 5$. This improvement is attributed to the higher per-user interactions as shown in Table I, allowing each party to exploit more information to counter the noise during local training. On the YahooMusic dataset, as the data per user is much sparser than the data per item, causing greater fluctuations in the algorithm's performance. This is due to the limited data amplifying the impact of noise, which is similarly observed in the results on Amazon dataset.




ML-10M, HR@10 ML-10M, NDCG@10

YahooMusic, HR@10 YahooMusic, NDCG@10

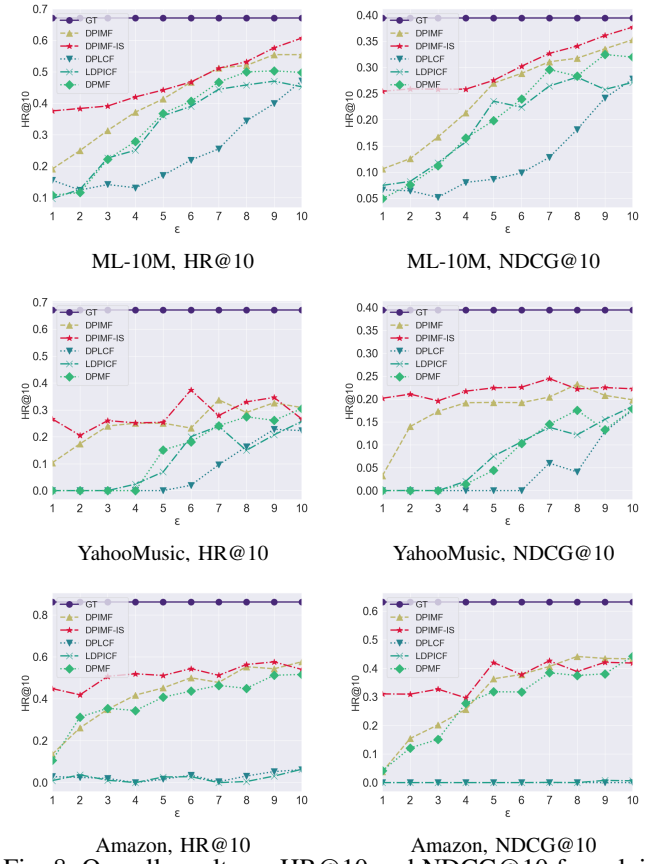
Amazon, HR@10 Amazon, NDCG@10

Fig. 7: Overall results on HR@10 and NDCG@10 for solving the item profile matrix.

## VII. Related Work

Differential Privacy, as a rigorous privacy notion which guarantees the privacy of any user participating in a statistical computation, has been widely applied to recommender systems. The seminal work by McSherry et al. [32] first introduced DP to the domain of collaborative filtering through perturbing the item covariance matrix. The studies in [33], [34] incorporated DP into KNN-based methods, where the perturbation is injected into the similarity computation and the score prediction phases. The schemes in [28], [35] employed the mechanism proposed in [31], which learns the item profile matrix while preserving DP via stochastic gradient Langevin dynamics. Considering the privacy of a user's overall data, Jain et al. [36] adopted a relaxed differential privacy, i.e., joint differential privacy (JDP), and proposed a private Frank-Wolfe algorithm in which the noise is added into the feasible gradient descent direction. Following this work, Chien et al. [17] designed MF methods preserving JDP based on the alternating least squares (ALS) algorithm. Berlioz et al. [15] designed a framework for DP matrix factorization (MF), where the privacy mechanisms are classified into input, in-process and output perturbations according to the process of MF. The schemes in [16] proposed to perturb the objective function of the MF problem, and learn the differentially private MF model based on the perturbed objective.

Most of the privacy-preserving studies focus on the case of explicit user preference. However, a privacy-preserving approach for implicit feedback with good utility is still lacking. From the privacy perspective, implicit preferences that record which items the user is interested in also reveal a lot about the user's privacy [37], [38]. Indeed, Weinsberg et al. [39] validated that the implicit preference data are highly relevant with sensitive attributes such as gender and age. By exploiting merely a small part of implicit data, Narayanan et al. [40] successfully de-anonymized the user records in Netflix. As far as we know, only the works [20], [19] attempted to introduce DP to protect the implicit data, and their schemes are based on the KNN model. These methods first apply the




ML-10M, HR@10 ML-10M, NDCG@10

YahooMusic, HR@10 YahooMusic, NDCG@10

Amazon, HR@10 Amazon, NDCG@10

Fig. 8: Overall results on HR@10 and NDCG@10 for solving the user profile matrix.

random bit flipping technique to obfuscate the implicit data in compliance with DP, then estimate cardinality to extract similar items from the obfuscated data for predicting user preference. However, the KNN model cannot provide sufficient generalization capability due to the fact that the implicit dataset is extremely sparse and its dimension is typically very high [7].

## VIII. Conclusion

This paper studies the problem of differentially private Implicit Matrix Factorization (DPIMF). We introduce objective perturbation to perturb the loss function, and the MF model is learned based on the perturbed loss, ensuring compliance with differential privacy (DP). To address utility degradation, we redesign the loss function and adopt an importance sampling strategy, effectively reducing the noise scale and improving the utility of IMF. We provide theoretical proofs of the utility guarantees for the proposed schemes. Experimental results on three benchmark datasets demonstrate that the proposed strategies effectively improve model utility, and our proposed solution achieves a good trade-off between privacy levels and recommendation quality.

## Appendix

*Lemma 1:* Given $L(\boldsymbol{q}_i)$ of Eq. (14), $L^{\mathrm{O}}(\boldsymbol{q}_i)$ of Eq. (7), let $\boldsymbol{q}_i^* = \arg\min L(\boldsymbol{q}_i)$, $\boldsymbol{q}_i' = \arg\min L'(\boldsymbol{q}_i)$, then $\|\boldsymbol{q}_i^*\|_2 \leq \sqrt{\frac{1}{\lambda}}$, and $\|\boldsymbol{q}_i'\|_2 \leq \sqrt{\frac{1}{\lambda}}$.

*Proof:* Since the loss term $L_I(\boldsymbol{q}_i)$ in $L(\boldsymbol{q}_i)$ is nonnegative, we have $L(\boldsymbol{q}_i) \geq R(\boldsymbol{q}_i)$. The optimal solution $\boldsymbol{q}_i^*$ satisfies $L(\boldsymbol{q}_i^*) \leq L(\boldsymbol{q}_i)$ for any $\boldsymbol{q}_i$. Let $\boldsymbol{q}_i = \mathbf{0}$, we can derive $L(\boldsymbol{q}_i^*) \leq L(\mathbf{0}) = |\boldsymbol{U}_i|$. Since $\alpha_0 \in [1, 0]$, we have

$$\lambda\left(|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i|\right)\|\boldsymbol{q}_i^*\|_2^2 \leq |\boldsymbol{U}_i|$$

$$\|\boldsymbol{q}_i^*\|_2 \leq \sqrt{\frac{|\boldsymbol{U}_i|}{\lambda\left(|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i|\right)}} \leq \sqrt{\frac{1}{\lambda}}$$

Similarly, we can derive $\|\boldsymbol{q}_i'\|_2 \leq \sqrt{\frac{1}{\lambda}}$, which completes the proof. ■

*Lemma 2:* Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^T$, where $x_j \sim Lap\left(\frac{\Delta}{\epsilon}\right)$, $j \in \{1, 2, \cdots, d\}$. Then $\mathbb{E}\left[\|\boldsymbol{x}\|_2\right] \leq \frac{\sqrt{2d}\Delta}{\varepsilon}$.

*Proof:* According to the Jensen's inequality, we have

$$\mathbb{E}\left[\|\boldsymbol{x}\|_2\right] = \mathbb{E}\left[\sqrt{\sum_j x_j^2}\right] \leq \sqrt{\mathbb{E}\left[\sum_j x_j^2\right]}.$$

Since the expectation of the laplace distribution is zero, we have

$$\mathbb{E}\left[\sum_j x_j^2\right] = \sum_j \mathbb{E}\left[x_j^2\right] = \sum_j 2\frac{\Delta^2}{\varepsilon^2} = \frac{2d\Delta^2}{\varepsilon^2}.$$

Therefore, we have $\mathbb{E}\left[\|\boldsymbol{x}\|_2\right] \leq \frac{\sqrt{2d}\Delta}{\varepsilon}$. ■

### A. Proof of Theorem 3

*Proof:* According to our design logic, spectral truncation and importance sampling do not require further access to the input database. Thus, based on the post-processing theorem, demonstrating that solving $\bar{\boldsymbol{q}}_i$ for the private loss $\bar{l}(\boldsymbol{q}_i)$ in Eq. (22) satisfies $\varepsilon$-DP for any arbitrary item $i$ is sufficient to establish that Algorithm 2 satisfies DP. Let $\boldsymbol{T} \in \mathbb{R}^{d\times d}, \boldsymbol{t} \in \mathbb{R}^d$, and $\tau \in \mathbb{R}$ be the perturbed coefficients of the second and first-order terms in $\bar{l}_I(\boldsymbol{q}_i)$, and the perturbed weight of the regularization term $\bar{R}(\boldsymbol{q}_i)$ of Eq. (21), respectively. Define dataset $D_i$ that records both the positive and non-positive users for $i$, then its neighboring dataset $D_i'$ is obtained from $D_i$ by moving one user from the positive set to the non-positive set (and vice versa). For a privacy budget $\varepsilon$ divided into $\varepsilon_1, \varepsilon_2, \varepsilon_3$, we have

$$\begin{aligned}
&\frac{\Pr\{\bar{l}(\boldsymbol{q}_i) \mid D_i\}}{\Pr\{\bar{l}(\boldsymbol{q}_i) \mid D_i'\}} \\
&= \frac{\Pr\{\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \alpha_0\sum_{u\in\tilde{\boldsymbol{U}}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \tilde{\boldsymbol{B}} = \boldsymbol{T}\}}{\Pr\{\sum_{u\in\boldsymbol{U}_i'}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \alpha_0\sum_{u\in\tilde{\boldsymbol{U}}_i'}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \tilde{\boldsymbol{B}}' = \boldsymbol{T}\}} \\
&\cdot\frac{\Pr\{2\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u + \boldsymbol{b} = \boldsymbol{t}\}\cdot\Pr\{|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i| + \eta = \tau\}}{\Pr\{2\sum_{u\in\boldsymbol{U}_i'}\boldsymbol{p}_u + \boldsymbol{b}' = \boldsymbol{t}\}\cdot\Pr\{|\boldsymbol{U}_i'| + \alpha_0|\tilde{\boldsymbol{U}}_i'| + \eta' = \tau\}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\prod_{j\le l}\Pr\left\{\tilde{B}_{jl} = T_{jl} - (\boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0 \boldsymbol{G}_{\tilde{\boldsymbol{U}}_i})_{jl}\right\}}{\prod_{j\le l}\Pr\left\{\begin{array}{l}\tilde{B}'_{jl} = T_{jl} - (\boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0 \boldsymbol{G}_{\tilde{\boldsymbol{U}}_i})_{jl} - \\ (1-\alpha_0)(\boldsymbol{p}_v \otimes \boldsymbol{p}_v)_{jl}\end{array}\right\}} \\
&\cdot \frac{\prod_{j=1}^{d}\Pr\left\{b_j = t_j - (2\sum_{u\in\boldsymbol{U}'_i}\boldsymbol{p}_u)_j\right\}}{\prod_{j=1}^{d}\Pr\left\{b_j = t_j - (2\sum_{u\in\boldsymbol{U}'_i}\boldsymbol{p}_u)_j - 2p_{vj}\right\}} \\
&\cdot \frac{\Pr\left\{\eta = \tau - |\boldsymbol{U}_i| - \alpha_0|\tilde{\boldsymbol{U}}_i|\right\}}{\Pr\left\{\eta = \tau - |\boldsymbol{U}_i| - \alpha_0|\tilde{\boldsymbol{U}}_i| - (1-\alpha_0)\right\}} \\
&\le \exp\left(\frac{\varepsilon_2 \max_{u\in\boldsymbol{U}}((1-\alpha_0)\|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}})}{\Delta_2}\right) \\
&\quad \cdot \exp\left(\frac{\varepsilon_1 \max_{u\in\boldsymbol{U}}(2\|\boldsymbol{p}_u\|_1)}{\Delta_1}\right)\exp\left(\frac{(1-\alpha_0)\varepsilon_3}{\Delta_3}\right) \\
&= \exp(\varepsilon_1+\varepsilon_2+\varepsilon_3) = \exp(\varepsilon),
\end{aligned}
$$

where $\boldsymbol{G}_{\boldsymbol{U}_i}$=$\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u$, $\boldsymbol{G}_{\tilde{\boldsymbol{U}}_i}$=$\sum_{u\in\tilde{\boldsymbol{U}}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u$. Thus solving $\bar{\boldsymbol{q}}_i$ satisfies $\varepsilon$-DP. Since the ratings of each item are disjoint, according to Theorem 1(i.e., parallel composition of DP), we conclude that DPIMF satisfies $\varepsilon$-DP. ■

### B. Proof of Theorem 4

*Proof:* Let $\boldsymbol{W} = \sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u$, we have:

$$
\begin{aligned}
&L'(\boldsymbol{q}_i^*) - L(\boldsymbol{q}'_i) \qquad (35)\\
=&L'(\boldsymbol{q}_i^*) - L(\boldsymbol{q}'_i) + L(\boldsymbol{q}'_i) - L(\boldsymbol{q}_i^*) + L(\boldsymbol{q}_i^*) - L'(\boldsymbol{q}'_i)\\
\le&L'(\boldsymbol{q}_i^*) + L(\boldsymbol{q}'_i) - L(\boldsymbol{q}_i^*) - L'(\boldsymbol{q}'_i)\\
=&(\boldsymbol{q}_i^* - \boldsymbol{q}'_i)^T\boldsymbol{W}(\boldsymbol{q}_i^* - \boldsymbol{q}'_i) + (\lambda\alpha_0|\boldsymbol{U}_i|)(\|\boldsymbol{q}_i^*\|_2^2 - \|\boldsymbol{q}'_i\|_2^2)
\end{aligned}
$$

According to Lemma 1, $\|\boldsymbol{q}_i^*\|_2 \le N$ and $\|\boldsymbol{q}'_i\|_2 \le N$. Let $\boldsymbol{z} = \boldsymbol{q}_i^* - \boldsymbol{q}'_i$, we have:

$$
\begin{aligned}
\boldsymbol{z}_i^T\boldsymbol{W}\boldsymbol{z}_i &= \sum_{(j,l)} W_{jl}z_{ij}z_{il}\\
&\le \left(\sum_{(j,l)} W_{jl}^2\right)^{\frac{1}{2}}\left(\sum_{(j,l)} z_{ij}^2 z_{il}^2\right)^{\frac{1}{2}}\\
&\le \left(\alpha_0 d|\boldsymbol{U}_i|c^2\right)\left(\sum_{(j,l)} z_{ij}^2 z_{il}^2\right)^{\frac{1}{2}}\\
&= \left(\alpha_0 d|\boldsymbol{U}_i|c^2\right)\left[\left(\sum_{j} z_{ij}^2\right)^2\right]^{\frac{1}{2}}\\
&= \left(\alpha_0 d|\boldsymbol{U}_i|c^2\right)\|\boldsymbol{z}\|_2^2\\
&\le \left(\alpha_0 d|\boldsymbol{U}_i|c^2\right)(\|\boldsymbol{q}_i^*\|_2^2 + \|\boldsymbol{q}'_i\|_2^2)\\
&\le 2\alpha_0 d|\boldsymbol{U}_i|c^2N^2
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
L'(\boldsymbol{q}_i^*) - L(\boldsymbol{q}'_i) &\le 2\alpha_0 d|\boldsymbol{U}_i|c^2N^2 + \lambda\alpha_0|\boldsymbol{U}_i|N^2\\
&= \alpha_0|\boldsymbol{U}_i|N^2(\lambda + 2dc^2)
\end{aligned}
$$

■

### C. Proof of Theorem 5

*Proof:* For the ease of readability, we repeat the private loss function $\bar{l}(\boldsymbol{q}_i)$ under $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ as follows:

$$
\begin{aligned}
\bar{l}^{(1)}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T(&\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \alpha_0\sum_{u\in\boldsymbol{U}}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \boldsymbol{B})\boldsymbol{q}_i - \qquad (36)\\
&\boldsymbol{q}_i^T(2\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u + \boldsymbol{b}) + \lambda\left(|\boldsymbol{U}_i| + \alpha_0|\boldsymbol{U}| + \eta\right)\|\boldsymbol{q}_i\|_2^2,\\
\bar{l}^{(2)}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T[&\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \alpha_0\sum_{u\in\tilde{\boldsymbol{U}}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \boldsymbol{B} + \qquad (37)\\
&\lambda(|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i| + \eta)\boldsymbol{E}]\boldsymbol{q}_i - \boldsymbol{q}_i^T(2\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u + \boldsymbol{b}),\\
\bar{l}^{(3)}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T[&\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \alpha_0\sum_{u\in\tilde{\boldsymbol{U}}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \tilde{\boldsymbol{B}} + \qquad (38)\\
&\lambda(|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i| + \eta)\boldsymbol{E}]\boldsymbol{q}_i - \boldsymbol{q}_i^T(2\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u + \boldsymbol{b}),\\
\bar{l}^{(4)}(\boldsymbol{q}_i) = &\boldsymbol{q}_i^T\boldsymbol{M}'\boldsymbol{q}_i - \boldsymbol{q}_i^T(\boldsymbol{A}_1 + \boldsymbol{b}) \qquad (39)
\end{aligned}
$$

Where $\boldsymbol{M}'$ and $\boldsymbol{A}_1$ are defined as in Algorithm 2. For an arbitraty minimizer $\bar{\boldsymbol{q}}_i^{(k)}$ of private loss $\bar{l}^{(k)}(\boldsymbol{q}_i)$ over the convex set $C = \{\boldsymbol{q}_i \mid \|\boldsymbol{q}_i\|_2 \le N\}$, $k = 1, 2, 3$, we have

$$
\begin{aligned}
L(\bar{\boldsymbol{q}}_i) - L(\boldsymbol{q}_i^*) &= L(\bar{\boldsymbol{q}}_i) - \bar{l}(\boldsymbol{q}_i^*) + \bar{l}(\boldsymbol{q}_i^*) - \bar{l}(\bar{\boldsymbol{q}}_i) + \bar{l}(\bar{\boldsymbol{q}}_i) - L(\boldsymbol{q}_i^*)\\
&\le \bar{l}(\boldsymbol{q}_i^*) - L(\boldsymbol{q}_i^*) + L(\bar{\boldsymbol{q}}_i) - \bar{l}(\bar{\boldsymbol{q}}_i)\\
&= (\boldsymbol{q}_i^{*T}\hat{\boldsymbol{B}}\boldsymbol{q}_i^* - 2\boldsymbol{q}_i^{*T}\boldsymbol{b} + \lambda\eta\|\boldsymbol{q}_i^*\|_2^2) +\\
&\quad (-\bar{\boldsymbol{q}}_i^T\hat{\boldsymbol{B}}\bar{\boldsymbol{q}}_i + 2\bar{\boldsymbol{q}}_i^T\boldsymbol{b} - \lambda\eta\|\bar{\boldsymbol{q}}_i\|_2^2).
\end{aligned}
$$

where we denote $\bar{\boldsymbol{q}}_i$ as $\bar{\boldsymbol{q}}_i^{(k)}$ for conciseness, and $\hat{\boldsymbol{B}}$ as either the i.i.d. noise matrix $\boldsymbol{B}$ or symmetric noise matrix $\tilde{\boldsymbol{B}}$. Since $\|\bar{\boldsymbol{q}}_i\|_2 \le N$, according to Lemma 1, we have the following inequalities:

$$
\begin{aligned}
2\bar{\boldsymbol{q}}_i^T\boldsymbol{b} &\le 2\|\bar{\boldsymbol{q}}_i\|_2\|\boldsymbol{b}\|_2 \le 2N\|\boldsymbol{b}\|_2,\\
-\bar{\boldsymbol{q}}_i^T\hat{\boldsymbol{B}}\bar{\boldsymbol{q}}_i &= -\sum_{(j,l)}\hat{B}'_{jl}\bar{q}_{ij}\bar{q}_{il}\\
&\le \left(\sum_{(j,l)}\hat{B}'^2_{jl}\right)^{\frac{1}{2}}\left(\sum_{(j,l)}\bar{q}_{ij}^2\bar{q}_{il}^2\right)^{\frac{1}{2}}\\
&= \|\hat{\boldsymbol{B}}\|_F\left[\left(\sum_{j}\bar{q}_{ij}^2\right)^2\right]^{\frac{1}{2}}\\
&\le \|\hat{\boldsymbol{B}}\|_F N^2.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
\mathbb{E}[L(\bar{\boldsymbol{q}}_i) - L(\boldsymbol{q}_i^*)] &= \mathbb{E}[(\boldsymbol{q}_i^{*T}\hat{\boldsymbol{B}}\boldsymbol{q}_i - 2\boldsymbol{q}_i^{*T}\boldsymbol{b} + \lambda\eta\|\boldsymbol{q}_i^*\|_2^2) \qquad (40)\\
&\quad (-\bar{\boldsymbol{q}}_i^T\hat{\boldsymbol{B}}\bar{\boldsymbol{q}}_i + 2\bar{\boldsymbol{q}}_i^T\boldsymbol{b} - \lambda\eta\|\bar{\boldsymbol{q}}_i\|_2^2)]\\
&= \mathbb{E}(-\bar{\boldsymbol{q}}_i^T\hat{\boldsymbol{B}}\bar{\boldsymbol{q}}_i + 2\bar{\boldsymbol{q}}_i^T\boldsymbol{b} - \lambda\eta\|\bar{\boldsymbol{q}}_i\|_2^2)\\
&\le \mathbb{E}\left[\|\hat{\boldsymbol{B}}\|_F N^2\right] + \mathbb{E}\left[2N\|\boldsymbol{b}\|_2\right]
\end{aligned}
$$

By Lemma 2, $\bar{\boldsymbol{q}}_i^{(1)}, \bar{\boldsymbol{q}}_i^{(2)}, \bar{\boldsymbol{q}}_i^{(3)}$ satisfy

$$
\begin{aligned}
\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(1)}) - L(\boldsymbol{q}_i^*)] &\le \frac{(cd^2+2cd+1)(N^2\sqrt{2}d + 2N\sqrt{2d})}{\varepsilon}\\
&\le \frac{\sqrt{2}cdN\left[(N + \frac{2\sqrt{d}}{d})(d^2 + 2d + \frac{1}{cd})\right]}{\varepsilon}
\end{aligned}
$$

$$\leq \frac{\sqrt{2}cdN\left[(N+\frac{2\sqrt{d}}{d})(d+1)^2\right]}{\varepsilon}$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(2)}) - L(\boldsymbol{q}_i^*)] \leq \frac{N^2\sqrt{2}d\Delta_2}{\varepsilon_2} + \frac{2N\sqrt{2d}\Delta_1}{\varepsilon_1}$$
$$\leq \frac{\sqrt{2}N^2cd^3(1-\alpha_0)}{\varepsilon_2} + \frac{4Ncd\sqrt{2d}}{\varepsilon_1}$$
$$= \frac{\sqrt{2}cdN\left[\varepsilon_1 Nd^2(1-\alpha_0) + 4\varepsilon_2\sqrt{d}\right]}{\varepsilon_1\varepsilon_2}.$$
$$= \frac{\sqrt{2}cdN\left[\frac{1}{\beta_2}Nd^2(1-\alpha_0) + \frac{4}{\beta_1}\sqrt{d}\right]}{\varepsilon}$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(3)}) - L(\boldsymbol{q}_i^*)]$$
$$\leq \frac{N^2\sqrt{2}d\Delta_2}{\varepsilon_2} + \frac{2N\sqrt{2d}\Delta_1}{\varepsilon_1}$$
$$\leq \frac{\frac{\sqrt{2}}{2}N^2cd^2(1+d)(1-\alpha_0)}{\varepsilon_2} + \frac{4Ncd\sqrt{2d}}{\varepsilon_1}$$
$$= \frac{\sqrt{2}cdN\left[\frac{1}{2}\varepsilon_1 Nd(1+d)(1-\alpha_0) + 4\varepsilon_2\sqrt{d}\right]}{\varepsilon_1\varepsilon_2}.$$
$$= \frac{\sqrt{2}cdN\left[\frac{1}{2\beta_2}Nd(1+d)(1-\alpha_0) + \frac{4}{\beta_1}\sqrt{d}\right]}{\varepsilon}$$

Similarly, we can derive the bound for $\bar{\boldsymbol{q}}_i^{(4)}$ as

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(4)}) - L(\boldsymbol{q}_i^*)] \leq \mathbb{E}\left[2N\|\boldsymbol{b}\|_2\right] \leq \frac{4\sqrt{2}cdN\sqrt{d}}{\varepsilon}$$

■

### D. Proof of Corollary 1

*Proof:* We first prove that $\gamma_1 > \gamma_2$ by presenting $(\gamma_1 - \gamma_2)$ is positive, which is defined as

$$\gamma_1 - \gamma_2 \propto N\left[d^2\left(\frac{\beta_2+\alpha_0-1}{\beta_2}\right) + 2d + 1\right] + \frac{\beta_1(d+1)^2-1}{2d\beta_1} = f(\boldsymbol{\theta}).$$

We compute the partial derivative of $f(\boldsymbol{\theta})$ w.r.t $N$ as follows

$$\frac{\partial f(\boldsymbol{\theta})}{\partial N} = d^2\left(\frac{\beta_2+\alpha_0-1}{\beta_2}\right) + 2d + 1,$$

where $\boldsymbol{\theta}$ denotes the parameters including $\alpha_0, d, N, \beta_1$ and $\beta_2$. By ensuring $\alpha_0 \geq 1-\beta_2$, we have $\frac{\partial f(\boldsymbol{\theta})}{\partial N} \geq 0$, which ensures $f$ is monotonically increasing with the value of $N \in (0, +\infty)$. Let $d \geq \sqrt{1/\beta_1} - 1$, we have $\lim_{N\to 0^+} f(\boldsymbol{\theta}) \geq 0$. Thus, the function $f(\boldsymbol{\theta})$ is strictly positive in the domain of $N \in (0, +\infty)$, which in turn proves $\gamma_1 > \gamma_2$. Since $\frac{1}{2}d(1+d) \leq d^2$ for all $d \in \mathbb{N}^+$, we have $\gamma_2 \geq \gamma_3$, and it is clear that $\gamma_3 \geq \gamma_4$ for all $\beta_1 \in [0, 1]$.

For all $\alpha_0, \beta_1, \beta_1 \in [0, 1]$, $d \in \mathbb{N}^+$ and for all $N, c \in (0, +\infty)$

$$\gamma_1 - \gamma_4 \propto N(d+1) + \frac{2\sqrt{d}(d^2+1)}{d} > 0$$

Thus, we have $\gamma_1 > \gamma_4$, which completes the proof. ■

### E. Proof of the unique solution of Problem 1

**Problem 1.** (Privacy-utility optimal sampling) For a privacy loss profile $\phi\colon\ [1, \infty) \times \Omega \to \mathbb{R}_{\geq 0}$, a target privacy guarantee $\varepsilon'$, and a data set of user-item pairs $\Omega$, we define the privacy-optimal sampling problem as

$$\underset{\boldsymbol{W}\in\mathbb{R}^{|\boldsymbol{U}|\times|\boldsymbol{I}|}}{\arg\min} \quad \sum\nolimits_{(u,i)\in\Omega} \frac{1}{w_{ui}} \tag{41a}$$
$$\text{s.t.} \quad \ln\left(1 + \frac{1}{w_{ui}}\left(e^{\phi(w_{ui},u,i)} - 1\right)\right) \leq \varepsilon', \quad \forall u, i, \tag{41b}$$
$$w_{ui} \geq 1, \quad \forall u, i. \tag{41c}$$

*Theorem 6:* Let $\varepsilon' \geq \phi(1, u, i)$ for all $(u, i) \in \Omega$ and $\varepsilon > \ln(1 + w(\exp^{\varepsilon'} - 1))$ for all $w \geq v_{ui}$ where $v_{ui} \geq 1$. Problem 1 has a unique solution $W^*$.

*Proof:* Without loss of generality, we consider each $w_{ui}$ independently. The problem for $w_{ui}$ can be equivalently formed as

$$\underset{\boldsymbol{W}\in\mathbb{R}^{|\boldsymbol{U}|\times|\boldsymbol{I}|}}{\arg\max} \quad \sum\nolimits_{(u,i)\in\Omega} w_{ui} \tag{42a}$$
$$\text{s.t.} \quad \ln\left(1 + \frac{1}{w_{ui}}\left(e^{\phi(w_{ui},u,i)} - 1\right)\right) \leq \varepsilon', \quad \forall u, i, \tag{42b}$$
$$w_{ui} \geq 1, \quad \forall u, i. \tag{42c}$$

According to the setting we concerned, the feasible region is bounded and not empty. Since the objective is strictly monotonic, the solution must be unique. ■

**Algorithm 3:** Privacy-Utility Optimal Sampling

**Input :** The user item pairs $\Omega$, target privacy budget $\varepsilon'$, the privacy profile $\phi$, strong convexity constants $\mu_{ui}, (u, i) \in \Omega$.
**Output:** $W^*$

**for** *each* $(u, i) \in \Omega$ **do**
  $g_{ui}(w) = \frac{\exp(\phi(w,u,i))}{w} - (\exp^{\varepsilon'} - 1)$
  $v_i \leftarrow 2(\exp(\varepsilon') - \frac{\partial\phi(1,u,i)}{\partial w}\exp(\phi(1, u, i)) + 1)/\mu_{ui} + 1$
  **if** $\phi(1, u, i) = \varepsilon'\phi(1, u, i) < 0$ **then**
    $w_{ui} \leftarrow$ Bisect $g_{ui}$ with bracket $(1, v_i]$
  **else**
    $w_{ui} \leftarrow$ Bisect $g_{ui}$ with bracket $[1, v_i]$
  **end if**
**return** $W^*$.

### F. Algorithm for solving Problem 1

The Problem 1 can be solved by the bisection-based convex optimizer described in Algorithm 2. Here, the strongly convexity is defined as follows:

*Definition 5:* (Strong Convexity). Let $\mu > 0$. A differentiable function $f : \mathbf{R}^d \to \mathbf{R}$ is $\mu$-strongly convex if for all $u, v \in \mathbb{R}^d$

$$f(v) \geq f(u) + \nabla f(u)^\top (v - u) + \frac{\mu}{2}\|v - u\|_2^2 \tag{43}$$

## REFERENCES


[1] S. Kalloori and S. Klingler, "Horizontal cross-silo federated recommender systems," in *Proceedings of the 15th acm conference on recommender systems*, 2021, pp. 680–684.

[2] K. Liu, S. Hu, S. Z. Wu, and V. Smith, "On privacy and personalization in cross-silo federated learning," *Advances in neural information processing systems*, vol. 35, pp. 5925–5940, 2022.

[3] L. Yang, B. Tan, V. W. Zheng, K. Chen, and Q. Yang, "Federated recommendation systems," in *Federated Learning: Privacy and Incentive*. Springer, 2020, pp. 225–239.

[4] Z. Li, B. Ding, C. Zhang, N. Li, and J. Zhou, "Federated matrix factorization with privacy guarantee," *Proceedings of the VLDB Endowment*, vol. 15, no. 4, 2021.

[5] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[6] S. Rendle, W. Krichene, L. Zhang, and Y. Koren, "ials++: Speeding up matrix factorization with subspace optimization," *arXiv preprint arXiv:2110.14044*, 2021.

[7] ——, "Revisiting the performance of ials on item recommendation benchmarks," *arXiv preprint arXiv:2110.14037*, 2021.

[8] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," *Recommender systems handbook*, pp. 91–142, 2021.

[9] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[10] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 502–511.

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

[12] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–39, 2023.

[13] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[14] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[15] A. Berlioz, A. Friedman, M. A. Kaafar, R. Boreli, and S. Berkovsky, "Applying differential privacy to matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 107–114.

[16] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15. AAAI Press, 2015, p. 1763–1770.

[17] S. Chien, P. Jain, W. Krichene, S. Rendle, S. Song, A. Thakurta, and L. Zhang, "Private alternating least squares: Practical private matrix completion with tighter rates," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1877–1887.

[18] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

[19] C. Gao, C. Huang, D. Lin, D. Jin, and Y. Li, "Dplcf: differentially private local collaborative filtering," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 961–970.

[20] T. Guo, J. Luo, K. Dong, and M. Yang, "Locally differentially private item-based collaborative filtering," *Information Sciences*, vol. 502, pp. 229–246, 2019.

[21] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American statistical association*, vol. 60, no. 309, pp. 63–69, 1965.

[22] R. Balu and T. Furon, "Differentially private matrix factorization using sketching techniques," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 57–62.

[23] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[24] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *Fourteenth ACM conference on recommender systems*, 2020, pp. 240–248.

[25] S. Rendle, "Item recommendation from implicit feedback," *arXiv preprint arXiv:2101.08769*, 2021.

[26] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[27] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.

[28] Z. Liu, Y.-X. Wang, and A. Smola, "Fast differentially private matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 171–178.

[29] S. Zhang, L. Liu, Z. Chen, and H. Zhong, "Probabilistic matrix factorization with personalized differential privacy," *Knowledge-Based Systems*, vol. 183, p. 104864, 2019.

[30] S. Lang, *Introduction to linear algebra*. Springer Science & Business Media, 2012.

[31] Y.-X. Wang, S. Fienberg, and A. Smola, "Privacy for free: Posterior sampling and stochastic gradient monte carlo," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2493–2502.

[32] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 627–636.

[33] T. Zhu, Y. Ren, W. Zhou, J. Rong, and P. Xiong, "An effective privacy preserving algorithm for neighborhood-based collaborative filtering," *Future Generation Computer Systems*, vol. 36, pp. 142–155, 2014.

[34] J. Wang and Q. Tang, "Differentially private neighborhood-based recommender systems," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2017, pp. 459–473.

[35] J.-Y. Jiang, C.-T. Li, and S.-D. Lin, "Towards a more reliable privacy-preserving recommender system," *Information Sciences*, vol. 482, pp. 248–265, 2019.

[36] P. Jain, O. D. Thakkar, and A. Thakurta, "Differentially private matrix completion revisited," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2215–2224.

[37] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, no. 1, p. 59, 2022.

[38] T. N. T. Tran, A. Felfernig, C. Trattner, and A. Holzinger, "Recommender systems in the healthcare domain: state-of-the-art and research issues," *Journal of Intelligent Information Systems*, vol. 57, no. 1, pp. 171–201, 2021.

[39] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings," in *Proceedings of the sixth ACM conference on Recommender systems*, 2012, pp. 195–202.

[40] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 2008, pp. 111–125.