Aegis: Post-Training Attribute Unlearning in Federated Recommender Systems against Attribute Inference Attacks

Wenhan Wu
School of Computer Science
Wuhan University
Wuhan, China
wenhanwu@whu.edu.cn

Jiawei Jiang
School of Computer Science
Wuhan University
Wuhan, China
jiawei.jiang@whu.edu.cn

Chuang Hu
State Key Laboratory of Internet of Things for Smart City
University of Macau
Macau, Macau SAR
chuanghu@um.edu.mo

Abstract

As privacy concerns in recommender systems become increasingly prominent, federated recommender systems (FedRecs) have emerged as a promising distributed training paradigm. FedRecs enable the collaborative training of a shared global recommendation model without requiring the exchange of raw client interaction data. However, models trained using standard FedRec methods remain vulnerable to personal information leakage, particularly through attribute inference attacks, which can expose sensitive user attributes such as gender and race. In this paper, we address these user-sensitive attributes as targets for federated unlearning. To protect users' sensitive information, attribute unlearning aims to eliminate sensitive attributes from user embeddings, thereby preventing inference attacks while preserving recommendation performance. We introduce a novel post-training federated unlearning framework, Aegis, which performs unlearning based on private attribute requests after the model has been trained, minimizing the degradation in recommendation accuracy. Aegis employs an information-theoretic multi-component loss function to balance privacy protection and recommendation performance. Additionally, Aegis adapts to scenarios where training interaction data may be unavailable, reflecting real-world centralized protection scenarios. Comprehensive evaluations of various benchmark datasets demonstrate that our proposed method effectively safeguards user privacy while maintaining high-quality recommendations.

CCS Concepts

• **Information systems** → **Social recommendation**; • **Security and privacy** → *Social network security and privacy*.

Keywords

Federated Learning; Recommender System; Attribute Unlearning.

ACM Reference Format:
Wenhan Wu, Jiawei Jiang, and Chuang Hu. 2025. Aegis: Post-Training Attribute Unlearning in Federated Recommender Systems against Attribute Inference Attacks. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28–May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3696410.3714823

WWW '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714823

1 Introduction

Recommender systems (RS) have become fundamental to modern web applications, driving personalized user experiences across platforms like e-commerce [5, 35] and streaming services [33, 34] by utilizing vast amounts of user interaction data. However, traditional RS often require users' personal data—such as browsing history and purchasing behavior—to be centralized on remote servers for processing, which raises significant privacy risks including the potential data breaches. To address these concerns, federated recommender systems (FedRecs) [2, 51, 52] have emerged as a promising solution that aggregates model updates on the server while keeping user interaction data local and performing model training on the client side. Despite this improvement, a critical challenge persists with stricter privacy requirements: recent privacy regulations such as the GDPR [60] and CCPA [28] emphasizing the *"right to be forgotten"* has promoted users' demand for the forgetting of personal private attributes in recommender systems, i.e. *attribute unlearning* [22]. This concept pertains to the removal of the inherent attributes of user embeddings in the trained model, such as race and gender, which are not used as training targets.

Attribute unlearning is intuitively important for privacy protection in recommender systems. Although existing FedRecs avoid the transmission of raw data, model training based on historical user behavior may still be vulnerable to *attribute inference attacks (AIAs)* [4, 32], which potentially reveal sensitive user attributes [20, 42]. Research [63] has demonstrated that basic machine learning models can successfully infer user attributes from user embeddings learned by collaborative filtering models. We conducted attribute inference attacks on FedNCF [55] systems trained on dataset MovieLens-100K (ML-100K) and MovieLens-1M (ML-1M) [23]. As shown in Table 1, the accuracy of attribute inference attackers is consistently higher than that of random attackers, revealing significant privacy leakage risks.

Some existing machine unlearning methods [6, 67] aim to make the unlearned model as consistent as possible with one retrained from scratch. However, in the context of attribute unlearning, this fails to decouple latent attributes from the model, hindering effective unlearning. Unlike regular machine forgetting, attribute unlearning cannot simply erase specific attribute traces. Moreover, most current approaches to privacy protection operate during training, relying on network modifications [22, 27, 71, 72] or adversarial training [3, 18, 20, 40, 49]. These in-training attribute-preserving methods are costly, complex, and require prior knowledge of privacy issues, making them less suited to dynamic privacy needs. In real-world scenarios, users' privacy requirements may change over time, and federated clients may want to adjust their privacy settings after training rather than determining them beforehand. This calls for a post-training federated attribute unlearning method that can handle dynamic requests without needing full retraining or redesign of the existing model structure. Furthermore, attribute unlearning requests are often unpredictable, and training data or historical updates may be inaccessible due to privacy regulations or data deletion [11]. Thus, federated attribute unlearning must work both with and without access to the interaction training data. In conclusion, a flexible and efficient post-training framework is essential to address evolving privacy demands in federated recommender system environments.

Existing methods protect privacy by artificially designed adding noise to user embeddings, such as through local differential privacy [1], but this often degrades recommendation performance [42]. Additionally, ensuring unlearning effectiveness is challenging, as attacks may come from complex machine learning or deep learning models [47], whose mechanisms are not fully understood [31]. In this paper, we focus on protecting trained user embeddings from potential attacks with two key objectives: i). making private attributes indistinguishable in the embeddings to reduce the success of inference attacks, and ii). preserving recommendation performance, as both users and service providers seek to avoid significant decline in quality. These objectives guide the design of our system.

To achieve these objectives, we propose the Aegis[1] framework, an innovative approach that seamlessly integrates attribute unlearning with performance retention. Our method fine-tunes a pre-trained recommender system to safeguard sensitive user attributes from attribute inference attacks. It employs a multi-component loss function grounded in information theory [58] to address the optimization problem. Specifically, it reduces the association between user embeddings and sensitive attributes while maintaining recommendation performance. Additionally, it includes a regularization component to ensure stability in user embeddings. Our framework not only facilitates localized training for attribute unlearning but also adapts to scenarios where client interaction data is inaccessible through centralized unlearning methods. We summarize our main contributions as follows:

- To the best of our knowledge, we are the first to investigate the post-training attribute unlearning in federated recommender systems, addressing scenarios both with and without access to training interaction data. This approach reflects more realistic privacy-preserving measures.
- We propose the Aegis framework for federated attribute unlearning, formalizing the setting of attribute attacks and identifying two key objectives: attribute indistinguishability and recommendation performance retention. Aegis leverages information-theoretic principles by introducing a multi-component loss function that synchronously optimizes both objectives, balancing privacy and recommendation accuracy.
- We implemented the Aegis system and conducted extensive evaluation on benchmark datasets to evaluate our method's performance in terms of attribute unlearning and recommendation knowledge retention. Results demonstrate that our framework effectively balances privacy and performance.

Table 1: FedRec Recommendation Utility and Attribute Inference Attack Results on Different Datasets

Dataset	Utility		Attribute Privacy		
	NDCG@10	HR@10	Gender	Age	Occupation
ML-100K	0.708	0.680	0.714	0.280	0.149
ML-1M	0.699	0.684	0.849	0.353	0.119
Random Attacker			0.500	0.143	0.048

2 Preliminaries and Objectives

2.1 Federated Recommendation

We first describe the general mathematical formulation of a federated recommender system. Let U represent the set of users with the total number of users denoted as $|U|$, and V represent the set of items with the total number of items denoted as $|V|$. Users across different clients collectively form the U. Each user $u_i \in U$ $(1 \le i \le |U|)$ owns a local dataset D_i, which is defined as: $D_i = \{(u_i, v_j, r_{ij}) | v_j \in V\}$, where $r_{ij} = 1$ indicates that user u_i has interacted with item v_j, and $r_{ij} = 0$ means no interaction, in which case v_j is considered a negative sample. The goal of the federated recommender system is to predict the score $\hat{r}_{ij} = s_\psi(\mathbf{em}_i, \mathbf{em_j})$ of user u_i on non-interacted items v_j, thereby generating a recommendation list $\hat{V}_i$, satisfying:

$$\hat{V}_i = \text{Top-K}(\{\hat{r}_{ij} | v_j \in V \setminus D_i\}), \tag{1}$$

where $s_\psi(\cdot)$ is a score function, which can be a dot product, a multi-layer perceptron, etc. $\mathbf{em}_i = f_p(f_\varphi(u)) \in \mathbb{R}^d$ and $\mathbf{em}_j = f_p(f_\varphi(j)) \in \mathbb{R}^d$ represent the embeddings of users and items, where d is the embedding dimension. The function f_φ represents an embedding layer that maps users/items to vectors, and f_p represents a propagation layer that captures collaborative signals [66]. FedRecs train the model across multiple distributed clients, such as users' mobile devices or computers. The central server does not directly access users' interaction data but instead coordinates multiple rounds of local training on each client and aggregates the model parameter updates uploaded by each client to form a global model. Specifically, in each global training round t: First, the central server distributes the global model parameters to each selected user/client i, $\theta_{i,t}^0 = \theta_{g,t}$. Each client combines the received global parameters with their local user embeddings to form a local recommender model. Then, the local recommender is optimized using the local dataset as below:

$$\theta_{i,t}^{t_l+1} = \theta_{i,t}^{t_l} - \eta \nabla_{\theta_i} \mathcal{L}(\theta_{i,t}, D_i), \tag{2}$$

where η is the learning rate, and $\mathcal{L}$ is the loss function (e.g., BPR loss [56]). After T_l rounds of local training, each client sends the updated global parameters $\theta_{i,t}^{T_l}$ (or the parameter updates $\Delta\theta_{i,t}^{T_l} = \theta_{i,t}^{T_l} - \theta_{i,t}^0$) back to the server. The server aggregates the received parameter using a specific aggregation strategy [50], such as:

$$\theta_{g,t+1} = \theta_{g,t} + \frac{1}{|U|} \sum_{i=1}^{|U|} \Delta\theta_{i,t}^{T_l} \quad \text{or} \quad \theta_{g,t+1} = \frac{1}{|U|} \sum_{i=1}^{|U|} \theta_{i,t}^{T_l}. \tag{3}$$

The above steps are repeated until the convergence condition is met, i.e., the performance of the model reaches the predefined standard.

[1] From Greek mythology, a powerful shield used by Zeus or Athena to fend off attacks.

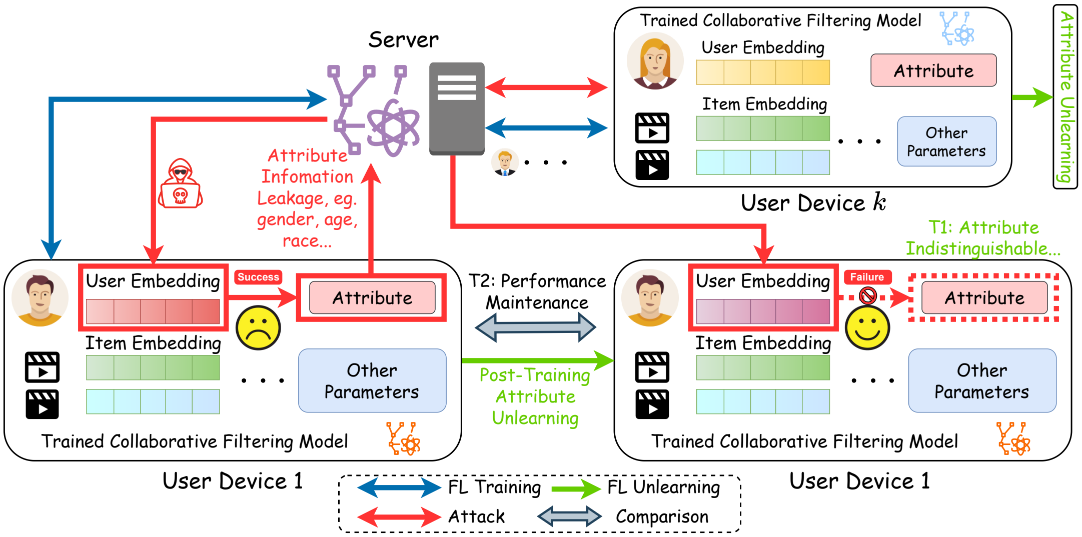

Figure 1: Overview of Post-Training Attribute Unlearning in Federated Recommender Systems.

2.2 Attribute Inference Attack

In recommender systems, attackers can infer private user attributes (such as gender, age, or race) based on the embedding data between users and items, as well as the trained model [20, 42]. This leads to attribute inference attacks (AIA) [4, 32, 71], which pose a significant threat to user privacy. A key issue in AIA is the unintentional leakage of information about non-target attributes during the recommendation process. To mitigate this risk, the concept of attribute unlearning has been introduced, which allows a system to "forget" certain sensitive attributes post-training while maintaining recommendation performance.

Threat Model: The threat model is illustrated in Figure 1: in federated learning (FL) environments, malicious servers may infer users' sensitive attributes by accessing user embedding vectors from FL clients, leading to privacy breaches. In this scenario, we assume the attacker adopts a grey-box attack strategy, meaning the attacker cannot access all model parameters but can access some user embedding vectors $\mathbf{em}_i$ and corresponding attribute information z_i. In practice, a malicious client may be interested in all users' private attributes. This client could reach an agreement with the server and intentionally leak a portion of its users' private attributes, or this information might be inadvertently leaked. Alternatively, a malicious server could collaborate with a curious client, where the client provides the server with private attribute data in exchange for some financial benefit. The server would then use this client's embeddings and corresponding private attributes to train a threat model. Since the server may have access to other clients' local models, it can use this trained model to infer the private attributes of users from other clients, leading to large-scale privacy leakage.

The threat model's goal is to infer the private attribute z_i from the user embedding $\mathbf{em}_i$. In this paper, this attack is framed as a classification problem, where the attacker employs a classification model g to predict the private attributes.

During the training phase of the threat model, we assume that the attacker does not have direct access to the original dataset. Instead, the attacker uses a shadow dataset D_{shadow} to train the model. This shadow dataset can be generated by sampling from the original user data or from other users within the same distribution [42, 57]. The input to the shadow dataset consists of user embeddings $\mathbf{Em}_{\text{shadow}}$, while the target comprises the attribute labels Z_{shadow}. Although using a shadow dataset may reduce the overall effectiveness of the attack, this assumption is reasonable, as assuming access to the full dataset would be overly idealistic and impractical. During training, the attacker constructs the threat model g by minimizing the classification loss on the shadow dataset:

$$\min_{g} \mathbb{E}_{(\mathbf{Em}_{\text{shadow}}, Z_{\text{shadow}})} [\mathcal{L}(Z_{\text{shadow}}, g(\mathbf{Em}_{\text{shadow}}))]. \tag{4}$$

In the inference phase, the attacker uses the trained model g to predict the attribute for a new user embedding $\mathbf{em}_i$. This results are in the prediction $\hat{Z}_i = g(\mathbf{Em}_i)$, where $\hat{Z}_i$ represents the attacker's estimated value of the private attributes of the user set $\mathbf{Em}_i$.

2.3 Our Objectives

In FedRecs, we collaboratively train a collaborative filtering model using data distributed across different clients in a privacy-preserving manner. Notice that users have dynamic privacy preferences, with the private attribute set A_u. Given a trained global model, we aim to perform "unlearning" on the private attribute set A_u for all the users $u_i \in U$, i.e., to generate new user embeddings $\mathbf{em}'_i$ from the original user embeddings $\mathbf{em}_i$ to mask the private attribute. To achieve this, we need to strike an optimal balance between privacy protection and recommendation performance. We formalize federated attribute unlearning as an optimization problem as follows:

# **Objective 1: Unlearning Objective**

Ensure that the user attribute au_i cannot be easily distinguished from different user embeddings, thereby protecting the privacy of the attribute information from potential attackers. We define a function dg to evaluate the distinguishability of the attributes:

$$\min_{\mathbf{em}'_i} \sum_{au_j \in A_u} D(dg(\mathbf{em}'_i), au_j), \tag{5}$$

where D is a measure of attribute distinguishability.

# **Objective 2: Recommendation Objective**
Ensure the performance of the FedRecs is maintained to avoid impacting the original recommendation quality. We can measure the change in recommendation performance as below:

$$\min_{\mathbf{em}'_i} \quad \mathrm{Dist}(M(\mathbf{em}_i), M(\mathbf{em}'_i)), \tag{6}$$

where Dist is a measure of the change in performance.

Additionally, to ensure the efficiency of the unlearning process, we need to limit the time overhead of unlearning. Combining the above objectives, we provide a systematic approach to achieve "unlearning" of user attributes while ensuring a high balance between privacy protection and recommendation performance.

3 Post-training Federated Attribute Unlearning for Recommendation

In this section, we demonstrate how Aegis achieves attribute indistinguishability through model fine-tuning with a carefully designed multi-component federated loss function.

3.1 Overview

Our Aegis framework is a complement to common FedRecs and is applicable to systems based on different training methods. Given the dynamic private attribute needs of federation clients, Aegis fine-tunes a trained federated recommender system to protect sensitive attributes from attribute inference attacks.

Aegis advocates two key properties for effective attribute unlearning: i) **Private Attribute Unlearning**, which effectively removes the association between user-marked attributes for deletion and user embeddings to prevent privacy leakage; and ii) **Recommendation Knowledge Retention**, which ensures that recommendation performance is maintained post-unlearning. Aegis follows a client-server architecture in federated learning, where clients update user embeddings based on unlearning methods using stored interaction data to eliminate the relationship between sensitive attributes and embeddings. The updated embeddings and other model parameters are then uploaded to the server, where FedAvg gradient aggregation [50] is performed, and the updated global model is sent back to the clients for iterative optimization. When post-training attribute unlearning is required but client training data is unavailable, Aegis also allows centralized unlearning to be executed solely on the client side.

3.2 Compositional Attribute Unlearning

To achieve attribute unlearning in privacy-sensitive scenarios, we propose an information theory [58] based multi-component loss function, treating it as an optimization problem for user embeddings. The core idea is to identify the mutual information between the embedding distribution and the attributes, effectively capturing their relationship, and then applying forgetting based on the privacy requirements. Our approach involves designing an appropriate loss function and using optimization techniques to obtain the desired user embeddings, temporarily disregarding intermediate operations and transformations. The goal is to ensure that the model's embedding $\mathbf{em}'_i$ achieves the following: i) Discard information related to specific (e.g., sensitive or private) attributes. ii) Retain information relevant to the recommendation task. This objective can be formalized as:

- Minimize the mutual information between the user embedding $\mathbf{em}'_i$ and the attributes to be forgotten $au_j \in A_u$, defined as:

$$\min_{\mathbf{em}'_i} \sum_{au_j \in A_u} \mathcal{I}(\mathbf{em}'_i; au_j); \tag{7}$$

- Maximize the mutual information between the embedding $\mathbf{em}'_i$ of user u_i and the primary recommendation task, defined as:

$$\max_{\mathbf{em}'_i} \mathcal{I}(\mathbf{em}'_i, V_{EM}; R_i); \tag{8}$$

where V_{EM} represents the item embeddings of items V and $R_i = [r_{i,j}]_{j=1}^{|V|}$ represents the interaction matrix between u_i and V.

We propose a multi-component loss function, with each component specifically designed to address one of the above goals.

3.2.1 *Private Attributes Information Loss.*

Directly computing mutual information $\mathcal{I}(\mathbf{em}'_i; au_j), au_j \in A_u$ is challenging because it requires estimating joint and marginal probability distributions, a process that becomes computationally expensive in high-dimensional embedding spaces. This difficulty arises due to unknown real distributions, complex non-linear dependencies, and the curse of dimensionality. Therefore, the exact estimation of mutual information is impractical without the use of approximation techniques. To address this, we approximate the mutual information using a variational upper bound based on the Kullback–Leibler (KL) divergence [38], which measures the difference between two probability distributions. The resulting loss function is as follows:

$$\begin{aligned}\mathcal{L}^{\mathrm{AU}}_{i,j} &= I(\mathbf{em}_i; au_j)\\ &\approx \sum\nolimits_{k=1}^{|U_{au_j}|} \frac{|S_{au_j=C_k}|}{|S_{au_j}|} D_{\mathrm{KL}}(q_\phi(\mathbf{em}_i|X_{au_j=C_k}) \| p(\mathbf{em}_i)),\end{aligned} \tag{9}$$

where $|S_{au_j}|$ represents the number of users on the current client whose attribute au_j equals C_k and $X_{au_j=C_k}$ represents the input X where the label of attribute au_j is C_k. $D_{\mathrm{KL}}(q\|p)$ denotes the Kullback–Leibler divergence between the variational distribution $q_\phi(\mathbf{em}_i|X_{au_j=C_k})$, which approximates the embedding distribution conditioned on the attribute $au_j = C_k$, and the marginal distribution $p(\mathbf{em}_i)$, which represents the embedding distribution conditioned only on the input X. By minimizing this KL divergence, we effectively reduce the information in the embedding $\mathbf{em}_i$ that is related to the attribute au_j, thus achieving the goal of unlearning the attribute. Intuitively, we aim to compute the distribution of different attribute classes au_j and minimize their KL divergence from the same global distribution, ensuring that the embedding data of users across different classes cannot be identified. To efficiently compute the distribution of user embeddings for the $|U_{au_j}|$ distinct classes within au_j and subsequently calculate the KL divergence, we perform the following operations on each client:

First, we compute the user embedding distribution for each class. Since Aegis is a post-training method, the user embedding data is already available before the unlearning process begins, with each class having an associated set of user embeddings. The probability distribution of embeddings for each class can be estimated, for instance, by fitting a Gaussian distribution or another suitable model [14, 17, 24, 62]. In this work, as our user embeddings are represented in a continuous vector space, we fit a Gaussian distribution to each class's embedding distribution. For each attribute class $C_k \in au_j$, the mean vector $\mu_{j,k}$ is computed as below:

$$\mu_{j,k} = \frac{1}{|S_{au_j=C_k}|} \sum\nolimits_{\mathbf{em}_i \in S_{au_j=C_k}} \mathbf{em}_i, \tag{10}$$

where $S_{au_j=C_k} = \{\mathbf{em}_i \mid au_j(\mathbf{em}_i) = C_k\}$ represents the set of user embeddings where the attribute au_j belongs to class C_k, and $|S_{au_j=C_k}|$ denotes the number of embeddings in this set. We compute the covariance matrix Σ_i of the Gaussian distribution as below:

$$\Sigma_{j,k} = \frac{1}{|S_{au_j=C_k}|} \sum\nolimits_{\mathbf{em}_i \in S_{au_j=C_k}} (\mathbf{em}_i - \mu_{j,k})(\mathbf{em}_i - \mu_{j,k})^T. \tag{11}$$

Second, we compute the user embedding distribution for the global set of embeddings by aggregating all the embeddings across classes. The global mean vector μ_{global} is computed as:

$$\mu_{\text{global}} = \frac{|S_{au_j=C_k}|}{N} \sum\nolimits_{k=1}^{N} \mu_{j,k}, \tag{12}$$

where N is the total number of user embeddings across all classes. Similarly, the global covariance matrix Σ_{global} is computed as:

$$\Sigma_{\text{global}} = \frac{1}{N} \sum\nolimits_{i=1}^{N} (\mathbf{em}_i - \mu_{\text{global}})(\mathbf{em}_i - \mu_{\text{global}})^T. \tag{13}$$

This global Gaussian distribution captures the overall structure of the user embeddings across all attribute classes.

Finally, we calculate the KL divergence between each class embedding distribution and the global embedding distribution. Notably, there is an analytical solution for computing the KL divergence between two multivariate Gaussian distributions. For a class C_k and the global Gaussian distributions $\mathcal{N}(\mu_{j,k}, \Sigma_{j,k})$ and $\mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})$, the formula for the KL divergence is:

$$\begin{aligned} D_{\text{KL}}(\mathcal{N}(\mu_{j,k}, \Sigma_{j,k}) \| \mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})) &= \frac{1}{2}\left(\log \frac{\det(\Sigma_{\text{global}})}{\det(\Sigma_{j,k})} - d\right) \\ + \frac{1}{2}\left(\text{Tr}(\Sigma_{\text{global}}^{-1}\Sigma_{j,k}) + (\mu_{\text{global}} - \mu_{j,k})^T \Sigma_{\text{global}}^{-1}(\mu_{\text{global}} - \mu_{j,k})\right), \end{aligned} \tag{14}$$

where d is the dimensionality of the user embeddings, $\det(\Sigma)$ is the determinant of the covariance matrix, and Tr denotes the trace operation. By fitting a Gaussian distribution to the embeddings, we compute the mean and covariance matrix for each class and then use the KL divergence to minimize the difference between the user embedding distributions of different classes. This allows us to calculate the unlearning loss function $\mathcal{L}_j^{AU}$ for au_j as below:

$$\mathcal{L}_j^{AU} = \sum\nolimits_{k=1}^{|U_{au_j}|} \frac{|S_{au_j=C_k}|}{|S_{au_j}|} D_{\text{KL}}(\mathcal{N}(\mu_{j,k}, \Sigma_{j,k}) \| \mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})). \tag{15}$$

Minimizing the loss function L_j^{AU} enables the unlearning of the private attribute au_j, thereby providing protection against AIAs.

3.2.2 ***Recommendation Knowledge Retention Loss.*** Since performing unlearning may lead to a degradation in recommendation performance, additional design is necessary to achieve objective 2. An intuitive approach is to directly use the recommendation loss function from the federated training phase (e.g., binary cross-entropy (BCE) [68], root mean squared error (RMSE) [29], or Bayesian personalized ranking (BPR) [56] loss) as the optimization objective to maintain recommendation quality. This recommendation loss $\mathcal{L}^{Rec}$ can be defined as:

$$\mathcal{L}^{Rec} = \mathcal{L}_{BCE/BPR,...}(s_\psi(f_{\varphi,p}(u), f_{\varphi,p}(i)), \mathbf{R}), \tag{16}$$

where $\mathbf{R}$ is the interaction matrix, and each element $r_{i,j} \in \mathbf{R}$ represents the interaction between user u_i and item v_j.

To accelerate the execution process, we only update user embeddings during unlearning, so we additionally propose the use of a regularization loss $\mathcal{L}^{Reg}$ to restrict the range of user embedding updates, preventing drastic changes in user embeddings and thus leveraging the prior learning. The L2-regularization [21, 42] term is defined as below:

$$\mathcal{L}^{Reg} = \sum\nolimits_{i=1}^{|U|} \|\mathbf{em}_i - \mathbf{em}'_i\|_2^2 = \sum\nolimits_{i=1}^{|U|} \sum\nolimits_{j=1}^{d} (em_{i,j} - em'_{i,j})^2, \tag{17}$$

where $\mathbf{em}_i$ and $\mathbf{em}'_i$ represent the user embeddings before and after unlearning, respectively. Since the interaction data may be inaccessible due to privacy restrictions or data modifications after training, the loss $\mathcal{L}^{Rec}$ might no longer be applicable. In such scenarios, the regularization term $\mathcal{L}^{Reg}$ will help preserve recommendation performance. The underlying rationale is that closer model parameters typically lead to more consistent model performance.

3.2.3 ***Summary.*** Eq. (15), (16), and (17) in Aegis represent two sub-objectives corresponding to the motivations: Eq. (15) focuses on the elimination of sensitive attribute information, and Eq. (16) and Eq. (17) aim to enhance recommendation performance. For each user, we achieve private user embedding training for each client through a federated training process. In each unlearning round, we sample a set of users and their historical interaction data from U. By jointly learning $\mathcal{L}^{AU}$, $\mathcal{L}^{Rec}$, and $\mathcal{L}^{Reg}$, the private embedding unlearning objective for clients can be formulated as:

$$\varphi, p = \arg\min_{\varphi,p} \mathcal{L}^{All} = \arg\min_{\varphi,p} \mathcal{L}^{Rec} + \beta\mathcal{L}^{Reg} + \gamma \sum\nolimits_{j \in A_u} \mathcal{L}_j^{AU}, \tag{18}$$

where β, and γ are hyperparameters that balance the trade-off between recommendation utility and privacy protection. A larger γ indicates stronger protection of attribute privacy, while a larger β enhance recommendation accuracy. It is worth noting that the weight for the privacy loss of each attribute can be adjusted; for simplicity, we assume users weigh all private attributes equally. When the training interaction data is unavailable and the recommendation loss cannot be applied, we only use the regularization term to prevent excessive degradation of recommendation performance. The training objective in this data-free case is:

$$\varphi, p = \arg\min_{\varphi,p} \mathcal{L}^{All'} = \arg\min_{\varphi,p} \beta\mathcal{L}^{Reg} + \gamma \sum\nolimits_{j \in A_u} \mathcal{L}_j^{AU}. \tag{19}$$

Finally, we use stochastic gradient descent (SGD) [36] on each client to optimize the total loss function $\mathcal{L}^{All}$ or $\mathcal{L}^{All'}$.

3.3 Training Process

Aegis operates after federated learning, focusing on fine-tuning to ensure privacy by unlearning specific attributes. Two models are involved in our framework: the embedding network for recommendation $f_{p,\varphi}$ and the recommendation score function s_{ψ}.

3.3.1 *Federated Learning (Pre-Unlearning Stage).*

We adopt the standard FedRec model to train the recommender system. In each training round, the central server begins by sampling a group of users and distributing the model parameters. The clients then perform local training, iterating over mini-batches to compute the loss and update the weights of $f_{p,\varphi}$ and s_{ψ}. After L_l local iterations, the clients upload the weights to the server. Finally, the server aggregates the weights using FedAvg [50] and updates the model.

3.3.2 *Attribute Unlearning Fine-tuning.*

As users' privacy preferences may change over time, when attributes that were previously considered non-sensitive become sensitive, Aegis needs to promptly provide protection. Based on the set of private attributes that need protection, we perform post-training fine-tuning of the trained recommender system. Aegis operates in two protection modes as below:

i) Aegis-Fed (federated protection): When clients are willing to participate in the fine-tuning process and provide the relevant recommendation data for unlearning, we use $\mathcal{L}^{\text{All}}$ from Eq. (18) as the local loss function and perform local training updates. The server handles aggregation and model distribution, following the usual federated training process.

ii) Aegis-CS (centralized protection): When clients are unwilling to participate in the fine-tuning process or are unable to provide the necessary recommendation data due to privacy concerns or data changes, we use $\mathcal{L}^{All'}$ from Eq. (19) as the loss function to conduct fine-tuning on the client making the unlearning request. We iteratively update the user embeddings until the update difference is less than a threshold ϵ, i.e., $\|e_{u,k+1} - e_{u,k}\| \leq \epsilon$.

4 Evaluation

4.1 Evaluation Setup

Testbed. We implemented Aegis using Python 3.8.0 and PyTorch 2.2.0, and ran all experiments on NVIDIA A100 Tensor Core GPUs.

Datasets. The experiments were conducted on four publicly accessible datasets as detailed in Table 2, each representative of various web applications. These include the movie rating datasets MovieLens-100K and MovieLens-1M [23] used in media streaming platforms, the clothing sales dataset ModCloth [61] relevant to e-commerce applications, and the music listening behavior dataset [7] employed in music recommendation services. The datasets encompass user-item interactions as input data, along with user attributes such as gender and age, making them well-suited for RS research. The private attribute 'Age' is divided into seven age groups following the method used in the MovieLens-1M. For location labels, we use continent tags based on the countries from the Last.FM-1K dataset.

FedRec Models and Hyperparameters. We use FedNCF [55] as our foundational FedRecs and employ dot product as the scoring function. We set the dimensions of user and item embeddings to 128 and use SGD [36] as the optimization algorithm with a learning rate of 0.01. For the base training loss function $\mathcal{L}^{Rec}$, we use BPR loss [57] to train the recommendation model. We use 10 federated clients, with default hyperparameters set to $\beta = 0.1$, and $\gamma = 10$.

Table 2: Summary of Datasets

Dataset	Users	Items	Ratings	Density
MovieLens-100K	943	1,682	100,000	6.30%
MovieLens-1M	6,040	3,952	1,000,209	4.19%
ModCloth	44,784	1,020	99,893	0.22%
Last.FM-1K	992	176,948	19,150,868	10.91%

Attacker Seeting. For selecting the attribute inference model for user embedding attacker, we utilize easily implementable and powerful machine learning models, including a three-layer MLP model [53] and the XGBoost model [16]. Both models are employed as private attribute classifiers and trained on shadow datasets.

Evaluation Metrics. In evaluating recommendation performance, we employ metrics widely used in recommender systems, reporting recommendation utility by calculating the average hit ratio (HR) [12] and normalized discounted cumulative gain (NDCG) [30] across the ranked item lists of all test users. We truncate the ranked lists for both metrics at positions 5 and 10. For privacy-preserving performance evaluation, we assess information leakage in user embeddings using the accuracy of attribute classifiers. The AIA's goal is high attack accuracy, but excessively low accuracy could trigger the "Streisand Effect" [10, 11, 19], inadvertently leading to privacy exposure. Our goal is to protect against AIAs, where scores closer to those of a random attacker indicate better privacy preservation.

Unlearning Methods. There are numerous studies on federated unlearning, but most of them are not applicable to the attribute unlearning problem. To the best of our knowledge, we are the first to study post-training federated attribute unlearning. We introduce two versions of Aegis: Aegis-Fed and Aegis-CS. The key difference lies in whether client interaction training data is involved during the unlearning process, corresponding to data-dependent **(DD)** and data-free **(DF)** settings, respectively. We compare our methods with existing defenses against attribute inference attacks:

- **UC-FedRec [27]**: A federated AIA defense method performed during training. It modifies the original federated recommender system by training attribute filters for each client, minimizing the attribute classification loss during training to achieve attribute unlearning. Although the background setting differs from our post-training approach, the comparison aids in a comprehensive understanding of the attribute unlearning problem.
- **U2U-R** and **D2D-R [42]**: The post-training attribute unlearning methods exclude training data, using user-to-user (U2U) loss and distribution-to-distribution (D2D) distance loss as attribute distinguishability losses to achieve unlearning, respectively. We extend it to the federated setting with multiple attribute labels.

4.2 Results and Analysis

4.2.1 *Attribute Unlearning Performance.*

The classification accuracy of attackers across different datasets reflects Aegis's performance of attribute unlearning, with results shown in Table 3. We treat the gender, age groups, and occupation of users in MovieLens-100K and MovieLens-1M, the body shape of users in ModCloth, and

Table 3: Results of Unlearning Performance (Attack Accuracy of XGBoost/MLP Attackers), where DD Indicates the Unlearning Process is Dependent on the Training Interaction Data, and DF Indicates that the Unlearning Process is Interaction Data-free.

Dataset			MovieLens-100K			MovieLens-1M			ModCloth	Last.FM-1K		
Sensitive Attributes			Gender	Age	Occupation	Gender	Age	Occupation	Body Shape	Gender	Age	Location
XGBoost Attacker		Original	0.7143	0.2804	0.1490	0.8487	0.3526	0.1192	0.7419	0.5989	0.4828	0.5604
	DD	Aegis-Fed	**0.5703**	**0.2222**	**0.1058**	**0.5968**	**0.1798**	**0.0589**	**0.5316**	**0.4798**	**0.2778**	**0.3586**
		UC-FedRec	0.6772	0.2857	0.2011	0.7280	0.3220	0.1543	0.7325	0.5606	0.2980	0.4899
	DF	Aegis-CS	**0.5450**	**0.2116**	**0.0794**	**0.6450**	**0.2064**	**0.0766**	**0.5709**	0.4506	**0.2929**	**0.3766**
		U2U-R	0.9921	0.5947	0.6980	0.9997	0.9998	0.9940	0.9999	0.9999	0.8690	0.9518
		D2D-R	0.5834	0.2464	0.1020	0.6821	0.2941	0.0923	0.7799	**0.4746**	0.4585	0.4876
MLP Attacker		Original	0.7105	0.3386	0.0954	0.7310	0.3470	0.1159	0.7654	0.6102	0.5350	0.5604
	DD	Aegis-Fed	0.6541	**0.1376**	0.0794	**0.5625**	**0.1598**	0.0762	**0.6602**	**0.5556**	**0.1162**	0.2828
		UC-FedRec	**0.6085**	0.1640	**0.0582**	0.3654	0.1836	**0.0263**	0.6784	0.4242	0.1364	**0.2171**
	DF	Aegis-CS	**0.6224**	**0.1693**	0.0688	**0.6032**	**0.1821**	**0.0389**	**0.6462**	**0.5253**	**0.0690**	**0.2677**
		U2U-R	0.6931	0.2646	0.0370	0.7193	0.0704	0.0985	0.7107	0.5650	0.3536	0.4088
		D2D-R	0.6720	0.2434	**0.0529**	0.6987	0.2359	0.0298	0.6728	0.5480	0.3103	0.2802
Random Attacker			0.5000	0.1429	0.0476	0.5000	0.1429	0.0476	0.5000	0.5000	0.1469	0.1667

Table 4: Utility Results of Recommendation Performance.

Datasets	Methods		Utility Metrics			
			NDCG@5	NDCG@10	HR@5	HR@10
MovieLens-100K		Original	0.7452	0.7080	0.7234	0.6804
	DD	Aegis-Fed	**0.7632**	**0.6905**	**0.7321**	**0.6959**
		UC-FedRec	0.6959	0.6452	0.7032	0.6698
	DF	Aegis-CS	**0.7209**	**0.6891**	0.6944	0.6571
		U2U-R	0.7151	0.6854	0.7040	0.6663
		D2D-R	0.7194	0.6862	**0.7129**	**0.6693**
MovieLens-1M		Original	0.6992	0.6901	0.6958	0.6843
	DD	Aegis-Fed	**0.6332**	0.6262	**0.6292**	**0.6211**
		UC-FedRec	0.6320	**0.6525**	0.6280	0.6204
	DF	Aegis-CS	0.6631	**0.6707**	**0.6641**	0.6604
		U2U-R	0.6481	0.6156	0.6420	0.5965
		D2D-R	**0.6928**	0.6623	0.6388	**0.6929**
ModCloth		Original	0.6077	0.6079	0.6047	0.6115
	DD	Aegis-Fed	**0.6044**	**0.6071**	**0.6352**	**0.6386**
		UC-FedRec	0.5664	0.5551	0.5622	0.5542
	DF	Aegis-CS	**0.5969**	**0.5814**	**0.5563**	**0.5530**
		U2U-R	0.5605	0.5437	0.5482	0.5319
		D2D-R	0.5854	0.5694	0.5653	0.5561
Last.FM-1K		Original	0.5724	0.5665	0.5806	0.5680
	DD	Aegis-Fed	**0.5888**	**0.5939**	**0.5962**	**0.5999**
		UC-FedRec	0.5182	0.5190	0.5243	0.5229
	DF	Aegis-CS	**0.5446**	0.5726	**0.5442**	0.5052
		U2U-R	0.5182	0.5190	0.5244	0.5229
		D2D-R	0.5282	**0.5975**	0.5362	**0.5490**

the gender, age groups, and country location of users in Last.FM-1K as sensitive attributes. We can draw the following conclusions: First, for FedRec without any protective measures, the XGBoost attack on the original user embedding achieves an average improvement of 21.54%, and the MLP attack improves by 21.15% compared to a random attacker. This indicates that private information in user embeddings can be leaked to attackers. Second, in the data-dependent DD scenario, for MLP attackers, both Aegis-Fed and UC-FedRec methods reduce MLP attack performance, with a reduction of 14.53% and 14.63%, respectively. For XGBoost attackers, Aegis-Fed performs better than UC-FedRec, reducing attack performance by 14.26% and 3.99%, respectively. This is because UC-FedRec's reliance on the MLP-based attribute filter limits its defense against XGBoost attackers. Moreover, Aegis, as a post-training privacy protection method, offers more flexibility without the need to modify the model modules. Third, in the data-free DF centralized protection scenario, Aegis-CS outperforms U2U-R and D2D-R. For MLP attackers, Aegis-CS reduces attack performance by 14.43%, compared to 7.22% and 10.30% reductions for U2U-R and D2D-R, respectively. For XGBoost attackers, U2U-R fails to deceive attackers and instead significantly increases attack performance, with an average increase of 90.99%. In contrast, Aegis-CS and D2D-R reduce attack performance by 12.94% and 5.96%, respectively.

4.2.2 ***Recommendation Performance.*** The recommendation performance evaluation based on Normalized Discounted Cumulative Gain (NDCG) and Hit Rate (HR) is shown in Table 4. We find that attribute unlearning methods can impact recommendation performance to varying degrees. Specifically, in the data-free DF centralized protection scenario, Aegis-CS results in an average reduction of 2.48% and 1.48% for NDCG@5 and NDCG@10, and a 3.63% and 4.21% decrease in HR@5 and HR@10, respectively. U2U-R exhibits larger declines across these four metrics, averaging 4.57%, 5.22%, 4.65%, and 5.67%. In contrast, D2D-R shows comparable performance to Aegis-CS, with average decreases of 2.47%, 1.43%, 3.78%, and 1.92%, respectively, outperforming U2U-R. In the DD scenario, Aegis-Fed achieves better recommendation performance than Aegis-CS, with average declines of only 0.87%, 1.37%, 0.30%, and 0.28% across the four metrics, and even improves performance in some instances. This improvement might be attributed to the reduction of attribute bias, leading to a more balanced data distribution. In comparison, UC-FedRec shows over a 4% average decrease across all recommendation performance metrics.

Table 5: Running Time Consumption of Unlearning Methods.

Time(s)	Aegis-Fed	UC-FedRec	Aegis-CS	U2U-R	D2D-R
ML-100K	392.04	941.74	10.91	9.80	5.43
ML-1M	4742.96	8223.53	37.66	109.68	54.45
ModCloth	571.22	981.43	165.18	167.37	87.35
Last.FM-1K	65085.79	163129.51	8.88	12.15	5.68

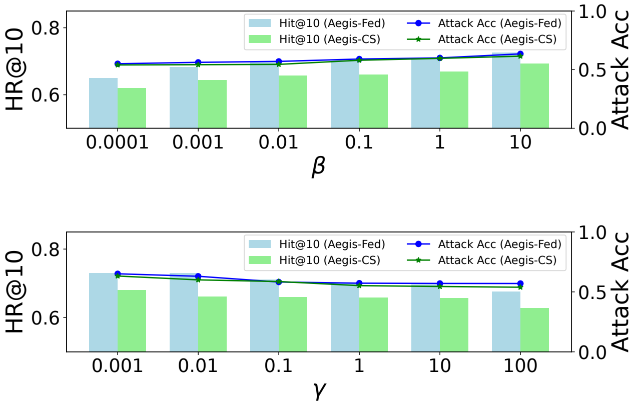

Figure 2: Impact of β, γ on Unlearning (Attack Accuracy) and Recommendation (HR@10) Performance on ML-100K.

4.2.3 ***Efficiency***. We recorded the runtime of the unlearning methods for the gender attribute on the MovieLens-100K, MovieLens-1M, and Last.FM-1M datasets, as well as the body shape attribute on the ModCloth dataset, to reflect the different efficiency. The results are presented in Table 5. We observe that in the data-dependent DD scenario, Aegis demonstrates superior efficiency compared to the UC-FedRec. This efficiency advantage arises from Aegis's ability to perform federated fine-tuning on the existing trained global model without introducing additional MLP-based attribute filtering modules. In the data-free DF scenario, Aegis-CS significantly accelerates the unlearning process by eliminating the need for interactive data, achieving comparable efficiency to U2U-R and D2D-R. Our methods provide new insights into rapid attribute unlearning.

4.2.4 ***Ablation Study***. To evaluate the robustness of our proposed multi-component loss function, we analyzed the effects of the trade-off parameters β and γ. By fixing one parameter at its default setting and varying the other, we measured recommendation performance using HR@10 and unlearning effectiveness using attack accuracy, as shown in Figure 2. The results reveal that increasing β leads to a slight improvement in recommendation performance and a minor decrease in unlearning effectiveness for both Aegis-Fed and Aegis-CS. In contrast, the sensitivity analysis of γ shows the opposite effect. Overall, both Aegis-Fed and Aegis-CS demonstrate robustness across different parameter settings, with minimal variation in recommendation and unlearning performance.

5 Related Work

Recommendation Unlearning. Machine unlearning has recently emerged as a method to quickly remove the impact of specific data on trained models [67]. Exact unlearning, exemplified by the SISA method [6], uses dataset partitioning and sub-model aggregation for fast retraining. Approximate unlearning manipulates model parameters, using techniques such as boundary learning [10] and knowledge distillation [37, 65]. Attribute unlearning is a type of machine unlearning. Early research on attribute unlearning [22] tackled privacy in facial recognition by adding network modules. In recommender systems, methods like [70] adjust gradient updates, while others [9, 41] apply SISA-based data grouping for unlearning. However, most recommender system unlearning has focused on the sample level [39, 46], with limited work on federated attribute unlearning. While [42] explored attribute unlearning with a custom loss function, it hasn't been extended to federated settings. There's a need for a more robust federated attribute unlearning framework that works with or without access to training data.

Privacy-preserving Recommender Systems. Federated learning (FL) [26, 50] has been applied in recommender systems to enhance user privacy by avoiding direct data sharing. However, federated recommender systems (FedRec) [51, 52] are still vulnerable to privacy risks, as central servers can infer private information from user parameters [59], such as user interactions, ratings, or attributes [8, 27, 64]. To mitigate these risks, privacy preservation mechanisms such as fake items, homomorphic encryption, and differential privacy (DP) have been employed. For example, [44, 45] use randomly sampled fake items to obscure user interactions, while homomorphic encryption [54, 69] ensures secure mathematical operations on encrypted data. Local differential privacy (LDP) [13, 15, 48] enables statistical computations and ensures the privacy of individual participants. However, encryption-based methods significantly increase communication costs, and DP primarily focuses on the transmission of model weights or updates, which is less effective in addressing inference attacks and privacy leakage in the learned federated recommender models [27, 59].

6 Conclusions

In this paper, we present Aegis, the first recommendation framework to address post-training attribute unlearning in a federated setting to our knowledge. Aegis enhances the privacy of FedRecs by selectively unlearning private user attributes while preserving recommendation quality. The framework balances private attribute unlearning and recommendation knowledge retention using a multi-component loss function based on information theory. Our approach minimizes the mutual information between user embeddings and sensitive attributes, combining this with regularization and recommendation losses to maintain performance. Aegis supports both interaction data-dependent and data-free unlearning, making it adaptable to different levels of data access.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (62302348, 62341410, 62472327), National Key Research and Development Program of China (2023YFE0205700), Science and Technology Development Fund, Macao S.A.R (FDCT) project (0078/2023/AMJ) and Key R&D Program of Hubei Province (2023BAB077). Chuang Hu and Jiawei Jiang are the corresponding authors of this paper.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Vienna, Austria, 308–318.

[2] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated Collaborative Filtering for Privacy-preserving Personalized Recommendation System. arXiv:1901.09888

[3] Caridad Arroyo Arevalo, Sayedeh Leila Noorbakhsh, Yun Dong, Yuan Hong, and Binghui Wang. 2024. Task-Agnostic Privacy-Preserving Representation Learning for Federated Learning against Attribute Inference Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. AAAI, Vancouver, Canada, 10909–10917.

[4] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-aware Recommendation with Private-attribute Protection using Adversarial Learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM, Houston, TX, USA, 34–42.

[5] Kailash Chowdary Bodduluri, Francis Palma, Ilir Jusufi, Arianit Kurti, and Henrik Löwenadler. 2024. Exploring the Landscape of Hybrid Recommendation Systems in E-Commerce: A Systematic Literature Review. *IEEE Access* 12 (2024), 28273–28296.

[6] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy*. IEEE, Virtual Event, San Francisco, CA, USA, 141–159.

[7] Òscar Celma Herrada et al. 2009. *Music Recommendation and Discovery in the Long Tail*. Universitat Pompeu Fabra, Barcelona,Spain.

[8] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. Secure Federated Matrix Factorization. *IEEE Intelligent Systems* 36, 5 (2020), 11–20.

[9] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation Unlearning. In *Proceedings of the ACM Web Conference 2022*. ACM, Virtual Event, Lyon, France, 2768–2777.

[10] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. 2023. Boundary Unlearning: Rapid Forgetting of Deep Networks via Shifting the Decision Boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Vancouver, Canada, 7766–7775.

[11] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. 2023. Zero-shot Machine Unlearning. *IEEE Transactions on Information Forensics and Security* 18 (2023), 2345–2354.

[12] Mukund Deshpande and George Karypis. 2004. Item-based Top-n Recommendation Algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177.

[13] Koustabh Dolui, Illapha Cuba Gyllensten, Dietwig Lowet, Sam Michiels, Hans Hallez, and Danny Hughes. 2019. Towards Privacy-preserving Mobile Applications with Federated Learning: The Case of Matrix Factorization. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, Seoul Republic of Korea, 624–625.

[14] Ludovic Dos Santos, Benjamin Piwowarski, and Patrick Gallinari. 2017. Gaussian Embeddings for Collaborative Filtering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Shinjuku, Tokyo, Japan, 1065–1068.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference*. Springer, New York, NY, USA, 265–284.

[16] Sanaz Eidizadehakhcheloo, Bizhan Alipour Pijani, Abdessamad Imine, and Michaël Rusinowitch. 2021. Divide-and-learn: A Random Indexing Approach to Attribute Inference Attacks in Online Social Networks. In *Data and Applications Security and Privacy XXXV: 35th Annual IFIP WG 11.3 Conference*. Springer, Calgary, Canada, 338–354.

[17] Ziwei Fan, Zhiwei Liu, Shen Wang, Lei Zheng, and Philip S Yu. 2021. Modeling Sequences as Distributions with Incertainty for Sequential Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM, Virtual Event, Queensland, Australia, 3019–3023.

[18] Tiantian Feng, Raghuveer Peri, and Shrikanth Narayanan. 2022. User-level Differential Privacy against Attribute Inference Attack of Speech Emotion Recognition in Federated Learning. arXiv:2204.02500

[19] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2024. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. AAAI, Vancouver, Canada, 12043–12051.

[20] Christian Ganhör, David Penz, Navid Rekabsaz, Oleg Lesota, and Markus Schedl. 2022. 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*. ACM, Madrid, Spain, 2142–2147.

[21] Chen Gao, Nian Li, Tzu-Heng Lin, Dongsheng Lin, Jun Zhang, Yong Li, and Depeng Jin. 2020. Social Recommendation with Characterized Regularization. *IEEE Transactions on Knowledge and Data Engineering* 34, 6 (2020), 2921–2933.

[22] Tao Guo, Song Guo, Jiewei Zhang, Wenchao Xu, and Junxiao Wang. 2022. Efficient Attribute Unlearning: Towards Selective Removal of Input Attributes from Feature Representations. arXiv:2202.13295

[23] F Maxwell Harper and Joseph A Konstan. 2015. The Movielens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2015), 1–19.

[24] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to Represent Knowledge Graphs with Gaussian Embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, Melbourne, Australia, 623–632.

[25] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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*. ACM, Xi'an, China, 639–648.

[26] Chuang Hu, Nanxi Wu, Siping Shi, Xuan Liu, Wenhan Wu, Bing Luo, Kanye Ye Wang, Jiawei Jiang, and Dazhao Cheng. 2024. PriFairFed: A Local Differentially Private Federated Learning Algorithm for Client-Level Fairness. *IEEE Transactions on Mobile Computing* Early Access, N/A (2024), 1–12.

[27] Qi Hu and Yangqiu Song. 2024. User Consented Federated Recommender System against Personalized Attribute Inference Attack. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. ACM, Merida, Mexico, 276–285.

[28] Erin Illman and Paul Temple. 2019. California Consumer Privacy Act. *The Business Lawyer* 75, 1 (2019), 1637–1646.

[29] Dietmar Jannach, Zeynep Karakaya, and Fatih Gedikli. 2012. Accuracy Improvements for Multi-criteria Recommender Systems. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM, Valencia, Spain, 674–689.

[30] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[31] Bargav Jayaraman and David Evans. 2022. Are Attribute Inference Attacks just Imputation?. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Los Angeles, CA, USA, 1569–1582.

[32] Jinyuan Jia and Neil Zhenqiang Gong. 2018. AttriGuard: A Practical Defense against Attribute Inference Attacks via Adversarial Machine Learning. In *27th USENIX Security Symposium*. USENIX, Baltimore, MD, USA, 513–529.

[33] Jyun-Yu Jiang, Cheng-Te Li, Yian Chen, and Wei Wang. 2018. Identifying Users behind Shared Accounts in Online Streaming Services. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Ann Arbor, MI, USA, 65–74.

[34] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior Recommendation with Graph Convolutional Networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Virtual Event, China, 659–668.

[35] Pantea Keikhosrokiani and Goh Man Fye. 2024. A Hybrid Recommender System for Health Supplement E-commerce based on Customer Data Implicit Ratings. *Multimedia Tools and Applications* 83, 15 (2024), 45315–45344.

[36] Nikhil Ketkar. 2017. *Stochastic Gradient Descent*. Apress, Berkeley, CA, 113–132.

[37] Hyunjune Kim, Sangyong Lee, and Simon S Woo. 2024. Layer Attack Unlearning: Fast and Accurate Machine Unlearning via Layer Level Attack and Knowledge Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. AAAI, Vancouver, Canada, 21241–21248.

[38] Solomon Kullback and Richard A Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86.

[39] Jens Leysen. 2023. Exploring Unlearning Methods to Ensure the Privacy, Security, and Usability of Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, Singapore, Singapore, 1300–1304.

[40] Jie Li, Tianqing Zhu, Wei Ren, and Kim-Kwang Raymond. 2023. Improve Individual Fairness in Federated Learning via Adversarial Training. *Computers and Security* 132 (2023), 103336.

[41] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Junlin Liu, and Jun Wang. 2024. Making Recommender Systems Forget: Learning and Unlearning for Erasable Recommendation. *Knowledge-Based Systems* 283 (2024), 111124.

[42] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Zhongxuan Han, Dan Meng, and Jun Wang. 2023. Making Users Indistinguishable: Attribute-wise Unlearning in Recommender Systems. In *Proceedings of the 31st ACM International Conference on Multimedia*. ACM, Ottawa, Canada, 984–994.

[43] Zhiwei Li, Guodong Long, and Tianyi Zhou. 2025. Federated Recommendation with Additive Personalization. In *The Twelfth International Conference on Learning Representations*. ICLR, Singapore.

[44] Feng Liang, Weike Pan, and Zhong Ming. 2021. Fedrec++: Lossless Federated Recommendation with Explicit Feedback. In *Proceedings of the AAAI conference on Artificial Intelligence*, Vol. 35. AAAI, Virtual Event, 4224–4231.

[45] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. 2020. Fedrec: Federated Recommendation with Explicit Feedback. *IEEE Intelligent Systems* 36, 5 (2020), 21–30.

[46] Wenyan Liu, Juncheng Wan, Xiaoling Wang, Weinan Zhang, Dell Zhang, and Hang Li. 2022. Forgetting Fast in Recommender Systems. arXiv:2208.06875

[47] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2022. ML-Doctor: Holistic Risk Assessment of Inference Attacks against Machine Learning Models. In *31st USENIX Security Symposium*. USENIX, Boston, MA, USA, 4525–4542.

[48] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S Yu. 2022. Federated Social Recommendation with Graph Neural Network. *ACM Transactions on Intelligent Systems and Technology* 13, 4 (2022), 1–24.

[49] Rui Lu, Siping Shi, Dan Wang, Chuang Hu, and Bihai Zhang. 2022. Preva: Protecting Inference Privacy through Policy-based Video-frame Transformation. In *2022 IEEE/ACM 7th Symposium on Edge Computing*. IEEE, Seattle, WA, USA, 175–188.

[50] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, Fort Lauderdale, FL, USA, 1273–1282.

[51] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. 2021. Stronger Privacy for Federated Collaborative Filtering with Implicit Feedback. In *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, Amsterdam Netherlands, 342–350.

[52] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. Fedfast: Going Beyond Average for Faster Training of Federated Recommender Systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Virtual Event, CA, USA, 1234–1242.

[53] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. 2021. Membership Inference Attack on Graph Neural Networks. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*. IEEE, Virtual Event, 11–20.

[54] Pascal Paillier. 1999. Public-key Cryptosystems based on Composite Degree Residuosity Classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Prague, Czech Republic, 223–238.

[55] Vasileios Perifanis and Pavlos S Efraimidis. 2022. Federated Neural Collaborative Filtering. *Knowledge-Based Systems* 242 (2022), 108441.

[56] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, Aachen, Germany, 452–461.

[57] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. arXiv:1806.01246

[58] Ralf Steuer, Jürgen Kurths, Carsten O Daub, Janko Weise, and Joachim Selbig. 2002. The Mutual Information: Detecting and Evaluating Dependencies between Variables. *Bioinformatics* 18, suppl_2 (2002), S231–S240.

[59] Zehua Sun, Yonghui Xu, Yong Liu, Wei He, Lanju Kong, Fangzhao Wu, Yali Jiang, and Lizhen Cui. 2024. A Survey on Federated Recommendation Systems. *IEEE Transactions on Neural Networks and Learning Systems* 36, 1 (2024), 6–20.

[60] Paul Voigt and Axel Von dem Bussche. 2017. The EU General Data Protection Regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.

[61] Mengting Wan, Jianmo Ni, Rishabh Misra, and Julian McAuley. 2020. Addressing Marketing Bias in Product Recommendations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM, Houston, TX, USA, 618–626.

[62] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Sydney NSW, Australia, 1235–1244.

[63] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond Inferring Class Representatives: User-level Privacy Leakage from Federated Learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, Paris, France, 2512–2520.

[64] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. Fedgnn: Federated Graph Neural Network for Privacy-preserving Recommendation. arXiv:2102.04925

[65] Chen Wu, Sencun Zhu, and Prasenjit Mitra. 2022. Federated Unlearning with Knowledge Distillation. arXiv:2201.09441

[66] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: a Survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[67] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. 2024. Machine Unlearning: Solutions and Challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence* 8, 3 (2024), 2150–2168.

[68] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Vol. 17. Morgan Kaufmann, Melbourne, Australia, 3203–3209.

[69] Enyue Yang, Yunfeng Huang, Feng Liang, Weike Pan, and Zhong Ming. 2021. FCMF: Federated Collective Matrix Factorization for Heterogeneous Collaborative Filtering. *Knowledge-Based Systems* 220 (2021), 106946.

[70] Wei Yuan, Hongzhi Yin, Fangzhao Wu, Shijie Zhang, Tieke He, and Hao Wang. 2023. Federated Unlearning for On-Device Recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. ACM, Singapore, 393–401.

[71] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Lizhen Cui, and Xiangliang Zhang. 2021. Graph Embedding for Recommendation against Attribute Inference Attacks. In *Proceedings of the ACM Web Conference 2021*. ACM, Ljubljana Slovenia, 3002–3014.

[72] Shijie Zhang, Wei Yuan, and Hongzhi Yin. 2023. Comprehensive Privacy Analysis on Federated Recommender System against Attribute Inference Attacks. *IEEE Transactions on Knowledge and Data Engineering* 36, 3 (2023), 987–999.

A Algorithm Workflow

As shown in Algorithm 1, Aegis implements attribute unlearning in federated recommendation systems with two main scenarios: Data Dependent (DD) and Data Free (DF).

- **Data Dependent (DD)**: Clients have access to their data and compute both privacy loss and recommendation retention loss locally. After updating the model, clients send gradients to the server, which aggregates and updates the global model. Training stops early if the updates are below a threshold.
- **Data Free (DF)**: Clients making unlearning requests do not have access to their data. They still update the model using available gradients and regularization loss. The server aggregates the gradients and updates the global model. Early stopping also occurs if the updates are small.

B Applicability in Modern Federated Recommender Systems

Aegis is designed to address privacy concerns in Federated Recommendation Systems (FRS) by defending against Attribute Inference Attacks (AIAs) targeting user embeddings. The key feature of Aegis is its ability to work seamlessly with any FRS that utilizes user embeddings, without requiring alterations to the core recommendation structure. This design choice ensures compatibility with widely used FRS frameworks, where user embeddings remain a critical component in models such as FedRAP [43] and Fed-LightGCN [25].

Table 6: Results with FedRAP on ML-100K

Metrics		XGBoost AIA			MLP AIA			NDCG@10	HR@10
Attribute		Gender	Age	Occupation	Gender	Age	Occupation		
Original		0.8050	0.3369	0.2440	0.7175	0.3660	0.2149	0.8671	0.9423
DD	Aegis-Fed	**0.5901**	**0.2082**	**0.1188**	0.6194	**0.1406**	**0.0543**	**0.8495**	**0.9485**
	UC-FedRec	0.6697	0.3050	0.2042	**0.6008**	0.2135	0.1074	0.8179	0.9018
DF	Aegis-CS	**0.5517**	**0.3209**	**0.0769**	**0.6034**	**0.1976**	**0.0782**	**0.7934**	**0.8919**
	U2U-R	0.9999	0.8952	0.9456	0.6631	0.2891	0.1127	0.7886	0.8871
	D2D-R	0.6061	0.3886	0.0862	0.6485	0.2493	0.0782	0.7620	0.8766

Table 7: Results with Fed-LightGCN on ML-100K

Metrics		XGBoost AIA			MLP AIA			NDCG@10	HR@10
Attribute		Gender	Age	Occupation	Gender	Age	Occupation		
Original		0.7029	0.3422	0.1923	0.7520	0.3382	0.2255	0.7847	0.7657
DD	Aegis-Fed	**0.5557**	**0.1499**	**0.0849**	0.5530	**0.1645**	**0.0836**	**0.7944**	**0.7768**
	UC-FedRec	0.7015	0.2103	0.1688	**0.5477**	0.1790	0.0838	0.6144	0.4962
DF	Aegis-CS	**0.5066**	**0.1698**	**0.1154**	**0.5782**	**0.2069**	**0.0716**	**0.7203**	**0.7018**
	U2U-R	0.9151	0.6989	0.8422	0.5875	0.2149	0.0796	0.6255	0.6033
	D2D-R	0.6273	0.2228	0.1658	0.6153	0.2480	0.0822	0.6150	0.6036

Algorithm 1 Aegis Attribute Unlearning Process

```
Input: Trained model parameters $\theta_0 = \{\varphi_0, p_0\}, \psi_0$, private attributes $A_u$, number of local iterations $L_l$, server aggregation rounds $T$, Hyperparameters $\beta, \gamma$, threshold $\epsilon$, learning rate $\eta$.
Output: Updated user embeddings $f_\theta(u)$ and model parameters $\theta$
The client submits an unlearning request for the attribute set $A_u$;
Initialize global parameter with trained model $\theta_0 = \{\varphi_0, p_0\}, \psi_0$;
Server: Distribute the attribute unlearning request of $A_u$ to clients;
if The clients can participate in unlearning with available data then
    for each global round $t = 0, 1 \ldots, T-1$ do
        sampling a fraction of clients $C_t \subseteq U$
        for each client $c_i \in C_t$ in parallel do
            for each local iteration $l = 1, 2, \ldots, L_l$ do
                Compute the private attribute unlearning loss $\mathcal{L}_t^{AU} = \sum_{j \in A_u} \mathcal{L}_{j,t}^{AU}$ using Eq. (15)
                Compute the recommendation knowledge retention loss $\mathcal{L}_t^{Rec}$ and regularization loss $\mathcal{L}_t^{Reg}$ using Eq. (16) and (17)
                Update model parameters $\theta_{t,i} = \{\varphi_{t,i}, p_{t,i}\}$ by minimizing $\mathcal{L}^{All}$ in Eq. (18): $\theta_{t,i} = \theta_{t,i} - \eta \nabla_\theta (\mathcal{L}_t^{Rec} + \beta \mathcal{L}_t^{Reg} + \gamma \mathcal{L}_t^{AU})$
            end for
            Upload updated unlearning gradients to the server
        end for
        Server: Aggregate the gradients and update the global model, $\theta_{t+1} = \frac{1}{\sum_{c_j \in C_t} |D_{c_j}|} \sum_{c_i \in C_t} |D_{c_i}| \theta_{t,i}$
        if The update in embeddings: $\|f_{\theta_{t+1}}(u) - f_{\theta_t}(u)\| < \epsilon$ then
            Break: End training early as updates are below the threshold
        end if
        Server: Distribute $\theta_{t+1}$ to all clients, $\theta_{t+1,i} = \theta_{t+1}, \forall i \in U$
    end for
else
    for each round $t = 0, 1, \ldots, T-1$ on Client making request do
        Compute the private attribute unlearning loss $\mathcal{L}_t^{AU} = \sum_{j \in A_u} \mathcal{L}_{j,t}^{AU}$ using Eq. (15)
        Compute the regularization loss $\mathcal{L}_t^{Reg}$ using Eq. (17)
        Update model parameters $\theta = \{\varphi, p\}$ by minimizing $\mathcal{L}^{All'}$ in Eq. (19): $\theta_{t+1} = \theta_t - \eta \nabla_\theta (\beta \mathcal{L}_t^{Reg} + \gamma \mathcal{L}_t^{AU})$
        if The update in embeddings: $\|f_{\theta_{t+1}}(u) - f_{\theta_t}(u)\| < \epsilon$ then
            Break: End training early as updates are below the threshold
        end if
    end for
end if
```

To validate the generality of the Aegis framework, we conducted additional experiments using Aegis-Fed and Aegis-CS applied to these two different Federated Recommendation models. FedRAP primarily utilizes a personalized training method that divides item embeddings into global and local components, while Fed-LightGCN substitutes the backbone model with LightGCN. The results demonstrate that Aegis offers significant privacy protection without compromising recommendation performance. Specifically, we observe lower AIA attack accuracy and improved recommendation quality, as measured by NDCG@10 and HR@10, across most scenarios. For instance, in the case of FedRAP, Aegis outperforms UC-FedRec by an average of 6.08%, with a 2.33% improvement in recommendation quality metrics. In the data-free scenario, Aegis-CS outperforms U2U-R by over 30% in privacy protection, with a 0.4% improvement in recommendation quality. It also surpasses D2D-R by 3.8% in privacy performance, with a 2.3% improvement in recommendation quality metrics. Detailed results are shown in Tables 6 and 7.

C Complexity Analysis

This section provides the algorithmic complexity of Aegis-Fed in one communication round and Aegis-CS in one iteration for a client.

C.1 Computing Privacy Attribute Loss

To compute the privacy attribute loss $\mathcal{L}_t^{AU}$ on a client t, we calculate the distribution of both global and individual privacy attribute classes for a given attribute $au_j \in A_u$. The primary computational complexity arises from calculating the covariance matrix, which has a complexity of $O(|U_t| \cdot d^2)$, where d is the embedding dimension. Next, we compute the KL divergence between these distributions. Since the complexity of computing the KL divergence between two distributions is $O(d^3)$, the complexity for an attribute a_j with $|U_{a_j}|$ privacy labels is $O(|U_{a_j}| \cdot d^3)$. Therefore, the total complexity for computing $\mathcal{L}_t^{AU}$ for all attributes is $O(\sum_{au_j \in A_u} |U_{a_j}| \cdot d^3 + |A_u| \cdot |U_t| \cdot d^2)$, where $|A_u|$ is the total number of attributes. Since d is typically small, the computation of $\mathcal{L}_t^{AU}$ remains manageable.

C.2 Computing Recommendation Knowledge Retention Loss

In Aegis-Fed, we additionally use the original training phase recommendation loss function $\mathcal{L}^{Rec}$. Assuming interactions with all items, the complexity per iteration is $O(|U_t| \cdot |V| \cdot d)$. For the regularization loss $\mathcal{L}^{Reg}$, the complexity is $O(|U_t| \cdot d)$. The computation of the recommendation loss function is independent of Aegis and depends on the FedRec training method used, while the regularization overhead is small and negligible in most cases.

C.3 Communication Overhead

In Aegis-Fed, the communication overhead between the server and clients is linear with respect to the number of items V and the embedding dimension d. This communication overhead is consistent with that of the FL training phase, and Aegis does not introduce any additional communication overhead.

Since user embeddings must always be updated, the complexity of any privacy-preserving algorithm cannot be lower than $O(|U_t| \cdot d)$. In this case, the complexity of Aegis is mainly determined by d in the privacy attribute loss, where d is typically much smaller than $|U_t|$ and is pre-determined. This ensures that the overall loss has low complexity and can be computed efficiently. In the Aegis-CS scenario, $|V|$ does not affect the scalability of the method. In the Aegis-Fed scenario, most computational tasks are concentrated in the training loss computation.