Privacy-Aware Representation Decoupling in Federated Recommendation against Attribute Inference Attacks

Xuhao Zhao, Yanmin Zhu, *Senior Member, IEEE*, Wenze Ma, Qihao Luo, Chenhao Zhai, Chunyang Wang, Jiadi Yu, *Member, IEEE*, Feilong Tang, *Senior Member, IEEE*

***Abstract*—Federated recommender systems (FedRec) aim to preserve user privacy by keeping sensitive data on client devices and sharing only model parameters with a central server. However, FedRec is still vulnerable to attribute inference attacks (AIA), where server-side adversaries exploit uploaded parameters to infer users' private attributes. Existing approaches face a sub-optimal privacy-performance trade-off. Privacy-focused methods mask attribute-related features in representations to protect sensitive information, but degrade recommendation accuracy. In contrast, performance-focused methods preserve accuracy by retaining these features but risk privacy leakage through uploaded representations. To balance privacy and performance, we propose PARD, a privacy-aware representation decoupling framework that explicitly decouples representations into *privacy-relevant* and *privacy-irrelevant* components. Only the privacy-irrelevant part is uploaded to the server, and the privacy-relevant part is retained locally. We introduce mutual information (MI) objectives to realize the decoupling: (1) minimizing MI between privacy-irrelevant representations and sensitive attributes to suppress leakage, and (2) maximizing MI for privacy-relevant representations to retain personalized preference signals. Since exact MI computation is intractable, we derive variational bounds and estimate them using privacy estimators under adversarial and cooperative training paradigms. Experimental results demonstrate that PARD outperforms state-of-the-art methods in both recommendation accuracy and privacy preservation. The code is available at https://github.com/XuHao-bit/PARD**

***Index Terms*—Federated Recommendation, Privacy-Preserving.**

I. Introduction

RECOMMENDER systems (RSs) are vital in modern online services, offering personalized content and alleviating information overload [1]. They support intelligent service optimization and enhance user experience through adaptive content delivery. Recent advances in deep learning have significantly improved recommendation quality by capturing complex user behavior patterns [2]–[6]. In addition, incorporating auxiliary signals such as user attributes [7]–[9], social relationships [10], [11], and user locations [12]–[14] has further improved recommendation performance.

Xuhao Zhao, Yanmin Zhu, Wenze Ma, Qihao Luo, Jiadi Yu, Feilong Tang are with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China (e-mail: zhaoxuhao@sjtu.edu.cn; yzhu@cs.sjtu.edu.cn; mawenze991226@sjtu.edu.cn; sjtu18815199753@sjtu.edu.cn; jiadiyu@cs.sjtu.edu.cn; tang-fl@cs.sjtu.edu.cn).

Chenhao Zhai is with Shenzhen International Graduate School, Tsinghua University, Shenzhen, China (e-mail: dch23@mails.tsinghua.edu.cn).

Chunyang Wang is with School of Data Science and Engineering, East China Normal University, Shanghai, China (e-mail: cywang@dase.ecnu.edu.cn).

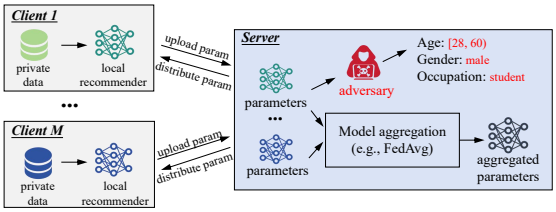


Fig. 1. A typical federated recommender system (FedRec) and its vulnerability to attribute inference attacks.

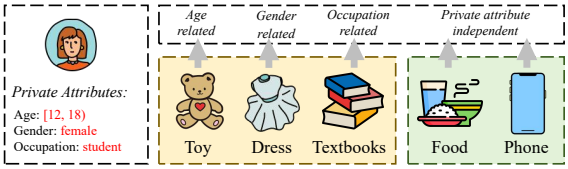


Fig. 2. The privacy-coupled user preferences reveal sensitive attributes through interaction patterns.

However, most RSs adopt centralized paradigms that require collecting user data on remote servers. This centralized setup raises serious privacy concerns, especially as user awareness grows and strict regulations like GDPR [15] come into force. To mitigate such risks [16]–[19], federated recommender systems (FedRec) have been proposed, inspired by Federated Learning (FL) [20]. As illustrated in Fig. 1, FedRec allows each client (user) to keep their private data locally while collaboratively training a recommender model. The typical model architecture comprises a user encoder, an item encoder for learning representations, and a prediction layer for rating prediction. During training, clients update model parameters based on their private data. Then, these parameters are periodically uploaded to a central server, which aggregates them using algorithms like FedAvg [20] and distributes the updated model back to clients. This decentralized setup helps to reduce the risk of direct data exposure [20].

Despite this, FedRec remains vulnerable to privacy risks, particularly attribute inference attacks (AIA) [21], [22]. In such attacks, as illustrated in Fig. 1, a malicious server-side adversary analyzes the uploaded model parameters (e.g., user and item encoders) to infer users' sensitive attributes [21],

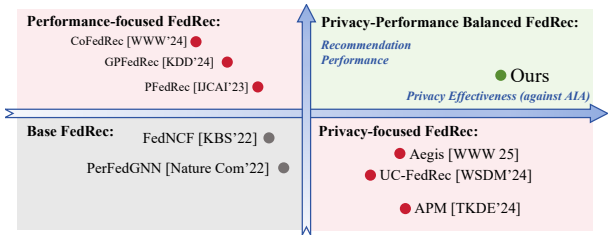


Fig. 3. Performance-privacy trade-off between FedRec methods, where our approach achieves high recommendation performance and privacy effectiveness.

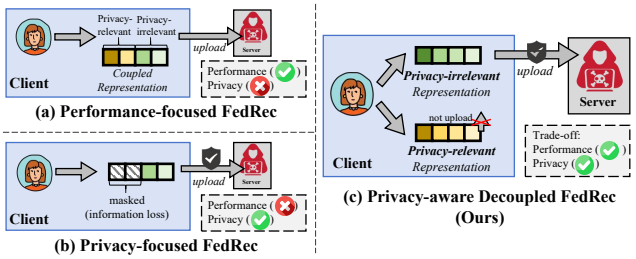


Fig. 4. Method comparison between existing FedRec methods and our approach.

[23]. This threat stems from the fact that model parameters inherently reflect user preferences, which are often coupled with private information. Fig. 2 exemplifies how privacy is coupled in preferences. Assume a user regards age, gender, and occupation as private. Her preferences for toys, dresses, and textbooks may suggest that the user is younger, female, and a student. In contrast, preferences for food and phone might not reflect such sensitive attributes. Notably, these privacy-coupled preferences are implicitly encoded in learned representations, even if private attributes are excluded from training data [21]. Consequently, adversaries can train inference models to predict sensitive attributes from uploaded parameters, making AIA a critical privacy concern in FedRec design.

Existing FedRec methods struggle to balance recommendation performance and privacy protection against AIA, as shown in Fig. 3. Current approaches generally fall into three categories: **Base FedRec** [24], [25] directly adapts traditional RSs to FL settings without explicit privacy or performance optimizations. **Performance-focused FedRec** [26]–[28] learns privacy-coupled representations and employs personalization techniques to enhance recommendation performance. However, they retain privacy-relevant features (i.e., features related to sensitive user attributes) in the uploaded representations, making them vulnerable to AIA (Fig. 4 (a)). **Privacy-focused FedRec** [21], [29], [30] attempts to defend against AIA by masking privacy-relevant information via differential privacy [31] or post-training unlearning [32]. While effective in reducing attribute leakage, these methods often degrade recommendation performance by discarding useful privacy-relevant preference features (Fig. 4 (b)).

To balance privacy and performance, we propose a Privacy-Aware Representation Decoupling approach, termed PARD (Fig. 4 (c)). The core idea is to explicitly decouple representations into two components: *privacy-relevant* representations, which encode sensitive attribute-related features, and *privacy-irrelevant* representations, which exclude such information. Only the privacy-irrelevant component is uploaded to the server, effectively mitigating the risk of AIA. The privacy-relevant component is retained locally to capture attribute-driven preference signals. This dual-component design preserves complete preference information for local prediction while preventing attribute inference on the server.

An intuitive approach is to learn two types of representations: one for server aggregation and the other for local retention. However, since privacy information is inherently coupled with user preferences, merely separating representations cannot guarantee privacy decoupling. To address this, we employ mutual information (MI) estimation to realize the decoupling process. First, we minimize MI between privacy-irrelevant representations and attributes to reduce privacy leakage. Second, we maximize MI between privacy-relevant representations and attributes for accurate recommendation. Moreover, computing mutual information (MI) between high-dimensional representations and attributes is intractable [33]. To overcome this, we derive variational bounds and estimate them using specially designed privacy estimators. This formulation allows the decoupling process to be achieved through the adversarial and cooperative learning paradigms within federated training. Experiments on real-world datasets validate that our method achieves strong privacy protection while maintaining good recommendation performance.

The main contributions of this work are as follows:

- To the best of our knowledge, this is the first privacy-aware decoupling framework for FedRec against AIA, which decouples representations into privacy-relevant and privacy-irrelevant components. This design achieves both recommendation accuracy and privacy protection.
- We propose mutual information (MI) objectives to guide privacy-aware representation decoupling and derive tractable variational bounds. We then estimate these bounds using specially designed privacy estimators under adversarial and cooperative training paradigms.
- Extensive experiments on three real-world datasets demonstrate that PARD outperforms several state-of-the-art privacy-preserving FedRec models in both privacy preservation and recommendation performance.

The remainder of this paper is organized as follows. Section II formally defines the problem. Section III details our proposed PARD framework. The experimental results and analyses are presented in Section IV. Section V discusses relevant literature and distinguishes our contributions from prior work. Finally, Section VI concludes our findings and future directions.

II. Preliminaries

In this section, we first illustrate the paradigm of Federated Recommender Systems (FedRec). Then, we introduce the details of attribute inference attacks and defense. Finally, we illustrate the mutual information estimation. For clarity, we summarize important notations in Table I.

A. Federated Recommender Systems

We consider a FedRec with M clients and a central server. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ denote the sets of users and items, respectively. Each client corresponds to a single user u and maintains u's interaction set S_u. The S_u consists of records in the form of tuples (x_u, x_i, r_{ui}), where $x_u \in \mathbb{R}^M$ and $x_i \in \mathbb{R}^N$ are the one-hot encodings of the user and item IDs, and $r_{ui} = 1$ indicates an interaction (e.g., a click or purchase), while $r_{ui} = 0$ indicates no interaction.

The server coordinates model training by collecting and aggregating model parameters from selected clients. However, it may passively act as an adversary [20], [34], attempting to infer users' private attributes from the uploaded parameters.

Recommender Model. The recommender model on each client consists of: (1) A user encoder f_{θ_u} typically implemented as an embedding layer $\theta_u = \boldsymbol{E_u} \in \mathbb{R}^{1\times d}$, where d denotes the embedding dimension. (2) An item encoder f_{θ_i} implemented as an embedding layer $\theta_i = \boldsymbol{E_i} \in \mathbb{R}^{N\times d}$. (3) A prediction layer h_ϕ typically implemented as a 3-layer MLP model. The recommender model predicts ratings using:

$$\hat{r}_{ui} = h_\phi([\boldsymbol{z_u}, \boldsymbol{z_i}]), \tag{1}$$

$$\boldsymbol{z_u} = f_{\theta_u}(x_u), \tag{2}$$

$$\boldsymbol{z_i} = f_{\theta_i}(x_i), \tag{3}$$

where $\boldsymbol{z_u} \in \mathbb{R}^d$ and $\boldsymbol{z_i} \in \mathbb{R}^d$ are the user and item representations, respectively. $[\cdot,\cdot]$ denotes feature concatenation.

Client-side Training. Each client trains its local model by minimizing the following objective:

$$\min \sum_{u\in\mathcal{U}} \sum_{(x_u,x_i,r_{ui})\in S_u} \mathcal{L}(\hat{r}_{ui}, r_{ui}), \tag{4}$$

where $\mathcal{L}$ is a loss function such as BPR [35] or binary cross-entropy. After local training, each client uploads part of its model parameters (denoted as Θ_u) to the server.

Server-side Federated Aggregation. The server aggregates the uploaded parameters from a subset of clients. Let $\mathcal{U}^* \subset \mathcal{U}$ denote the set of selected users at global round r. Let $\Theta_{u,r}$ denote the parameters uploaded by client $u \in \mathcal{U}^*$ at global round r. The server computes the global model Θ_{r+1} using the FedAvg algorithm [36]:

$$\Theta_{r+1} = \frac{1}{|\mathcal{U}^*|} \sum_{u\in\mathcal{U}^*} \Theta_{u,r}. \tag{5}$$

The aggregated parameters Θ_{r+1} are then distributed to all clients for the next round of training or recommendation.

Privacy Setting. We assume each client u maintains a private attribute set $\mathcal{T}_u$ (e.g., age, gender), which is not used for predicting $\hat{r}_{ui}$ during training. Each attribute $t \in \mathcal{T}_u$ has a corresponding value $\mu_{u,t} \in \mathbb{R}^{C_t}$, where C_t is the category of attribute t. Following [21], we also assume the existence of a small set of privacy-insensitive users $\mathcal{U}_{ins}$, who voluntarily share both their model parameters and attribute values. In practice, such users commonly exist on open platforms where data sharing is opt-in for personalized services or reward programs (e.g., social media or e-commerce sites). These are used to construct a public dataset:

$$D_{pub} = \{(\Theta_u, \mu_{u,t}) | u \in \mathcal{U}_{ins}, t \in \mathcal{T}_u\}, \tag{6}$$

TABLE I
SUMMARY OF NOTATIONS

Notation	Description
$\mathcal{U}, \mathcal{I}$	User set, Item set
$\mathcal{U}^*$	Users selected for training
$\mathcal{U}_{ins}$	Privacy-insensitive users
M, N	Number of users and items
S_u	Interaction set of user u
x_u, x_i	ID of user u, item i
$r_{ui}, \hat{r}_{ui}$	True and predicted rating from u to i
$\mathcal{T}_u$	Private attribute set of user u
$t \in \mathcal{T}_u$, $\mu_{u,t}$	Attribute and its value (e.g., age, gender)
$f_{\theta_u^{re}}; f_{\theta_u^{ir}}, f_{\theta_i^{ir}}$	Privacy-relevant/irrelevant encoders
$\boldsymbol{z_u^{re}}; \boldsymbol{z_u^{ir}}, \boldsymbol{z_i^{ir}}$	Privacy-relevant/irrelevant representations
h_ϕ	Prediction layer
$g_{\psi_{u,t}^{ir}}; g_{\psi_{u,t}^{FP}}, g_{\psi_{u,t}^{IP}}$	Privacy estimators for u of t
D_{pub}	Public dataset with $\mathcal{U}_{ins}$'s data
$\Theta_{u,r}$	Uploaded parameters of user u on round r
Θ_r	Aggregated parameters on round r
$\mathcal{K}$	Adversary's knowledge
$\mathcal{A}_t$	Attack model for inferring attribute t
$\lambda_{ir}, \lambda_{re}$	Hyperparameters in loss function
α	Learning rate
R, L	Global rounds, local iterations

which is accessible to the server and all clients. This dataset can facilitate downstream tasks, including attribute inference and defense design.

B. Attribute Inference Attacks and Defense

1) Task 1 (Attributes Inference Attacks): In FedRec, an adversary aims to infer a user's private attribute value $\mu_{u,t}$ (e.g., age or gender) for each sensitive attribute $t \in \mathcal{T}_u$, based on the user's uploaded model parameters Θ_u. This constitutes an Attribute Inference Attack (AIA).

Adversary's Knowledge. We assume that the adversary on the server is honest-but-curious [20] and possesses the following prior knowledge $\mathcal{K}$: (1) The model architecture and uploaded parameters Θ_u for each client $u \in \mathcal{U}^*$; (2) A public dataset D_{pub} from privacy-insensitive users; (3) Each user's positively interacted items $\mathcal{I}'_u = \{i' | r_{u,i'} = 1\}$, inferred by analyzing the differences between client and server-side item embeddings during training. Thus, $\mathcal{K} = \{\Theta_u, D_{pub}, \mathcal{I}'_u\}$.

Adversary's Objective. The adversary constructs attribute attack classifiers $\{\mathcal{A}_t \mid t \in \mathcal{T}_u\}$ to predict $\mu_{u,t}$ using $\mathcal{K}$. Specifically, Θ_u includes user/item encoder parameters. Each attack model $\mathcal{A}_t$ takes as input: (1) The user encoder parameter $\theta_u = \boldsymbol{E_u} \in \mathbb{R}^{1\times d}$, and (2) the averaged parameters of positively interacted items $\theta^*_{u,i} = Avg(\{\boldsymbol{E_i}[i'] | i' \in \mathcal{I}'_u\}) \in \mathbb{R}^{1\times d}$, where $\boldsymbol{E_i}[i']$ denotes the parameter of item i' in item encoder. The concatenated feature is $[\theta_u, \theta^*_{u,i}]$, which encodes attribute-related preference patterns. The attack models are then trained using features extracted from public data D_{pub}. After training, the attack model is used to infer the attribute values of users $u \in \mathcal{U} \setminus \mathcal{U}_{ins}$:

$$\hat{\mu}_{u,t} = \mathcal{A}_t([\theta_u, \theta^*_{u,i}]), \tag{7}$$

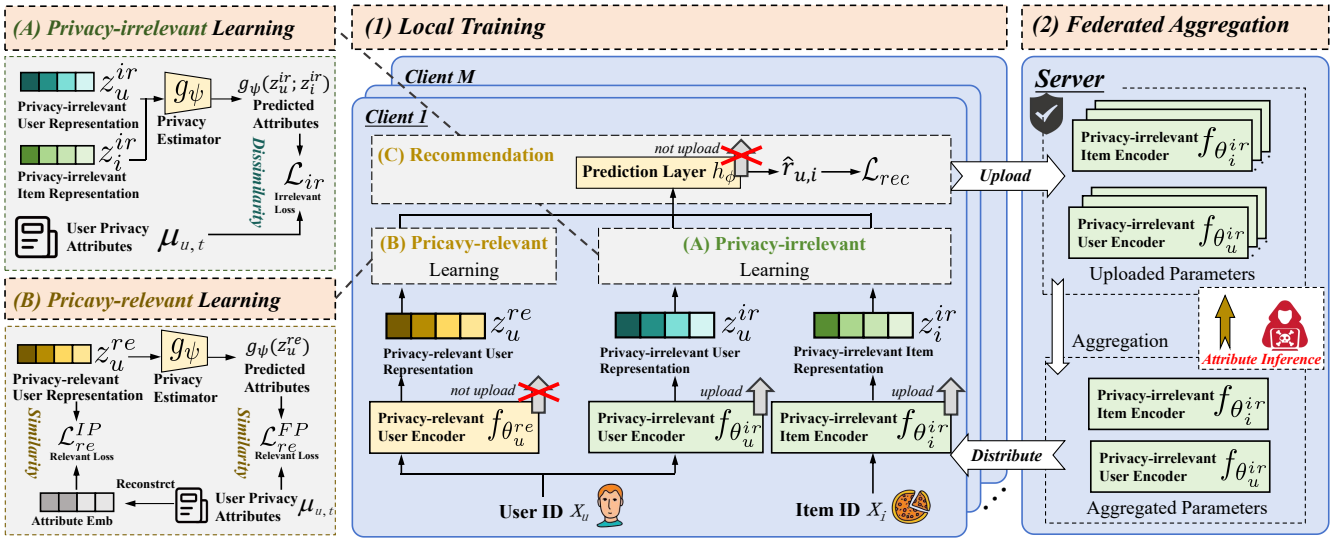


Fig. 5. Illustration of the framework PARD. First, each client decouples user representations into privacy-irrelevant and privacy-relevant components, while item representations are maintained solely in privacy-irrelevant form for efficiency. Second, clients upload only privacy-irrelevant encoders (user and item) to the server, retaining the privacy-relevant encoder and prediction layer locally. Finally, the server aggregates parameters via federated averaging and distributes updated models to clients.

where $\hat{\mu}_{u,t}$ denotes the predicted value of attribute t for user u. $\mathcal{A}_t$ is implemented via a 3-layer MLP model. This attack process is repeated for each target attribute.

2) Task 2 (Defense against AIA): In this scenario, we assume that the local client acts as the defender, applying a defense strategy during model training to mitigate AIA risks. The defense has two core objectives:

(1) Prevent attribute inference. The defender prevents the recommender model from encoding sensitive attribute information into the parameters uploaded to the server, thereby reducing the adversary's ability to perform attribute inference.

(2) Preserve the recommendation performance. The defender ensures that the recommendation quality is not significantly compromised while protecting attributes from inference.

C. Mutual Information Estimation

Mutual Information (MI) is a fundamental concept in information theory that quantifies the amount of shared information between two random variables. Formally, the MI between a representation $\boldsymbol{z}$ and a private attribute μ is defined as:

$$I(\boldsymbol{z}; \mu) = H(\mu) - H(\mu \mid \boldsymbol{z}),$$

where $H(\mu)$ is the entropy of μ, and $H(\mu \mid \boldsymbol{z})$ is the conditional entropy of μ given $\boldsymbol{z}$. The conditional entropy $H(\mu \mid \boldsymbol{z})$ quantifies how much uncertainty about the private attribute μ remains after observing $\boldsymbol{z}$. A lower $H(\mu \mid \boldsymbol{z})$ indicates that $\boldsymbol{z}$ provides more information about μ, reducing this uncertainty. In other words, a higher MI $I(\boldsymbol{z}; \mu)$ indicates that the representation $\boldsymbol{z}$ contains more information about the private attribute μ.

In our context, we decouple representations $\boldsymbol{z}$ into privacy-irrelevant $\boldsymbol{z}_{ir}$ and privacy-relevant $\boldsymbol{z}^{re}$, generated by encoders $f_{\theta^{ir}}$ and $f_{\theta^{re}}$. We aim to minimize the MI $I(\boldsymbol{z}_{ir}; \mu_{u,t})$ to eliminate privacy information in θ^{ir}, while maximizing MI $I(\boldsymbol{z}^{re}; \mu_{u,t})$ to retain attribute-related preferences in θ^{re}. The detailed design is provided in Section III-A and III-B.

III. Proposed Approach

As illustrated in Figure 5, PARD consists of M clients and a central server, and operates in two main phases: local training and federated aggregation.

In the *local training* phase, each client jointly optimizes three objectives: (1) A *privacy-irrelevant* representation learning objective (Section III-A) that learns user and item representations with minimized attribute leakage, supporting privacy-preserving aggregation on the server. (2) A *privacy-relevant* representation learning objective (Section III-B) applied only to user representations, which encourages them to retain sensitive attribute signals locally. Item representations are excluded to reduce memory usage. (3) A *recommendation* objective (Section III-C) that combines both types of representations for accurate interaction prediction.

In the *federated aggregation* phase (Section III-D), each client uploads only the privacy-irrelevant encoders, while the privacy-relevant user encoder and the prediction layer remain local. The server aggregates the uploaded parameters and then distributes the updated model to all clients.

A. Privacy-irrelevant Representation Learning

To suppress the attribute $\mu_{u,t}$ information encoded in representations, we introduce learn privacy-irrelevant representation $\boldsymbol{z^{ir}}$. This goal is formulated as minimizing the MI: $\min I(\boldsymbol{z^{ir}}; \mu_{u,t})$. Since both user representation $\boldsymbol{z_u}$ and the interacted items' representations $\boldsymbol{z_i}$ reflect sensitive user attributes, we apply the learning objective to both.

We first get the representation from user ID x_u and item ID x_i, where the privacy-irrelevant user and item representations are denoted as $\boldsymbol{z_u^{ir}}$ and $\boldsymbol{z_i^{ir}}$:

$$\boldsymbol{z_u^{ir}} = f_{\theta_u^{ir}}(x_u), \tag{8}$$

$$\boldsymbol{z_i^{ir}} = f_{\theta_i^{ir}}(x_i), \tag{9}$$

where $f_{\theta_u^{ir}}$ and $f_{\theta_i^{ir}}$ represent the corresponding encoders, we implement the encoders by embedding layers.

We then minimize the MI between representation and attributes as the following MI objective:

$$\min_{\theta} I(\boldsymbol{Z}_{u,i}^{ir}; \mu_{u,t}), \tag{10}$$

$$\boldsymbol{Z}_{u,i}^{ir} = [\boldsymbol{z_u^{ir}}, \boldsymbol{z_i^{ir}}], \tag{11}$$

where $[\cdot,\cdot]$ denotes feature concatenation. In the following, we denote $\boldsymbol{Z}_{u,i}^{ir}$ as the two variables for simplicity.

To compute the intractable MI goal [33] in Equation (10), we propose optimizing the variational CLUB (I_{vCLUB}) upper bound proposed in [37] as follows,

$$\min_{\theta} I(\boldsymbol{Z}_{u,i}^{ir}; \mu_{u,t}) \tag{12}$$

$$\leq \min_{\theta} I_{vCLUB}(\boldsymbol{Z}_{u,i}^{ir}; \mu_{u,t}) \tag{13}$$

$$= \min_{\theta} \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})}[\log q_\psi(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})] \tag{14}$$

$$- \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir})p(\mu_{u',t})}[\log q_\psi(\mu_{u',t}|\boldsymbol{Z}_{u,i}^{ir})]$$

$$\approx \min_{\theta} \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})}\left[\log q_\psi(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})\right], \tag{15}$$

where $q_\psi(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})$ is a variational approximation to the true posterior $p(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})$, and we realize the distribution by a three-layer multi-layer perceptron (MLP), denoted as privacy estimator $g_{\psi_{u,t}^{ir}}(\cdot)$. The privacy estimator g_{ψ_t} takes representations as input and predicts the attribute $\mu_{u,t}$. In Equation (14), the second term represents the MI between the representation pair $(\boldsymbol{Z}_{u,i}^{ir})$ and negative samples $\mu_{u',t}$ from other users ($u' \neq u$). For implementation efficiency, we discard this term and derive a simplified bound in Equation (15). Following [22], we derive the objective as the *cross-entropy* (CE) loss, and optimize the encoders through the client's private data $\mathcal{T}_u$:

$$\mathcal{L}_{ir} = -\sum_{\mu_{u,t} \in \mathcal{T}_u} CE(g_{\psi_{u,t}^{ir}}(\boldsymbol{Z}_{u,i}^{ir}), \mu_{u,t}), \tag{16}$$

which encourages the learned representations to obfuscate sensitive attributes, making them difficult for the privacy estimator to predict.

In addition, realizing the goal of I_{vCLUB} should satisfy a specific condition to ensure q_ψ provides a valid approximation. The condition of Equation (16) is derived as follows,

$$\min_{\psi} KL\left(p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t}) \parallel q_\psi(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})\right) \tag{17}$$

$$= \min_{\psi} \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})}\left[\log \frac{p(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})p(\boldsymbol{Z}_{u,i}^{ir})}{q_\psi(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})p(\boldsymbol{Z}_{u,i}^{ir})}\right] \tag{18}$$

$$= \min_{\psi} \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})}\left[\log p(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})\right] \tag{19}$$

$$- \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})}\left[\log q_\psi(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})\right]$$

$$\Leftrightarrow \max_{\psi} \mathbb{E}_{p(\boldsymbol{Z}_{u,i}^{ir}, \mu_{u,t})}\left[\log q_\psi(\mu_{u,t}|\boldsymbol{Z}_{u,i}^{ir})\right], \tag{20}$$

where $KL(p(\cdot)||q(\cdot))$ is the Kullback–Leibler divergence between distribution $p(\cdot)$ and $q(\cdot)$. In Equation (19), the first term $\mathbb{E}_{p(\cdot)}[\log p(\cdot)]$ is independent of the estimator g_ψ, so maximizing the second term suffices. To this end, we achieve the condition by the following CE loss on public data D_{pub} to train the privacy estimator:

$$\mathcal{L}_{con} = \sum_{(\boldsymbol{Z}_{u',i'}^{ir}, \mu_{u',t}) \in D_{pub}} CE(g_{\psi_{u,t}^{ir}}(\boldsymbol{Z}_{u',i'}^{ir}), \mu_{u',t}), \tag{21}$$

which encourages accurate prediction of attributes from their privacy-irrelevant representations. In practice, we also include user u's own data for training the estimator.

Summary. The training of privacy-irrelevant representation follows an **adversarial learning** paradigm: The privacy estimator $g_{\psi_{u,t}^{ir}}$ is trained to *predict* sensitive attributes from representation via loss function $\mathcal{L}_{con}$ (Equation (21)). Conversely, the encoders $f_{\theta_u^{ir}}$ and $f_{\theta_i^{ir}}$ are optimized by $\mathcal{L}_{ir}$ (Equation (16)) to *prevent* the estimator from inferring attribute information. Here, the estimator and the encoder pursue **opposite objectives**, forming the adversarial learning paradigm.

B. Privacy-relevant Representation Learning

To ensure attribute information $\mu_{u,t}$ is effectively encoded in representations, we introduce learn privacy-relevant representation $\boldsymbol{z^{re}}$. A straightforward approach to learning $\boldsymbol{z^{re}}$ is to learn dense embeddings directly from one-hot encoded user attributes, i.e., $\boldsymbol{z^{re}} = Emb(\mu_u)$. However, this introduces a significant privacy risk in federated settings. Since each client only possesses attributes from a single user, learning meaningful attribute embeddings would require aggregating attribute information across multiple clients during training, which inevitably leads to privacy leakage. Therefore, we design a privacy-relevant representation learning approach from the MI estimation perspective.

To learn $\boldsymbol{z^{re}}$, we introduce a MI objective as $\max I(\boldsymbol{z^{re}}; \mu_{u,t})$. This objective encourages $\boldsymbol{z^{re}}$ to capture as much attribute-relevant information as possible. In addition, we aim for $\boldsymbol{z^{re}}$ to maintain privacy-relevant features while excluding privacy-irrelevant features exclusively. This design prevents feature redundancy between $\boldsymbol{z^{re}}$ and $\boldsymbol{z^{ir}}$. To achieve this, we introduce a conditional entropy objective $\min H(\boldsymbol{z^{re}}|\mu_{u,t})$ that constrains $\boldsymbol{z^{re}}$'s feature diversity given $\mu_{u,t}$, forcing it to focus solely on attribute-related features.

We apply privacy-relevant representation learning only to users, not items, as maintaining two item encoders would incur prohibitive memory overhead. Specifically, we generate a privacy-relevant user representation from user ID x_u via:

$$\boldsymbol{z_u^{re}} = f_{\theta_u^{re}}(x_u), \tag{22}$$

where $f_{\theta_u^{re}}$ is the privacy-relevant user encoder.

The representation $\boldsymbol{z_u^{re}}$ is then optimized using two complementary objectives:

$$\max I(\boldsymbol{z_u^{re}}; \mu_{u,t}), \quad \min H(\boldsymbol{z_u^{re}}|\mu_{u,t}). \tag{23}$$

Forward Predictive Objective. Inspired by [38], we adopt a *forward predictive* bound to approximate the mutual in-

formation term $I(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}; \mu_{u,t})$ as the computation of mutual information is intractable. The derivation is as follows:

$$\min_{\theta} I(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}; \mu_{u,t}) \tag{24}$$

$$= \min_{\theta} H(\mu_{u,t}) - H(\mu_{u,t}|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}) \tag{25}$$

$$\Leftrightarrow \min_{\theta} -H(\mu_{u,t}|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}) \tag{26}$$

$$= \min_{\theta} \mathbb{E}_{p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \mu_{u,t})}[\log p(\mu_{u,t}|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}})] \tag{27}$$

$$\geq \min_{\theta} \mathbb{E}_{p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \mu_{u,t})}[\log q_{\psi}(\mu_{u,t}|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}})]. \tag{28}$$

In Equation (25), $H(\mu_{u,t})$ is a constant, thus can be removed. We propose a variational distribution q_{ψ} to approximate the true distribution, parameterized by a privacy estimator $g_{\psi_{u,t}^{FP}}$.

We derive the objective in Equation (28) for the user encoder as CE loss, and optimize the user encoder through the client's private data $\mathcal{T}_u$ with:

$$\mathcal{L}_{re}^{FP} = \sum_{\mu_{u,t} \in \mathcal{T}_u} CE(g_{\psi_{u,t}^{FP}}(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}), \mu_{u,t}), \tag{29}$$

which encourages $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}$ to preserve attribute information that facilitates accurate prediction.

To achieve Equation (29), the estimator $g_{\psi_{u,t}^{FP}}$ is trained using public data D_{pub} via:

$$\mathcal{L}_{con}^{FP} = \sum_{(\boldsymbol{z}_{\boldsymbol{u'}}^{\boldsymbol{re}}, \mu_{u',t}) \in D_{pub}} CE(g_{\psi_{u,t}^{FP}}(\boldsymbol{z}_{\boldsymbol{u'}}^{\boldsymbol{re}}), \mu_{u',t}). \tag{30}$$

Inverse Predictive Objective. To optimize the conditional entropy $\min H(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}|\mu_{u,t})$ in Equation (23), we propose *inverse predictive* objective. Given $H(\mu_{u,t}|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}) = -\mathbb{E}_{p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \mu_{u,t})}[\log p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}|\mu_{u,t})]$, similar to Equation (26), we use $\mathbb{E}_{p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \mu_{u,t})}[\log q_{\psi}(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}|\mu_{u,t})]$ as a lower bound:

$$\min_{\theta} H(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}|\mu_{u,t})$$

$$= \min_{\theta} -\mathbb{E}_{p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \mu_{u,t})}[\log p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}|\mu_{u,t})] \tag{31}$$

$$\Leftrightarrow \max_{\theta} \mathbb{E}_{p(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \mu_{u,t})}[\log q_{\psi}(\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}|\mu_{u,t})], \tag{32}$$

we approximate the true conditional distribution using another estimator $g_{\psi_{u,t}^{IP}}$, which takes attribute $\mu_{u,t}$ as input and output the predicted representation $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}$.

We derive the objective in Equation (32) for the user encoder as L2 loss, and optimize it with:

$$\mathcal{L}_{re}^{IP} = \|\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}} - g_{\psi_{u,t}^{IP}}(\mu_{u,t})\|_2. \tag{33}$$

Importantly, the estimator $g_{\psi_{u,t}^{IP}}$ is trained only on public data D_{pub} to prevent attribute leakage. The corresponding training loss is:

$$\mathcal{L}_{con}^{IP} = \sum_{(\boldsymbol{z}_{\boldsymbol{u'}}^{\boldsymbol{re}}, \mu_{u',t}) \in D_{pub}} \|\boldsymbol{z}_{\boldsymbol{u'}}^{\boldsymbol{re}} - g_{\psi_{u,t}^{IP}}(\mu_{u',t})\|_2. \tag{34}$$

Summary. Our privacy-relevant representation learning integrates both forward and inverse predictive objectives. Consequently, this can be viewed as a **cooperative learning** paradigm: The forward predictive task encourages the estimator to infer attributes $\mu_{u,t}$ from $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}$, while the inverse predictive task encourages it to recover $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}$ from $\mu_{u,t}$. Here, both encoder and privacy estimators share a **common objective** to strengthen the mutual predictability between $\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}$ and $\mu_{u,t}$. Together, these objectives guide the user encoder $f_{\theta_u^{re}}$ to produce privacy-relevant representations that are both informative and compact concerning user attributes.

C. Recommendation and Overall Objectives

Recommendation Objective. On the client side, the privacy-aware representations are all participants in the recommendation process. Specifically, the predicted rating $\hat{r}_{ui}$ of user u on item i is computed by a prediction layer h_{ϕ} as follows:

$$\hat{r}_{ui} = h_{\phi}([\boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{ir}}, \boldsymbol{z}_{\boldsymbol{u}}^{\boldsymbol{re}}, \boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{ir}}]), \tag{35}$$

where $[\cdot, \cdot]$ denotes feature concatenation. We implement h_{ϕ} as a 3-layer MLP model.

The recommender is trained using the *Binary Cross-Entropy* (BCE) loss over observed positive and negative interactions:

$$\mathcal{L}_{rec} = -\sum_{(u,i) \in S_u^+} \log \hat{r}_{ui} - \sum_{(u,i') \in S_u^-} \log(1 - \hat{r}_{ui'}), \tag{36}$$

where S_u^+ denotes the set of positive interactions ($r_{ui} = 1$), and S_u^- includes negative interactions ($r_{ui} = 0$).

Overall Optimization Process. The learning process on each client contains two categories of learnable parameters: the privacy estimators ψ, and the recommender model θ, ϕ. To decouple their conflicting loss functions and stabilize training, we adopt a two-stage optimization strategy.

First, we update the privacy estimators to maximize their ability to infer sensitive attributes:

$$\psi_* = \arg\min_{\psi_*} (\mathcal{L}_{con} + \mathcal{L}_{con}^{FP} + \mathcal{L}_{con}^{IP}), \tag{37}$$

where $\psi_* = \{\psi_{u,t}^{ir}, \psi_{u,t}^{FP}, \psi_{u,t}^{IP} \mid t \in \mathcal{T}_u\}$. During this step, the privacy-aware representations are detached from the gradient computation via stop-gradient operations, ensuring that only the estimators are updated.

Next, we optimize the recommender model by jointly considering recommendation accuracy and privacy preservation:

$$\theta_*, \phi = \arg\min_{\theta_*, \phi} (\mathcal{L}_{rec} + \lambda_{ir}\mathcal{L}_{ir} + \lambda_{re}(\mathcal{L}_{re}^{FP} + \mathcal{L}_{re}^{IP})), \tag{38}$$

where $\theta_* = \{\theta_u^{re}, \theta_u^{ir}, \theta_i^{ir}\}$. During this phase, the stop gradient operation is applied to privacy estimators g_{ψ} so that the encoders are trained to either eliminate or preserve attribute information in line with their respective objectives. Notably, we need to train the privacy estimators g_{ψ} for each attribute $t \in \mathcal{T}_u$ and eliminate privacy information for all attributes in Equation (38).

D. Parameters Upload and Federated Aggregation

After local training on each client, model parameters are uploaded to the server for aggregation. We denote the parameters uploaded by the client u in global round r as $\Theta_{u,r}$. To prevent privacy leakage, only the user and item encoders of privacy-irrelevant representations are transmitted, i.e., $\Theta_{u,r} = \{\theta_u^{ir}, \theta_i^{ir}\}$.

Algorithm 1: Federated Training Procedure of PARD

Input: $\mathcal{U}$: Training user set, $\mathcal{I}$: Item set, $\lambda_{ir}, \lambda_{re}$: Hyperparameters, η: Privacy-insensitive user ratio, R: Global rounds, L: Local iterations

Output: θ_i^{ir}, θ_u^{ir}: Parameters of privacy-irrelevant encoder, $\{\theta_u^{re}\}_{u\in\mathcal{U}}, \{\phi_u\}_{u\in\mathcal{U}}$: Parameters of client-side privacy-relevant user encoders and prediction layers, ψ^*: Parameters of privacy estimators

Initialize all parameters θ_i^{ir}, θ_u^{ir}, θ_u^{re}, ϕ_u, ψ^*
Select privacy-insensitive users $\mathcal{U}^{ins}$ by ratio η
for *global round* $r = 0$ *to* $R-1$ **do**
  Randomly select client subset $\mathcal{U}^* \subseteq \mathcal{U}$
  for *each client* $u \in \mathcal{U}^*$ ***in parallel*** **do**
    if $r > 0$ **then**
      `// Initialize client's parameters`
      $\theta_u^{ir}, \theta_i^{ir} \leftarrow \Theta_r$
    end
    for *local iteration* $l = 0$ *to* $L-1$ **do**
      `// Privacy estimator updates`
      Compute $\mathcal{L}_{con}$, $\mathcal{L}_{con}^{FP}$, $\mathcal{L}_{con}^{IP}$ via (21,30,34)
      Update ψ^* via (37)
      `// Recommender updates`
      Compute $\mathcal{L}_{rec}$ via (36)
      Compute $\mathcal{L}_{ir}$, $\mathcal{L}_{re}^{FP}$, $\mathcal{L}_{re}^{IP}$ via (16,29,33)
      Update θ_u^{ir}, θ_i^{ir}, θ_u^{re}, ϕ_u via (38)
    end
    `// Client-server communication`
    Upload $\Theta_{u,r} = \{\theta_u^{ir}, \theta_i^{ir}\}$ to server
    Retain θ_u^{re}, ϕ_u, ψ^* on client
    if u *in* $\mathcal{U}^{ins}$ **then**
      Upload $\theta_u^{ir}, \theta_u^{re}, \theta_i^{ir}$ and μ_u to dataset D_{pub}
    end
  end
  `// Federated aggregation on server`
  $\Theta_{r+1} \leftarrow \frac{1}{|\mathcal{U}^*|}\sum_{u\in\mathcal{U}^*}\Theta_{u,r}$
  Distribute Θ_{r+1} to all clients
end

Inspired by [26], we retain the prediction layer h_ϕ locally on each client to support personalized modeling, avoiding the degradation of individual recommendation quality due to global homogenization. In addition, a small fraction η of clients (denoted as set $\mathcal{U}_{ins}$) voluntarily contribute their private data as public knowledge, forming the shared dataset D_{pub}.

The server performs uniform aggregation over the uploaded parameters using the FedAvg algorithm [36] in Equation (5) to get Θ_{r+1}. This approach ensures fair contribution from each participant. The aggregated global parameters Θ_{r+1} are then broadcast to all clients, including those not involved in the current round, enabling system-wide knowledge sharing.

The full federated training procedure proceeds for R global rounds, and its pseudocode is summarized in Algorithm 1.

TABLE II
STATISTICS OF THE THREE DATASETS.

Datasets	**ML-100k**	**Taobao**	**ML-1M**
#Users	943	3,198	6,040
#Items	1,682	4,282	3,706
#Interactions	100,000	48,002	1,000,209
Sparsity	93.70%	99.65%	95.53%
Private attributes (#Categories)	Age (3), Gender (2), Occupation (21)	Age (7), Gender (2)	Age (3), Gender (2), Occupation (21)

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets: Following [21], [39], we use datasets: ML-100k and ML-1M [40] and Taobao [41]. ML-100k and ML-1M are collected through the MovieLens website, and each user has at least 20 interactions. Taobao is collected from Taobao's recommendation system, and we excluded users and items with less than 10 and 5 interactions, respectively. The ML-100k and ML-1M have age, gender, and occupation as private attributes with 3, 2, and 21 categories, respectively, and Taobao includes age and gender with 7 and 2 categories. The statistical information is summarized in Table II.

2) Evaluation Metrics: (1) **Recommendation-Related Metrics**. Following the leave-one-out strategy in [2], we use Recall (R@K) to measure the proportion of test items in top-K recommendations and Normalized Discounted Cumulative Gain (N@K) to evaluate ranked positions, emphasizing higher-ranked items. (2) **Privacy-Related Metrics**. As in [21], [22], privacy is evaluated via attribute inference attacks. For binary attributes (e.g., gender), AUC measures the probability that positive samples rank higher than negatives. For multi-class attributes (e.g., age), micro-F1 balances class imbalances, with lower scores indicating better privacy protection.

3) Base Recommender Models: We implement PARD and all baselines within the following federated recommendation (FedRec) paradigms: (1) **FedNCF** [24]: This model extends Neural Collaborative Filtering (NCF) [2] to a federated learning setting. User embeddings are treated as private and are updated locally on each client, while item embeddings and the prediction layer are shared and collaboratively updated via server-side aggregation. (2) **FedGCN**: We extend LightGCN [42] to the federated learning framework. Each client trains the local LightGCN with the first-order interaction subgraph. We implement the FedGCN model following [21].

4) Compared Models: We compare the recommendation performance and privacy-preserving capability of PARD with the following categories. (1) **Random Attack**. (2) Base FedRec: **FedAvg** [36]. (3) Performance-focused FedRec: **PFedRec** [27]. (4) Privacy-focused FedRec: **Early-stop**, **APM** [21], **UC-FedRec** [22]. The detailed information of these methods is as follows, and unless otherwise noted, clients upload item embeddings and the prediction layer to the server during training.

- **Random Attack**: A naive attacker that predicts the target user's attribute by the distribution of that attribute in

TABLE III
PERFORMANCE COMPARISON OF VARIOUS MODELS IN RECOMMENDATION. FOR "PRIVACY", THE TOP RESULT IS BOLDED AND THE SECOND-BEST IS UNDERLINED. FOR "UTILITY", FOCUSING ON THE PERFORMANCE LOSS OF PRIVACY-PRESERVING ALGORITHMS, ONLY THESE ALGORITHMS ARE MARKED, EXCLUDING BASIC FEDREC MODELS. FOR ACRONYMS USED, "GEN" REFERS TO GENDER, AND "OCC" MEANS OCCUPATION.

Methods		ML-100k					ML-1M					Taobao			
		Privacy (↓)			Utility (↑)		Privacy (↓)			Utility (↑)		Privacy (↓)		Utility (↑)	
Base	Protection	Age	Gen	Occ	R@10	N@10	Age	Gen	Occ	R@10	N@10	Age	Gen	R@10	N@10
Random Attack		0.4026	0.5151	0.0834	-	-	0.4245	0.5056	0.0743	-	-	0.2301	0.5026	-	-
FedNCF	FedAvg	0.6371	0.7348	0.2411	0.6277	0.3478	0.7032	0.8251	0.1952	0.1685	0.0867	-	-	0.0638	0.0321
	PFedRec	0.6530	0.7454	0.2384	0.6267	0.3638	0.7034	0.8052	0.1960	0.1677	0.0842	0.3978	0.5929	0.1748	**0.1215**
	Early-stop	0.6132	0.7201	0.2411	0.5536	0.3023	0.6819	0.8065	0.1889	0.1318	0.0659	-	-	-	-
	APM	0.6013	0.6970	0.2238	0.5535	0.3041	0.6774	0.7991	0.1858	0.1301	0.0655	-	-	-	-
	UC-FedRec	0.6185	0.7260	0.2305	0.6299	0.3629	0.6639	0.7674	0.1757	0.1639	0.0824	0.3861	0.5302	0.1647	0.0998
	PARD	**0.5828**	**0.5777**	**0.2079**	**0.6331**	**0.3697**	**0.6169**	**0.7370**	**0.1691**	**0.1808**	**0.0923**	**0.3540**	**0.4405**	**0.1829**	0.1125
FedGCN	FedAvg	0.6238	0.6972	0.2212	0.6235	0.3517	0.6838	0.8033	0.1867	0.1536	0.0776	0.4674	0.6135	0.1754	0.0990
	PFedRec	0.6411	0.7323	0.2278	0.6246	0.3550	0.6790	0.7908	0.1861	0.1553	0.0779	0.4068	0.5793	0.1757	**0.1338**
	Early-stop	0.6079	0.6861	0.2278	0.5864	0.3227	0.6604	0.7882	0.1794	0.1303	0.0649	0.4642	0.6328	0.1376	0.0734
	APM	0.6066	0.6738	0.2265	0.5864	0.3224	0.6480	0.7751	0.1753	0.1298	0.0660	0.4482	0.6165	0.1372	0.0728
	UC-FedRec	0.6013	0.7241	0.2331	0.6225	0.3520	0.6629	0.7786	0.1856	0.1578	0.0797	0.4189	0.5548	0.1651	0.0945
	PARD	**0.5960**	**0.5961**	**0.2185**	**0.6257**	**0.3680**	**0.6256**	**0.7170**	**0.1623**	**0.1636**	**0.0826**	**0.3513**	**0.4525**	**0.1764**	0.1122

the public dataset D_{pub}. This serves as a lower-bound reference for attribute inference accuracy.

- **FedAvg** [36]: A standard federated learning algorithm where clients locally train models and upload updates to a central server, which performs average pooling to obtain a global model.
- **PFedRec** [27]: A performance-focused federated recommender that keeps the prediction layer entirely local to each client for learning personalized user preferences. The global item embeddings are downloaded from the server and then fine-tuned locally. Only item embeddings are uploaded to the server, reducing the privacy exposure of user-specific parameters.
- **Early-stop**: This approach halts model training at an early stage to reduce the risk of memorizing sensitive user information. By avoiding overfitting, it mitigates potential privacy leakage through over-trained model parameters.
- **APM** [21]: This method investigates which components of FedRec (user embeddings, item embeddings, prediction layers) contribute to privacy leakage. It then applies adaptive differential privacy mechanisms to each component with module-specific privacy budgets, balancing utility and privacy.
- **UC-FedRec** [22]: It introduces adversarial training via surrogate attackers that aim to infer private attributes from user embeddings. These attackers are trained jointly with the recommender model by public data from privacy-insensitive users ($\mathcal{U}_{ins}$). It upload privacy-preserving user embeddings and item embeddings to the server.

5) Implementation Details: We adopt the standard FedRec paradigm with parameter-exchange architecture (consistent with [21], [22]), using SGD for optimization. Key settings include: embedding dimension 64, learning rate 0.5, batch size 256, and 1 local iteration per communication round ($L = 1$). For APM, we apply additive Laplace noise with factors $\{0.017, 0.033\}$ as recommended. Privacy estimators are 3-layer MLPs trained via SGD ($lr = 0.1$), with loss coefficients $\lambda_{ir}, \lambda_{re}$ selected via grid search over $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The proportion of privacy-insensitive users in D_{pub} is set to $\eta = 0.2$, contributing both model parameters and attribute labels. For fair comparison, PFedRec and UC-FedRec are tuned to match FedAvg's recommendation performance, while Early-stop is calibrated to align with APM's utility level. This categorizes PFedRec/UC-FedRec as minor-utility-loss privacy methods and APM/Early-stop as strong-privacy-with-degradation baselines.

B. Privacy Evaluation

We apply PARD to two representative FedRec models (FedNCF and FedGCN) and compare its privacy-preserving and recommendation performance against various baselines, as summarized in Table III. On the Taobao dataset under the FedNCF paradigm, some entries (e.g., FedAvg, Early-stop, and APM) are marked as "-" because the user-item interaction matrix is extremely sparse. Under this high sparsity, FedAvg-based FedNCF fails to converge, making both utility and privacy evaluations unreliable. As Early-stop and APM are built upon this framework, they also cannot be effectively trained. Therefore, their results are omitted in this case. From Table III, our observations are as follows.

First, base FedRec (FedAvg) and performance-focused FedRec (PFedRec) achieve strong recommendation performance but are most vulnerable to attribute inference attacks. This is because their uploaded parameters fully encode user behavior patterns and users' privacy-coupled preferences, inevitably leaking private user information. PFedRec slightly improves privacy by retaining the prediction layer locally, but still uploads item embeddings containing privacy information.

Second, Early-stop and APM are designed to improve privacy protection. Early-stop prevents models from overfitting to user-specific signals, while APM adds differentially private noise to sensitive components. Between the two, APM generally achieves better privacy protection under the same utility level, especially for the more sensitive attribute "Age"

TABLE IV
ABLATION STUDY ON THE DECOUPLED LEARNING GOALS.

Strategy	ML-100k		Taobao	
	Age	R@10	Age	R@10
(0) Base (PARD w/o both)	0.6424	0.6405	0.4091	0.1836
(1) Base + privacy-irrelevant learning	0.5841	0.6277	0.3544	0.1745
(2) Base + privacy-relevant learning	0.6411	0.6426	0.4115	0.1854
(3) Base + both (PARD)	0.5828	0.6331	0.3540	0.1829

on ML-100k and ML-1M. This shows the effectiveness of adaptive differential privacy. However, APM still leaks some private signals and suffers a noticeable utility drop.

Third, UC-FedRec employs adversarial learning to suppress sensitive information in user embeddings. Compared to FedAvg, it consistently enhances privacy protection across all datasets and models, while incurring less utility degradation than APM and Early-stop. This makes UC-FedRec a strong baseline for balancing privacy and recommendation accuracy. However, its privacy-preserving performance is not consistently the second-best. This limitation arises because UC-FedRec assumes that only user embeddings contain sensitive information and focuses solely on purging privacy signals from them, while overlooking the potential leakage through item embeddings associated with user interactions.

Fourth, PARD consistently achieves the best privacy-preserving performance across all settings, often outperforming all baselines by a large margin. This is because PARD ensures that the uploaded parameters are privacy-irrelevant, while the potentially sensitive but utility-relevant signals are retained in the local model. Remarkably, this privacy advantage does not sacrifice utility-instead, PARD outperforms other methods in recommendation accuracy as well.

Specifically, PARD even surpasses vanilla FedRec in recommendation performance. This is due to its more personalized design: by keeping the privacy-relevant user encoder and prediction layer entirely local, PARD avoids the limitations of globally shared representations that fail to capture user-specific signals. This suggests that privacy-preserving design can also enhance personalization.

Furthermore, on Taobao, the gender attack performance against PARD drops below the random attack baseline, indicating that the local models in PARD may learn to retain misleading privacy patterns, confusing potential attackers.

C. Ablation Study

To better understand the contribution of each component in our decoupled learning framework, we conduct an ablation study by selectively disabling the two learning objectives in PARD, as shown in Table IV. Specifically, we analyze the following variants:

- Base (w/o both): Removes both privacy-irrelevant and privacy-relevant learning goals. This acts as the backbone model without any decoupling design.
- Base + privacy-irrelevant learning: Adds only the privacy-irrelevant learning objective.
- Base + privacy-relevant learning: Adds only the privacy-relevant learning objective.

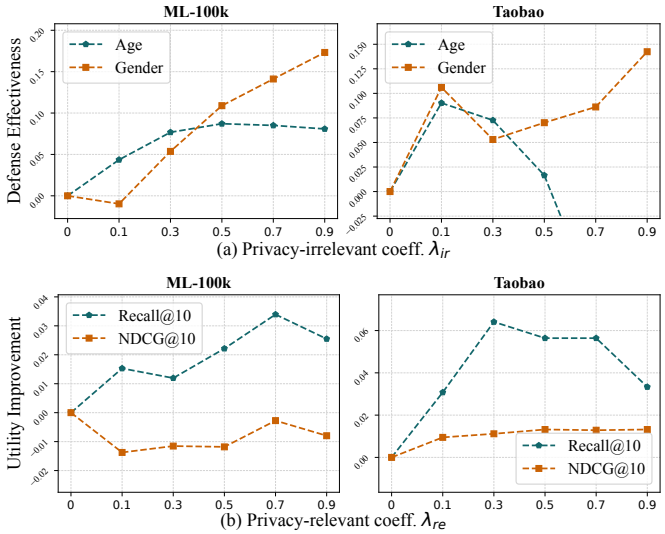


Fig. 6. Effect of decoupled learning coefficients.

- Base + both (PARD): Full model with both objectives.

We summarize the key observations below:

First, comparing (1) with the Base, we observe a substantial improvement in privacy protection (Age AUC drops from 0.6424 to 0.5841) with a reduction in recommendation accuracy (R@10 drops from 0.6405 to 0.6277). This confirms the effectiveness of the privacy-irrelevant learning objective in reducing the mutual information between uploaded parameters and private attributes. However, removing private information comes at a minor cost to utility.

Second, comparing (2) with the Base shows that applying only privacy-relevant learning slightly improves the recommendation performance (R@10 improves from 0.6405 to 0.6426), but does not improve privacy (Age AUC remains high at 0.6411). This indicates that while privacy-relevant learning helps recover recommendation utility, it alone does not suppress private signal leakage, as no explicit privacy-minimization is enforced.

Third, combining both objectives in (3) yields the best trade-off: Age AUC further drops to 0.5828, while the accuracy of the recommendation reaches 0.6331, outperforming both (1) and (2). This demonstrates the complementary roles of the two learning objectives. Moreover, (3) slightly outperforms (2) in R@10, showing that privacy-relevant learning can effectively compensate for the utility degradation caused by (1).

Finally, the recommendation performance gap between Base and (3) suggests that privacy-irrelevant learning may also eliminate some item-side privacy-relevant signals, which cannot be locally recovered efficiently due to storage constraints (e.g., limited ability to storage privacy-relevant item representations). Addressing this limitation (e.g., via partial item personalization) is left as future work.

D. Effect of Hyper-parameters

We study the impact of the decoupled learning coefficients in PARD, namely the privacy-irrelevant coefficient λ_{ir} and the privacy-relevant coefficient λ_{re}. These two hyperparameters

TABLE V
EFFECT OF THE RATIO OF $\mathcal{U}_{ins}$ ON ML-100K OF FEDNCF.

Ratio η	Models	Privacy		
		Age	Gen	Occ
0.1	FedAvg	0.5948	0.7350	0.2179
	APM	0.6042	0.6935	0.2085
	PARD	**0.5795**	**0.6391**	**0.1731**
0.2	FedAvg	0.6371	0.7348	0.2411
	APM	0.6013	0.6970	0.2238
	PARD	**0.5828**	**0.5777**	**0.2079**
0.3	FedAvg	0.6520	0.7547	0.2330
	APM	0.6112	0.6970	0.2269
	PARD	**0.5900**	**0.6543**	**0.2239**

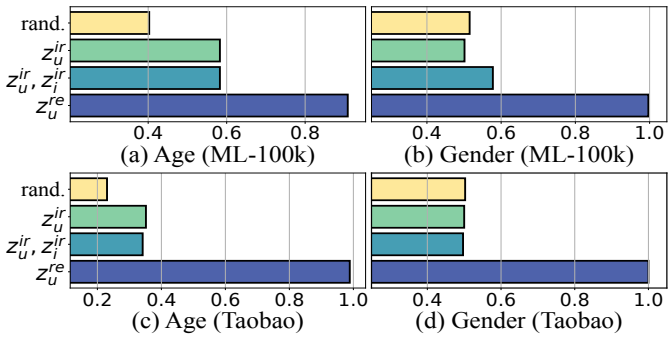


Fig. 7. Vulnerability of different representations.

are designed to control the extent of privacy suppression and utility enhancement, respectively.

To evaluate their effect, we report: (1) *Defense Effectiveness*, the relative reduction in AIA accuracy, measuring how well privacy-relevant information is suppressed (Fig. 6 (a), varying λ_{ir}). (2) *Utility Improvement*, the relative gain in recommendation performance, measured by Recall@10 and NDCG@10 (Fig. 6 (b), varying λ_{re}).

1) Effect of Privacy-Irrelevant Coefficient λ_{ir}: As shown in Fig. 6 (a), increasing λ_{ir} generally improves defense effectiveness across both datasets, indicating stronger suppression of private signals in the uploaded representations. However, the degree of improvement varies by attribute: For Gender, defense effectiveness steadily improves with larger λ_{ir} on both datasets, reflecting its binary and relatively easy-to-remove nature. For Age, performance peaks at moderate values of λ_{ir} (e.g., 0.3 or 0.5), then plateaus (ML-100k) or drops sharply (Taobao). This suggests over-suppressing may cause the model to misallocate learning focus, hurting the obfuscation of more complex attributes.

These trends highlight that while a higher λ_{ir} strengthens privacy protection, overly large values may introduce imbalance, especially for attributes with higher cardinality. Practically, a moderate λ_{ir} offers better overall protection.

2) Effect of Privacy-Relevant Coefficient λ_{re}: Fig. 6 (b) shows that increasing λ_{re} consistently enhances recommendation utility, particularly Recall@10. The gains are more pronounced on the Taobao dataset, suggesting that emphasizing privacy-relevant representations effectively captures user-specific preference signals.

However, the performance plateaus when $\lambda_{re} = 0.7$ in ML-100k and $\lambda_{re} = 0.3$ in Taobao, indicating that most of the utility benefit has been captured by that point. Beyond this, additional emphasis on privacy-relevant components contributes diminishing returns and may even slightly harm performance

3) Summary: Overall, these findings suggest that carefully balancing λ_{ir} and λ_{re} is crucial. For instance, moderate λ_{ir} helps suppress privacy signals without harming difficult attributes, while increasing λ_{re} improves recommendation utility up to a point.

E. In-depth Analysis

1) Public data ratio η: Table V reports the privacy protection performance under varying ratios η of insensitive users $\mathcal{U}_{ins}$, which reflects the amount of public data accessible to the attacker. As η increases from 0.1 to 0.3, all models experience degraded privacy protection, since the attacker can exploit more labeled data to train stronger inference models.

Despite this, PARD consistently outperforms FedAvg and APM across all attributes and ratios, demonstrating its robustness even under stronger attacker assumptions. Notably, the performance degradation of PARD is relatively moderate, especially on sensitive attributes like Age and Gender, indicating that the decoupled learning strategy generalizes well against adversaries with increasing prior knowledge.

2) Privacy study on decoupled representations: To verify whether PARD successfully decouples privacy-relevant and privacy-irrelevant information, we conduct AIA on: (1) privacy-irrelevant user embeddings $\boldsymbol{z_u^{ir}}$, (2) combined privacy-irrelevant user and item embeddings $\boldsymbol{z_u^{ir}}, \boldsymbol{z_i^{ir}}$, and (3) privacy-relevant user embeddings $\boldsymbol{z_u^{re}}$. Fig. 7 demonstrates the AIA results on multi-attributes and two datasets.

We observe that AIA accuracy on $\boldsymbol{z_u^{ir}}$ is close to random guessing, suggesting effective removal of sensitive signals from the user's privacy-irrelevant representation. Incorporating $\boldsymbol{z_i^{ir}}$ leads to similar or slightly lower inference accuracy, suggesting that privacy leakage is also mitigated in the privacy-irrelevant item embeddings. In contrast, $\boldsymbol{z_u^{re}}$ yields high AIA accuracy, demonstrating that privacy-relevant signals are preserved in this embedding.

These results validate the effectiveness of our decoupled representation design: privacy-irrelevant components suppress attribute leakage, while privacy-relevant components retain user-specific signals essential for personalization. This confirms that PARD achieves both privacy preservation and utility retention through explicit decoupling of representations.

V. RELATED WORK

A. Privacy Risks in Recommender Systems

In this section, we categorize the privacy risks in Recommender Systems (RSs) into two categories: direct and indirect privacy risks, as detailed in [43]. Direct privacy risks occur when the adversary gains access to users' actual private data. Indirect privacy risks, conversely, involve the adversary inferring or guessing information without direct data access.

Direct privacy risks often materialize when RSs, either intentionally or unintentionally, collect users' private data for

training the recommender model [19], [25]. This exposes users' personal attributes, users' location information, and interaction histories to potential adversaries.

Regarding indirect privacy risks, adversaries lack direct access to private data. Recent research [23], [44], [45] has shown that model parameters tend to "memorize" features or attributes from the training data. As a result, adversaries can infer users' private information by accessing the recommender model's parameters [34] or its output [46]. This process is known as privacy inference attacks, which can be further classified into two major categories: (1) Membership Inference Attacks (MIA) [47], [48] determine whether specific user data or user-item interactions were used during the training of the recommender model. For instance, [46], [49] have revealed the identities of users involved in the training, while [34], [50] have inferred user-item interactions in both federated and general recommender systems. Related studies [51], [52] have explored similar inference tasks in different contexts, such as predicting user location interactions or trajectories. (2) Attribute Inference Attacks (AIA) aim to deduce sensitive user attributes (e.g., age, gender, location) from the model's parameters, even when such attributes are not explicitly part of the training data. For example, [23] designed an RNN module to infer attributes from user and item embeddings in general recommenders, and [53] uncovered attribute leakage in graph-based recommenders. Recent works [21], [22] have further demonstrated attribute privacy risks in FedRec.

Our work focuses on attribute privacy issues. Although these vulnerabilities have been studied in general recommender systems, the development of effective defense mechanisms in FedRec remains an open area of research.

B. Federated Recommender Systems

FedRec have emerged as a privacy-preserving solution to the direct privacy leakage problems associated with centralized recommender systems, leveraging the power of Federated Learning (FL) frameworks [20].

FedRec methods can be divided into three categories: (1) **Base FedRec**: Early efforts in FedRec focused on adapting traditional recommendation algorithms to the federated setting. For example, FCF [16] adapted Collaborative Filtering (CF) to the FedAvg [36] algorithm, and FedMF [19] used homomorphic encryption to protect gradient privacy in matrix factorization. Neural network-based approaches, such as FedNCF [24], extended Neural Collaborative Filtering (NCF) to FL, while FedPerGNN [25] and FeSoG [29] integrated graph structures into FL to capture high-order user-item correlations. Other studies have concentrated on improving the efficiency of FedRec [54], [55]. (2) **Performance-focused FedRec**: More recently, personalization has become a key focus. PFedRec [26] eliminated shared user embeddings in favor of local score functions, CoFedRec [28] used clustering to group users by preferences, and GPFedRec [27] aggregated user relation graphs for personalized item embeddings. Despite FedRec's ability to achieve good recommendation performance without accessing users' data, (3) **privacy-focused FedRec** [23], [44] has shown that model parameters can "memorize" data features and attributes. As a result, uploading parameters or gradients to the server can still lead to indirect privacy leakage. [21], [22] have demonstrated that uploaded parameters can disclose users' private attributes, even when these attributes are not used in training. [34] and [51], [56] have explored methods to uncover user-item interactions or locations from uploaded parameters.

Our work specifically addresses the indirect privacy leakage of user attributes in FedRec. Although these vulnerabilities have been previously investigated, the development of effective defense mechanisms in FedRec remains in its early stages.

C. Defenses Against Attribute Inference Attacks

Existing defenses against attribute inference attacks in RS can be grouped into the following categories: (1) Adversarial learning [22], [23], [57], [58]: This approach trains the original recommender model with a surrogate attacker. The surrogate attacker attempts infer sensitive attributes, compelling the recommender model to adapt its parameters to thwart such inferences. (2) Differential privacy [21], [53]: It safeguards against attribute leakage by adding noise to either user data before training or model parameters during training. The noise is carefully calculated to mask the true value of sensitive information while still allowing the model to learn relevant patterns. (3) Data modification [59], [60]: This strategy misleads attackers by either injecting dummy interactions or modifying the values of attribute information. (4) Post-training unlearning [30], [32], [61]: After the initial training, this method introduces regularization-based loss (e.g., minimizing distributional distances between attribute groups of user embeddings to make users indistinguishable) during the post-training phase.

Current defense mechanisms, while effective in protecting privacy, often come at the cost of reduced recommendation quality, as user attribute information is unlearned or perturbed from user representations. Our work addresses this issue by introducing a representation decoupling method. This method decouples privacy-relevant and privacy-irrelevant representations, keeping sensitive parameters local and only sharing insensitive parameters with the server. This approach ensures both personalized recommendations and user privacy.

VI. Conclusion and Future Work

In this paper, we present PARD, a novel privacy-aware representation learning framework for federated recommendation systems. PARD introduces a decoupled representation mechanism that decouples representations into privacy-relevant and privacy-irrelevant components. This decoupling allows the model to preserve attribute-related preference signals on privacy-relevant components while suppressing private attribute leakage on privacy-irrelevant components, achieving a better balance between privacy and utility. Extensive experiments on multiple datasets demonstrate the superiority of PARD over state-of-the-art methods in defending against attribute inference attacks while maintaining or improving recommendation performance.

For future work, we plan to explore space-efficient decoupling of item-side representations to further improve the scalability and efficiency of our approach. In addition, we are interested in developing adaptive mechanisms that dynamically adjust the decoupling strength based on attribute sensitivity and client behavior.

REFERENCES

[1] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics reports*, vol. 519, no. 1, pp. 1–49, 2012.

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

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[4] C. Meng, C. Zhai, Y. Yang, H. Zhang, and X. Li, "Parallel knowledge enhancement based framework for multi-behavior recommendation," in *CIKM*. ACM, 2023, pp. 1797–1806.

[5] C. Zhai, C. Meng, Y. Yang, K. Zhang, X. Zhao, and X. Li, "Combinatorial optimization perspective based framework for multi-behavior recommendation," 2025. [Online]. Available: https://arxiv.org/abs/2502.02232

[6] X. Zhao, Y. Zhu, C. Wang, M. Jing, J. Yu, and F. Tang, "Task-difficulty-aware meta-learning with adaptive update strategies for user cold-start recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3484–3493.

[7] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai, "Addressing the item cold-start problem by attribute-driven active learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 631–644, 2020.

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

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

[10] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 203–210.

[11] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[12] X. Rao, R. Jiang, S. Shang, L. Chen, P. Han, B. Yao, and P. Kalnis, "Next point-of-interest recommendation with adaptive graph contrastive learning," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 3, pp. 1366–1379, 2025.

[13] W. Chen, H. Huang, Z. Zhang, T. Wang, Y. Lin, L. Chang, and H. Wan, "Next-poi recommendation via spatial-temporal knowledge graph contrastive learning and trajectory prompt," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 6, pp. 3570–3582, 2025.

[14] Q. Zhang, P. Yang, J. Yu, H. Wang, X. He, S. Yiu, and H. Yin, "A survey on point-of-interest recommendation: Models, architectures, and security," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 6, pp. 3153–3172, 2025.

[15] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[16] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888*, 2019.

[17] I. Hsieh and C. Li, "Netfense: Adversarial defenses against privacy attacks on neural networks for graph data," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 796–809, 2023.

[18] W. Ali, K. Umer, X. Zhou, and J. Shao, "Hidattack: An effective and undetectable model poisoning attack to federated recommenders," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 3, pp. 1227–1240, 2025.

[19] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 11–20, 2020.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[21] S. Zhang, W. Yuan, and H. Yin, "Comprehensive privacy analysis on federated recommender system against attribute inference attacks," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 987–999, 2024.

[22] Q. Hu and Y. Song, "User consented federated recommender system against personalized attribute inference attack," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 276–285.

[23] G. Beigi, A. Mosallanezhad, R. Guo, H. Alvari, A. Nou, and H. Liu, "Privacy-aware recommendation with private-attribute protection using adversarial learning," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 34–42.

[24] V. Perifanis and P. S. Efraimidis, "Federated neural collaborative filtering," *Knowledge-Based Systems*, vol. 242, p. 108441, 2022.

[25] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie, "A federated graph neural network framework for privacy-preserving personalization," *Nature Communications*, vol. 13, no. 1, p. 3091, 2022.

[26] C. Zhang, G. Long, T. Zhou, P. Yan, Z. Zhang, C. Zhang, and B. Yang, "Dual personalization on federated recommendation," *arXiv preprint arXiv:2301.08143*, 2023.

[27] C. Zhang, G. Long, T. Zhou, Z. Zhang, P. Yan, and B. Yang, "Gpfedrec: Graph-guided personalization for federated recommendation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4131–4142.

[28] X. He, S. Liu, J. Keung, and J. He, "Co-clustering for federated recommender system," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3821–3832.

[29] Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu, "Federated social recommendation with graph neural network," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–24, 2022.

[30] W. Wu, J. Jiang, and C. Hu, "Aegis: Post-training attribute unlearning in federated recommender systems against attribute inference attacks," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 3783–3793.

[31] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS*. ACM, 2016, pp. 308–318.

[32] Y. Li, C. Chen, X. Zheng, Y. Zhang, Z. Han, D. Meng, and J. Wang, "Making users indistinguishable: Attribute-wise unlearning in recommender systems," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 984–994.

[33] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine, "Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow," in *ICLR (Poster)*. OpenReview.net, 2019.

[34] W. Yuan, C. Yang, Q. V. H. Nguyen, L. Cui, T. He, and H. Yin, "Interaction-level membership inference attack against federated recommender systems," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1053–1062.

[35] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[37] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "CLUB: A contrastive log-ratio upper bound of mutual information," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 1779–1788.

[38] Y. H. Tsai, Y. Wu, R. Salakhutdinov, and L. Morency, "Self-supervised learning from a multi-view perspective," in *ICLR*. OpenReview.net, 2021.

[39] Q. Pi, W. Bian, G. Zhou, X. Zhu, and K. Gai, "Practice on long sequential user behavior modeling for click-through rate prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2671–2679.

[40] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[41] H. Zhu, X. Li, P. Zhang, G. Li, J. He, H. Li, and K. Gai, "Learning tree-based deep model for recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1079–1088.

[42] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

[43] F. Mireshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, "Privacy in deep learning: A survey," *arXiv preprint arXiv:2004.12254*, 2020.

[44] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing," in *23rd USENIX security symposium (USENIX Security 14)*, 2014, pp. 17–32.

[45] Z. Wang, J. Yu, M. Gao, H. Yin, B. Cui, and S. Sadiq, "Unveiling vulnerabilities of contrastive recommender systems to poisoning attacks," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3311–3322.

[46] M. Zhang, Z. Ren, Z. Wang, P. Ren, Z. Chen, P. Hu, and Y. Zhang, "Membership inference attacks against recommender systems," in *CCS*. ACM, 2021, pp. 864–879.

[47] 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.

[48] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[49] Q. Wang, H. Yin, T. Chen, J. Yu, A. Zhou, and X. Zhang, "Fast-adapting and privacy-preserving federated recommender system," *The VLDB Journal*, vol. 31, no. 5, pp. 877–896, 2022.

[50] D. Zhong, X. Wang, Z. Xu, J. Xu, and W. H. Wang, "Interaction-level membership inference attack against recommender systems with long-tailed distribution," in *CIKM*. ACM, 2024, pp. 3433–3442.

[51] J. Long, T. Chen, G. Ye, K. Zheng, Q. V. H. Nguyen, and H. Yin, "Physical trajectory inference attack and defense in decentralized poi recommendation," in *WWW*. ACM, 2024, pp. 3379–3387.

[52] K. Cai, J. Zhang, Z. Hong, W. Shand, G. Wang, D. Zhang, J. Chi, and Y. Tian, "Where have you been? A study of privacy risk for point-of-interest recommendation," in *KDD*. ACM, 2024, pp. 175–186.

[53] S. Zhang, H. Yin, T. Chen, Z. Huang, L. Cui, and X. Zhang, "Graph embedding for recommendation against attribute inference attacks," in *WWW*. ACM / IW3C2, 2021, pp. 3002–3014.

[54] K. Muhammad, Q. Wang, D. O'Reilly-Morgan, E. Tragos, B. Smyth, N. Hurley, J. Geraci, and A. Lawlor, "Fedfast: Going beyond average for faster training of federated recommender systems," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1234–1242.

[55] J. Yi, F. Wu, C. Wu, R. Liu, G. Sun, and X. Xie, "Efficient-fedrec: Efficient federated learning framework for privacy-preserving news recommendation," *arXiv preprint arXiv:2109.05446*, 2021.

[56] Z. Ye, X. Zhang, X. Chen, H. Xiong, and D. Yu, "Adaptive clustering based personalized federated learning framework for next POI recommendation with location noise," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 5, pp. 1843–1856, 2024.

[57] B. Wang, J. Guo, A. Li, Y. Chen, and H. Li, "Privacy-preserving representation learning on graphs: A mutual information perspective," in *KDD*. ACM, 2021, pp. 1667–1676.

[58] C. Ganhör, D. Penz, N. Rekabsaz, O. Lesota, and M. Schedl, "Unlearning protected user attributes in recommendations with adversarial training," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 2142–2147.

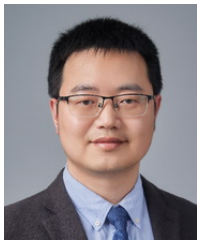
[59] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: inferring and obfuscating user gender based on ratings," in *RecSys*. ACM, 2012, pp. 195–202.

[60] J. Jia and N. Z. Gong, "Attriguard: A practical defense against attribute inference attacks via adversarial machine learning," in *USENIX Security Symposium*. USENIX Association, 2018, pp. 513–529.

[61] C. Chen, Y. Zhang, Y. Li, J. Wang, L. Qi, X. Xu, X. Zheng, and J. Yin, "Post-training attribute unlearning in recommender systems," *ACM Transactions on Information Systems*, vol. 43, no. 1, pp. 1–28, 2024.

VII. Biography Section

Xuhao Zhao received the B.E. degree in computer science from Tongji University, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include services computing, privacy-preserving recommender system, and federated learning.



Yanmin Zhu (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, in 2007. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Before that, he was a Research Associate with the Department of Computing, Imperial College London, U.K. His research interests include big data analytics, recommendation models, and mobile computing.

Wenze Ma received the B.E. degree in IEEE Honor Class (Computer Science) from Shanghai Jiao Tong University, China, in 2021. He is currently a Ph.D. student in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China. His research interests include graph-based representation learning and recommendation systems.

Qihao Luo is currently an undergraduate student majoring in Computer Science at Shanghai Jiao Tong University, China. His research interests include recommendation systems and large language models.

Chenhao Zhai received the B.E. degree in software engineering from Tongji University, China, in 2023. He is currently pursuing the M.S. degree with the Shenzhen International Graduate School, Tsinghua University, China. His research interests include recommender system, and uplift modeling.

Chunyang Wang received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2024. He is currently an assistant professor with the School of Data Science and Engineering, East China Normal University, Shanghai, China. He received his B.E. degree in Computer Science from the Harbin Institute of Technology, China, in 2019. His research interests include data mining and recommendation systems.

Jiadi Yu (Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2007. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Prior to join Shanghai Jiao Tong University, he was with the Stevens Institute of Technology, USA, as a Post-Doctoral Researcher. He has published more than 100 refereed papers in international leading journals and key conferences in the areas of wireless communications and networking, mobile computing, and security and privacy. His current research interests include mobile computing and sensing, cyber security and privacy, the Internet of Things (IoT), and smart healthcare. He is a member of the IEEE Communication Society.

Feilong Tang (Senior Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University (SJTU), China, in 2005. Currently, he is a Professor with the Department of Computer Science and Engineering, SJTU. In past years, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow in Japan. His research interests focus on integrated space and terrestrial networks, mobile cognitive networks, big data analysis, and cloud computing. He is an IET Fellow. He has received five best papers from international conferences. He also received the Distinguished Pu-Jiang Scholars Award from Shanghai Municipality. He served as the program co-chairs for nine international conferences/workshops.