

# Defending against Attribute Inference Attacks in Post-Training of Recommendation Systems via Unlearning

Wenhan Wu<sup>†</sup>, Yili Gong<sup>†</sup>, Jiawei Jiang<sup>†</sup>, Chuang Hu<sup>‡</sup>, Xiaobo Zhou<sup>‡</sup> and Dazhao Cheng<sup>†</sup>

<sup>†</sup>School of Computer Science, Wuhan University, Wuhan, China

{wenhanwu, yiligong, jiawei.jiang, dcheng}@whu.edu.cn

<sup>‡</sup>State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau SAR

{chuanghu, waynexzhou}@um.edu.mo

**Abstract**—Attribute Inference Attacks (AIAs) pose a significant threat to recommendation systems (RS) by enabling adversaries to use threat models to infer sensitive user attributes like gender or race from user embeddings, resulting in privacy breaches such as unauthorized profiling and discriminatory policies against specific groups. Existing attribute protection methods are primarily applied during training, suffering from significant limitations, such as architectural inflexibility, dependence on interaction data, and potential catastrophic degradation in recommendation performance. To overcome these challenges, we propose AttrCloak, an efficient and effective post-training attribute unlearning (AU) framework that removes sensitive information from user embeddings without altering RS training architectures. AttrCloak employs dual-objective optimization with parameter self-sharing to minimize mutual information between user embeddings and sensitive attributes while preserving recommendation quality. Furthermore, it accommodates data-free scenarios by leveraging regularization loss when interaction data is unavailable. Comprehensive evaluations on four real-world datasets demonstrate AttrCloak’s good performance in privacy protection and recommendation performance.

**Index Terms**—Recommendation System, Attribute Unlearning, Multi-objective Optimization, Privacy-preserving.

## I. INTRODUCTION

Recommendation systems (RSs) utilize user-item interaction data to train models that provide personalized service recommendations. Existing studies [1], [2] have shown that these systems are vulnerable to *Attribute Inference Attacks* (AIAs) [3], [4], [5], where attackers leverage threat models to infer sensitive user attributes like age or gender from embeddings generated through collaborative filtering (CF). These inferred attributes can expose private user information [6], leading to significant privacy breaches, including unauthorized profiling [1] and discriminatory practices [7] based on characteristics such as gender or race, which underscore the urgent need for effective privacy-preserving mechanisms in RS.

Depending on whether attribute-wise privacy protection measures are applied during training or after model training is completed, the protection methods can be divided into two categories: *in-training* and *post-training*. Since post-training methods often rely on techniques such as adversarial training [8], [9], these methods require changes to the recom-

mendation model’s architecture and involve more complex computations. Besides, due to their predefined requirements for sensitive attributes, they are also not suitable for more flexible and dynamic privacy needs. Therefore, we aim to propose a more flexible post-training approach. Inspired by “*Recommendation Unlearning (RU)*” [10], [11], we hope to effectively “unlearning” the attribute contributions or characteristics associated with user embeddings in the RS model.

Most existing RU methods use input data as the unlearning target [12], which aims to erase the influence of specific users, items, or instances. However, they cannot effectively decouple sensitive attributes from user embeddings to defend against AIAs, and even these attributes are not explicitly used as input during training. In this context, *attribute unlearning (AU)* is proposed to safeguard privacy by eliminating sensitive attributes that are implicitly embedded in user embeddings within RS models in post-training. We define the gold standard for AU as an *attribute-indistinguishability mechanism*. Specifically, the desired outcome is that user embeddings become indistinguishable with respect to their sensitive attributes, while the recommendation performance remains on par with that of the original model.

As a post-training framework, AU faces the challenge posed by data-free environments. AU requests are often unpredictable, and due to privacy regulations [13] or data deletion, training data or historical updates may become inaccessible [14], [15]. Accessing additional data incurs costs, and in many cases, only the trained model is retained without the original data. For instance, interaction data such as personal shopping records may only be temporarily accessible with user consent and then deleted after the withdrawal of authorization. Even after withdrawal, users still require protection for their attribute privacy. Therefore, a “*data-free*” AU approach, which does not rely on interaction data, can more effectively address these challenges than data-dependent unlearning methods.

Additionally, due to overfitting to unlearning targets or conflicts in multi-task optimization, unlearning methods often result in significant model accuracy loss, termed “*catastrophic unlearning*” [16], [17], [18]. Fixed thresholds or weights of unlearning loss in current methods often struggle to balance

unlearning effectiveness and recommendation accuracy [16]. Other methods for preserving attribute privacy also face the challenge of balancing privacy protection and model utility [19], [20]. For example, [2] proposes making privacy attributes indistinguishable in distance or distribution, but their single-objective approach often yields suboptimal results. Differential privacy [21], [22] protects attributes by adding noise, which typically degrades recommendation performance [2].

To address the evolving privacy and data requirements in RS, we propose a flexible and efficient post-training AU framework, termed AttrCloak<sup>1</sup>. AttrCloak fine-tunes RS models post training without altering the model architecture to defend against AIAs. We present a dual-objective optimization method to achieve two primary goals: i) making private attributes indistinguishable, and ii) preserving recommendation performance. The first objective is accomplished by minimizing the variational upper bound of mutual information between user embeddings and sensitive attributes. The second is fulfilled by minimizing either the original training loss or a regularization loss, the latter being crucial when interaction data is unavailable, which constrains updates to user embeddings to preserve previously learned information. Finally, the dual-objective optimization strikes a balance between the two goals, thereby preventing catastrophic unlearning. Our optimization method is within the framework of *multi-task learning (MTL)* [23], [24], which is a learning paradigm that aims to improve the performance of multiple related tasks by leveraging shared representations and parameters across tasks. In MTL, different tasks share the hidden layers of the model while maintaining independent output layers for each task. AttrCloak shares all parameters responsible for generating user embeddings, optimizing both privacy and performance objectives through gradient descent. Due to the lack of task-specific layers common in MTL, we refer to it as “*parameter self-sharing*” [16], [25], and we utilize a simple gradient clipping technique to identify the steepest descent direction. To sum up, our main contributions are as follows:

- We present the AttrCloak framework for post-training attribute unlearning, designed for scenarios with or without interaction data access. It better handles flexible privacy requirements and adapts to real-world data environments.
- We reformulate attribute unlearning as a dual-objective optimization problem with parameter self-sharing, focusing on both sensitive attribute indistinguishability and recommendation performance retention.
- We design a novel effective multi-component loss based on information theory achieving unlearning by minimizing mutual information between user embeddings and attributes.
- We implemented AttrCloak and evaluated its ability to balance privacy and recommendation utility on four real-world datasets, outperforming existing methods with average improvements of 19.64% and 9.44%, respectively.

<sup>1</sup>symbolizing the concealment of sensitive Attributes, akin to the invisibility Cloak in Harry Potter.

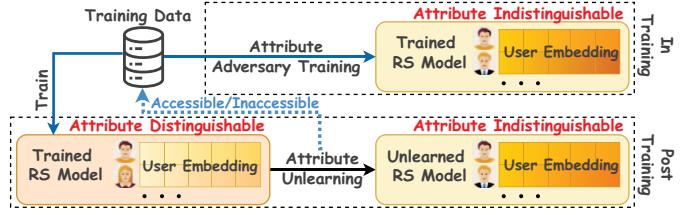


Fig. 1: Comparison of Post-Training and In-Training Attribute-wise Privacy Protection Methods.

## II. RELATED WORK

### A. Privacy-preserving Recommendation Systems.

Private-attribute inference attacks in recommendation systems aim to infer a user’s private attributes from their user embeddings. As shown in Fig. 1, current attribute-wise privacy protection architectures can be divided into two frameworks: *in-training* and *post-training*. The key distinction between them lies in whether the privacy protection process occurs during training or after the model has been trained.

Most existing works fall under the *in-training* category, which rely on adversarial training strategies [6], [8] or data modification-based methods [5], [26], [27] to create privacy-preserving RS models. On the one hand, adversarial learning methods, like [6], integrate recommendation with attribute protection by introducing an inference attacker model and adversarial training. Other adversarial approaches, such as [8], use a privacy discriminator to remove sensitive user information from node representations. These methods require additional network structures, complicating the original recommendation model and reducing flexibility. On the other hand, data modification-based privacy-preserving methods, such as [26], add a predefined number of dummy items to each user’s profile, where these dummy items are negatively correlated with the user’s actual attributes. Afterward, anonymized user-item rating data is released. Attrguard [5] samples attribute values probabilistically and adds minimal noise to user-item data to mislead attackers. However, modifying the data inevitably affects recommendation quality. Besides, data modification methods cannot operate without the original data, which contradicts the changeable privacy requirements of RS.

With the *in-training* methods tending to be resource-intensive and complex, requiring prior knowledge of protected attributes [1], [28], it is less suitable for environments where privacy requirements are flexible or change over time. In contrast, post-training protection methods like AttrCloak in this paper offer more flexibility by applying privacy safeguards to pre-trained models without the need for full retraining, architectural changes, and predetermining defending settings, which are consistent with “Recommendation Unlearning.”

### B. Recommendation Unlearning.

Recently, machine unlearning (MU) has emerged as a method for quickly eliminating the influence of specific data on already trained models [29]. Exact unlearning ensures that

TABLE I: Model Utility and Attribute Inference Attack Results via Data-Input Level Recommendation Unlearning.

Stage	Utility		Attribute Privacy		
	NDCG@10↑	HR@10↑	Gender	Age	Location
Original	0.632	0.610	0.751	0.604	0.588
Unlearned	0.603	0.589	0.728	0.632	0.609
Random Attacker			0.500	0.143	0.167

the model after unlearning is as effective as if it had been retrained from scratch. A representative approach is the SISA method [30], which leverages dataset partitioning and sub-model aggregation to achieve fast unlearning. In contrast, Approximate unlearning is achieved by performing parameter manipulations on trained models, such as boundary learning [31] and knowledge distillation [32], [33]. Recommendation unlearning (RU) is a form of MU in RS. [34] utilizes the magnitude of historical gradient updates and sampling techniques to modify update directions for approximate RU, while [12], [35] follow SISA, grouping similar data and using attention-based aggregation for RU. However, most existing work on RU focuses on the data-input level [10], [11], e.g., users, items, and instances. Although they can efficiently remove user/item data influences, they cannot protect the RS from AIAs. Specifically, we applied a data-input level RU method [12] to randomly forget 10% of the samples in the Last.FM-1K dataset [36]. The results before and after unlearning are shown in Table. I. Notably, the attack accuracy of AIAs on the three private attributes showed no significant reduction after unlearning, remaining significantly higher than that of random attackers, reaching up to 48.9% in some cases. Meanwhile, the recommendation performance metrics, NDCG@10 [37] and HR@10 [38], on the remaining users showed a slight decline. These findings indicate that data-input level RU methods are ineffective in mitigating AIAs.

Therefore, it is essential to explore attribute unlearning (AU). The earliest research on AU [9] focused on removing the impact of facial privacy information, rather than applying it directly within RS. [2] first explored AU in RSs and proposed a multi-component system with a custom loss function. However, they treat AU as a single-objective optimization problem, often relying on fixed values or thresholds to constrain the extent of unlearning, which is difficult to define precisely. Therefore, this paper formulates AU as a dual-objective optimization problem to strike a Pareto-optimal balance between the unlearning effect and recommendation performance.

### III. PRELIMINARIES

#### A. Attribute-wise Privacy in Recommendation Systems.

Modern RSs learn user embeddings – low-dimensional vectors capturing user preferences through interaction patterns. However, these embeddings may inadvertently encode sensitive attributes (e.g., gender, age), enabling adversaries to infer private information via Attribute Inference Attacks (AIAs) [5], [39]. For instance, an attacker could train a classifier on leaked user embeddings to predict gender with high accuracy,

TABLE II: Recommendation System Utility and Attribute Inference Attack Results on Different Datasets.

Dataset	Utility		Attribute Privacy		
	NDCG@10↑	HR@10↑	Gender	Age	Occupation
ML-100K	0.772	0.748	0.763	0.450	0.228
ML-1M	0.610	0.585	0.796	0.416	0.200
Random Attacker			0.500	0.143	0.048

even if the recommendation model never explicitly processes such attributes. To reveal the risk of AIAs, we conducted AIAs on CF models trained using the MovieLens-100K (ML-100K) and MovieLens-1M (ML-1M) datasets [40]. As shown in Table II, although the recommendation performance is good, the accuracy of AIAs significantly outperformed random baseline predictions up to 30.7%. These findings highlight that, although the RS model does not directly observe sensitive attributes and the attacker's access to the data is limited, the feasibility of inferring sensitive user information still exists. Due to the unauthorized profiling and potential discriminatory policies caused by unintended leakage of sensitive attribute information, there is a requirement for attribute-wise privacy-preserving mechanisms in RSs. Our goal is to remove these sensitive associations post-training, a process we refer to as “post-training attribute unlearning”. We mathematically formulated the general workflow of RS and AIAs as below:

**Recommendation Systems.** Let  $U$  represent the set of users, with the total number of users denoted as  $|U|$ , and  $V$  the set of items with  $|V|$  the total number of items. Each user  $u_i \in U$  interacts with the dataset  $D$ , defined as:

$$D = \{(u_i, v_j, r_{ij}) | v_j \in V\}, \quad (1)$$

where  $r_{ij} = 1$  indicates user  $u_i$  has interacted with item  $v_j$ , and  $r_{ij} = 0$  denotes no interaction, in which case  $v_j$  is considered a negative sample. The RS predicts scores  $\hat{r}_{ij}$  for unobserved items  $v_j$ , generating a recommendation list  $\hat{V}_i$ :

$$\hat{V}_i = \text{Top-K}(\{\hat{r}_{ij} | v_j \in V \setminus D_i\}). \quad (2)$$

Here,  $\hat{r}_{ij} = s_\psi(\mathbf{em}_i, \mathbf{em}_j)$  is a scoring function, such as a dot product or a multi-layer perception (MLP). The user and item embeddings  $\mathbf{em}_i$  and  $\mathbf{em}_j$  are derived from an embedding layer  $f_\varphi = [f_{\varphi_u}, f_{\varphi_v}]$  and refined through propagation  $f_p$  [41] to capture collaborative signals, denoted as  $\mathbf{em}_i = f_p(f_{\varphi_u}(u_i)) \in \mathbb{R}^d$  and  $\mathbf{em}_j = f_p(f_{\varphi_v}(v_j)) \in \mathbb{R}^d$ , where  $d$  is the embedding dimension.  $f_{\varphi_u}$  and  $f_{\varphi_v}$  generate user and item embeddings, respectively. The model is iteratively trained on a global dataset  $D$  from all users until convergence, with the entire model parameters  $\theta$  updated as:

$$\theta^{t+1} = \theta^t - \eta \nabla_\theta \mathcal{L}(\theta, D), \quad (3)$$

where  $\mathcal{L}$  represents the loss function (e.g., BPR loss [42]), and  $\eta$  denotes the learning rate. RSs trained using this process could achieve recommendation goals.

**AIA Threat Model.** AIAs exploit user embeddings  $\mathbf{em}_i$  to infer sensitive attributes  $z_i$ , such as gender, age, or race [6], [43]. The threat models in AIAs adopt a grey-box approach,

meaning the attackers do not have access to all model parameters but can access some user embedding vectors  $\mathbf{em}_i$  and corresponding attribute information  $z_i$ . The attack is framed as a classification task where the attackers train a model  $g$  to predict private attributes  $z_i$  for the user embedding  $\mathbf{em}_i$ . The threat model is trained on a shadow dataset  $D_{\text{shadow}}$ , which can be generated by sampling from the original user data within the same distribution [2], [44]. 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  $\mathcal{L}_c$  as below:

$$\min_g \mathbb{E}_{(\mathbf{Em}_{\text{shadow}}, Z_{\text{shadow}})} [\mathcal{L}_c(Z_{\text{shadow}}, g(\mathbf{Em}_{\text{shadow}}))], \quad (4)$$

where  $\mathbf{Em}_{\text{shadow}}$  and  $Z_{\text{shadow}}$  are the user embeddings and attributes in  $D_{\text{shadow}}$ , respectively. During inference, the attacker uses  $g$  to predict sensitive attributes  $\hat{Z}_i = g(\mathbf{Em}_i)$ .

### B. Parameter Sharing and Gradient Descent in MTL

AttrCloak's objective is to optimize user embeddings through dual-objective optimization that balances performance and attribute privacy. We model this as a multi-task learning (MTL) problem [23], [24], utilizing the hard parameter sharing method. Hard parameter sharing [45], [46], as opposed to soft parameter sharing [47], [48], [49], is one of the two main approaches in the MTL domain. In this method, different tasks share the hidden layers of the model while maintaining independent output layers for each task, as illustrated in Fig. 2. We define the MTL problem over an input space  $\mathcal{X}$  and a set of task spaces  $\{\mathcal{Y}_t\}_{t \in [T]}$ , where the dataset consists of  $N$  data points  $\{x_i, y_i^1, \dots, y_i^T\}_{i \in [N]}$ , with  $x_i$  representing the input and  $y_i^t$  representing the label for the  $t$ -th task used to compute the loss. In solving the  $t$ -th task, the hard parameter sharing method can be expressed as optimizing the task function  $f_t(x; \theta_{\text{sh}}, \theta_t) : \mathcal{X} \rightarrow \mathcal{Y}_t$ , where  $\theta_{\text{sh}}$  are the shared parameters used across different tasks, and  $\theta_t$  represents the task-specific output layers. The hard parameter sharing solution in MTL can be realized by minimizing the empirical weighted loss:

$$\min_{\theta_{\text{sh}}, \theta_1, \dots, \theta_T} \sum_{t=1}^T \alpha_t \mathcal{L}_t(\theta_{\text{sh}}, \theta_t), \quad (5)$$

where  $\alpha_t$  denotes the static or dynamic weight assigned to the empirical loss  $\mathcal{L}_t(\theta_{\text{sh}}, \theta_t)$  of the  $t$ -th task. Since our framework does not have task-specific layers like MTL when updating user representations, we refer to our method as parameter self-sharing [16]. The shared layers are utilized to generate user representations, while the two objective losses of AU serve as task-specific objectives. To solve this optimization problem, we can transform the MTL problem into a multi-objective optimization problem [24], and handle it using gradient descent to reach a local optimum. This approach is known as the multiple gradient descent algorithm (MGDA) [24], [50]. MGDA is based on the Karush-Kuhn-Tucker (KKT) conditions [51],

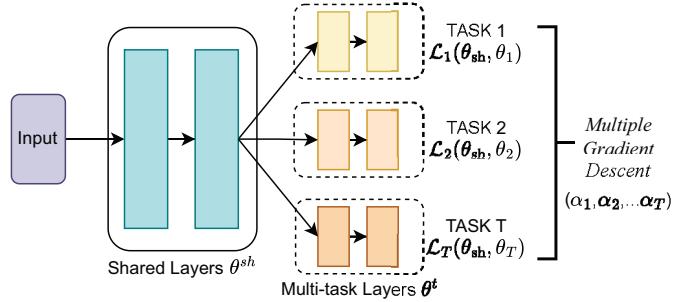


Fig. 2: Hard Parameter Sharing for Multi-task Learning.

which are necessary for optimality [50], [52], [53]. The KKT conditions for all the parameters can be denoted as follows:

$$\begin{cases} \exists \alpha_1, \dots, \alpha_T \geq 0, \text{ s.t. } \sum_{t=1}^T \alpha_t = 1 \text{ and } \sum_{t=1}^T \alpha_t \nabla_{\theta_{\text{sh}}} \mathcal{L}_t(\theta_{\text{sh}}, \theta_t) = 0, \\ \forall t, \nabla_{\theta_t} \mathcal{L}_t(\theta_{\text{sh}}, \theta_t) = 0. \end{cases} \quad (6)$$

Any solution that satisfies these conditions is called a *Pareto stationary point* [24], [50]. It should be noted that although every Pareto optimal point is Pareto stationary, the converse does not necessarily hold. To meet these conditions, we can further represent the optimization problem as:

$$\min_{\alpha_1, \dots, \alpha_T} \left\| \sum_{t=1}^T \alpha_t \nabla_{(\theta_{\text{sh}}, \theta_t)} \mathcal{L}_t(\theta_{\text{sh}}, \theta_t) \right\|_2^2, \quad (7)$$

subject to  $\sum_{t=1}^T \alpha_t = 1$  and  $\alpha_t \geq 0$  for all  $t$ . [24] and [50] showed that if the solution to this optimization problem is 0, the KKT conditions are satisfied. Otherwise, the solution provides a descent direction that can improve all tasks. Based on the above research, we propose the parameter self-sharing method to optimize the dual objectives of AU.

## IV. ATTRCLOAK DESIGN

### A. Overview

As shown in Fig. 3, AttrCloak advocates for two key properties to achieve effective attribute unlearning: i) **Attributes Indistinguishable**, which effectively removes the association between user-marked attributes for deletion and user embeddings to prevent privacy leakage, making user attributes  $a_j$  cannot be inferred from user embeddings  $\mathbf{em}'_i$ :

$$\min_{\mathbf{em}'_i} \sum_{a_j \in \mathcal{A}} D(\text{infer}(\mathbf{em}'_i), a_j), \quad (8)$$

where  $D$  is a measure of attribute distinguishability and  $\text{infer}$  is a function to infer the attributes. Specifically,  $D$  can be a metric function used to measure the difference between the predicted attribute  $a_j$  obtained from the AIA model ( $\text{infer}(\mathbf{em}'_i)$ ) and a random guesser:

$$D \triangleq \sum_{a_j \in \mathcal{A}} |\text{Acc}(\text{infer}(\mathbf{em}'_i), a_j) - \text{Acc}(\text{Random}(\mathbf{em}'_i), a_j)| \quad (9)$$

where  $\text{Acc}(\text{infer}(\mathbf{em}'_i), a_j)$  is the classification accuracy for  $a_j$  predicted by the AIA model from  $\mathbf{em}'_i$ , and  $\text{Acc}(\text{Random}(\mathbf{em}'_i), a_j)$  is the accuracy of random guessing,

typically  $\frac{1}{|U_{a_j}|}$ ; and ii) **Recommendation Knowledge Retention**, which ensures recommendation performance remains consistent post-unlearning:

$$\min_{\mathbf{em}'_i} \text{Dist}(M(\mathbf{em}_i), M(\mathbf{em}'_i)), \quad (10)$$

where  $\text{Dist}$  represents a measure of performance change of recommendation model  $M(\cdot)$ . Traditional unlearning methods [54], [55], [56] rely on a fixed threshold to mitigate catastrophic unlearning. In contrast, to simultaneously optimize these objectives for a balance between privacy and performance, we formulate AU as a dual-objective optimization problem for user embeddings via parameter self-sharing, minimizing the impact on recommendation quality. For the recommendation objective, we can relatively easily achieve this by limiting updates through the original training loss or regularization loss. For the privacy objective, we propose an information-theoretic [57] based loss function component. The core idea is to minimize the mutual information between the embedding distribution and the attribute to be forgotten  $a_j \in \mathcal{A}$ :

$$\min_{\mathbf{em}'_i} \sum_{a_j \in \mathcal{A}} \mathcal{I}(\mathbf{em}'_i; a_j). \quad (11)$$

For the overall workflow, AttrCloak fine-tunes a pre-trained RS model to safeguard sensitive attributes set  $\mathcal{A}$  against AIA. It iteratively updates new user embeddings  $\mathbf{em}'_i$  from  $\mathbf{em}_i$  through a carefully designed multi-component loss function between attribute information loss and recommendation knowledge retention loss. It adapts to both data-dependent and data-free scenarios by selecting an appropriate retention loss.

### B. Compositional Attribute Unlearning.

We propose a multi-component loss function to achieve the goals of attribute indistinguishability and recommendation performance retention across different data environments.

1) **Private Attributes Information Loss**: Directly computing mutual information  $\mathcal{I}(\mathbf{em}'_i; a_j)$ ,  $a_j \in \mathcal{A}$  is challenging because it requires estimating joint and marginal probability distributions, which is computationally expensive due to unknown real distributions, complex non-linear dependencies, and the curse of dimensionality. To address this, we approximate the mutual information using a variational upper bound based on the Kullback–Leibler (KL) divergence [58] (detailed in §V-A), which measures the difference between two probability distributions. The resulting loss function is as follows:

$$\begin{aligned} \mathcal{L}_j^{AU} &= I(\mathbf{em}_i; a_j) \\ &\approx \sum_{k=1}^{|U_{a_j}|} \frac{|S_{a_j=C_k}|}{|S_{a_j}|} D_{\text{KL}}(q_\phi(\mathbf{em}_i | \mathbf{Em}_{a_j=C_k}) \| p(\mathbf{em}_i)), \end{aligned} \quad (12)$$

where  $|U_{a_j}|$  represents the number of different labels in attribute  $a_j$ ,  $S_{a_j=C_k} = \{\mathbf{em}_i \mid a_j(\mathbf{em}_i) = C_k\}$  represents the set of user embeddings where the attribute  $a_j$  belongs to class  $C_k$ , and  $|S_{a_j=C_k}|$  denotes the number of embeddings in this set.  $|S_{a_j}|$  denotes the total number of user embeddings across all attribute classes and  $\mathbf{Em}_{a_j=C_k}$  represents the input  $\mathbf{Em}$  where the label of attribute  $a_j$  is  $C_k$ .  $D_{\text{KL}}(q \| p)$

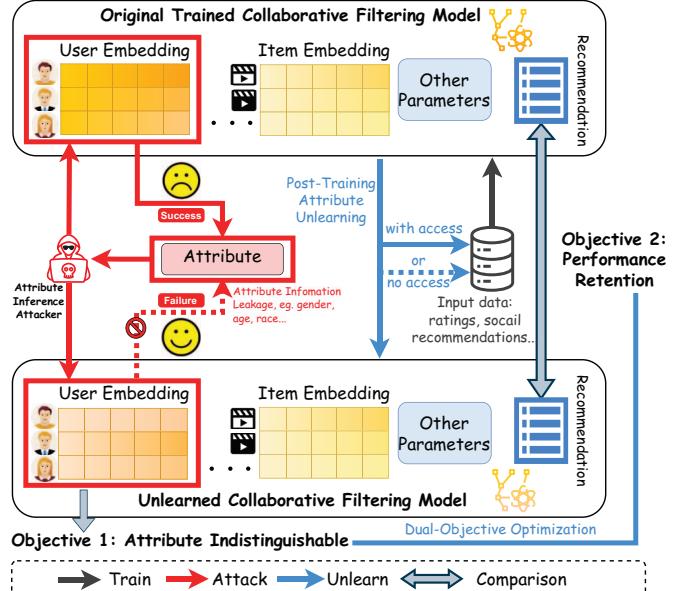


Fig. 3: Overview of AttrCloak for Attribute Unlearning.

denotes the KL divergence between the variational distribution  $q_\phi(\mathbf{em}_i | \mathbf{Em}_{a_j=C_k})$ , which approximates the embedding distribution conditioned on the attribute  $a_j = C_k$ , and the prior distribution  $p(\mathbf{em}_i)$ , which represents the embedding distribution. By minimizing this KL divergence, we effectively reduce the information in the embedding  $\mathbf{em}_i$  that is related to the attribute  $a_j$ , thereby achieving the goal of AU.

Intuitively, we aim to compute the distribution of different attribute classes  $a_j$  and minimize their KL divergence from the same global distribution. By introducing a variational distribution such as Gaussian, we can efficiently approximate mutual information via its variational upper bound using the KL divergence. This greatly simplifies the optimization process. Variational approximations also provide a tractable solution that scales well to large datasets and high-dimensional embeddings [59], [60]. The detailed steps are as follows:

First, we compute the user embedding distribution for each class. Since AttrCloak 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, usually by fitting a Gaussian distribution or a suitable model [61], [62], [63], [64]. 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  $C_k \in a_j$ , the mean vector  $\mu_{j,k}$  is computed as:

$$\mu_{j,k} = \frac{1}{|S_{a_j=C_k}|} \sum_{\mathbf{em}_i \in S_{a_j=C_k}} \mathbf{em}_i. \quad (13)$$

We compute the covariance matrix  $\Sigma_i$  as below:

$$\Sigma_{j,k} = \frac{1}{|S_{a_j=C_k}|} \sum_{\mathbf{em}_i \in S_{a_j=C_k}} (\mathbf{em}_i - \mu_{j,k})(\mathbf{em}_i - \mu_{j,k})^T. \quad (14)$$

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

$$\mu_{\text{global}} = \frac{|S_{a_j=C_k}|}{|S_{a_j}|} \sum_{k=1}^N \mu_{j,k}. \quad (15)$$

Similarly, the global covariance matrix  $\Sigma_{\text{global}}$  is computed as:

$$\Sigma_{\text{global}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{em}_i - \mu_{\text{global}})(\mathbf{em}_i - \mu_{\text{global}})^T. \quad (16)$$

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:

$$D_{\text{KL}}(\mathcal{N}(\mu_{j,k}, \Sigma_{j,k}) \| \mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})) = \frac{1}{2} \left( \log \frac{\det(\Sigma_{\text{global}})}{\det(\Sigma_{j,k})} - d \right) + \frac{1}{2} \left( \text{Tr}(\Sigma_{\text{global}}^{-1} \Sigma_{j,k}) + (\mu_{\text{global}} - \mu_{j,k})^T \Sigma_{\text{global}}^{-1} (\mu_{\text{global}} - \mu_{j,k}) \right), \quad (17)$$

where  $d$  is the dimensionality of the user embeddings,  $\det(\Sigma)$  is the determinant of the covariance matrix, and  $\text{Tr}$  denotes the trace operation. This allows us to calculate the unlearning loss function  $\mathcal{L}_j^{AU}$  for  $a_j$  as below:

$$\mathcal{L}_j^{AU} = \sum_{k=1}^{|U_{a_j}|} \frac{|S_{a_j=C_k}|}{|S_{a_j}|} D_{\text{KL}}(\mathcal{N}(\mu_{j,k}, \Sigma_{j,k}) \| \mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})). \quad (18)$$

Minimizing the loss function  $\mathcal{L}_j^{AU}$  ensures that embeddings across different sensitive classes in  $a_j$  have similar distributions, making it difficult for AIA attackers to infer sensitive attributes. The overall attribute loss  $\mathcal{L}^{AU}$  can be expressed as:

$$\mathcal{L}^{AU} = \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{A}|} \mathcal{L}_j^{AU}. \quad (19)$$

**2) Recommendation Knowledge Retention Loss:** Since AU may lead to a degradation in recommendation performance, additional design is necessary to achieve RS performance retention. An intuitive approach is to directly use the recommendation loss function from the training phase (e.g., binary cross-entropy (BCE) [65], root mean squared error (RMSE) [66], or Bayesian personalized ranking (BPR) [42] loss) as the optimization objective to maintain recommendation quality. This recommendation loss  $\mathcal{L}^{Rec}$  is defined as:

$$\mathcal{L}^{Rec} = \mathcal{L}_{BCE/BPR, \dots}(s_\psi(f_{p,\varphi_u}(u), f_{p,\varphi_v}(v)), \mathbf{R}), \quad (20)$$

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  $v_j$ . Since the interaction data may be inaccessible due to privacy restrictions or data modifications post training, the loss  $\mathcal{L}^{Rec}$  might no longer be applicable. In such scenarios, we use a L2-regularization [2], [67] term  $\mathcal{L}^{Reg}$  as below:

$$\mathcal{L}^{Reg} = \sum_{i=1}^{|U|} \|\mathbf{em}_i - \mathbf{em}'_i\|_2^2 = \sum_{i=1}^{|U|} \sum_{j=1}^d (\mathbf{em}_{i,j} - \mathbf{em}'_{i,j})^2, \quad (21)$$

where  $\mathbf{em}_i$  and  $\mathbf{em}'_i$  represent the user embeddings before and after unlearning, respectively.  $\mathcal{L}^{Reg}$  restricts the range of user embedding updates, preventing drastic changes in user embeddings and thus leveraging the prior learning. The underlying rationale is that closer model parameters typically lead to more consistent model performance.

### C. Dual-objective Optimization with Parameter Self-sharing.

We propose an approach to achieve an optimal balance between unlearning sensitive attributes and retaining recommendation knowledge. This is modeled as a dual-objective optimization problem as follows:

$$\min_{\mathbf{em}'_i} \mathcal{L}^{All} = \min_{\mathbf{em}'_i} \alpha_u \cdot \mathcal{L}^{AU} + \alpha_r \cdot \mathcal{L}^R, \quad (22)$$

where  $\mathcal{L}^{AU}$  and  $\mathcal{L}^R$  are the loss functions for attribute unlearning and recommendation knowledge retention goals, respectively. Here,  $\mathcal{L}^{AU}$  is defined in Eq. (19), and when interaction data is available during unlearning,  $\mathcal{L}^R = \mathcal{L}^{Rec}$  from Eq. (20) minimizes performance loss using the original recommendation loss. Otherwise,  $\mathcal{L}^R = \mathcal{L}^{Reg}$  in Eq. (21), which limits performance loss via regularization. The weights  $\alpha_u$  and  $\alpha_r$  are either statically or dynamically computed to balance the two tasks effectively. Given the complexity of finding optimal settings for Eq. (22) through heuristic methods or grid search across multiple proportions [23], [24], we reformulate this dual-objective unlearning problem as a multi-objective optimization problem to achieve Pareto optimality between the unlearning and retention tasks.

#### Definition 1 (Pareto Optimality of Attribute Unlearning).

- (a) For the unlearning and knowledge retention goals, a solution  $\mathbf{em}'_i$  is superior to  $\mathbf{em}'_i$  if and only if  $\mathcal{L}^{AU}(\mathbf{em}'_i) \leq \mathcal{L}^{AU}(\mathbf{em}'_i)$  and  $\mathcal{L}^R(\mathbf{em}'_i) \leq \mathcal{L}^R(\mathbf{em}'_i)$ .
- (b) A solution  $\mathbf{em}'_i$  is a Pareto optimal solution if no  $\mathbf{em}'_i$  outperforms  $\mathbf{em}'_i$  in the sense of (a).

The set of Pareto optimal solutions is referred to as the Pareto set  $P(\mathbf{em}'_i)$ . Typically, local optima of multi-objective optimization problems are achieved through gradient descent (GD) [50], [52], [53], [68], [69], identifying Pareto stationary points, which are necessary conditions for Pareto optimality. We also define Pareto stationarity as follows.

#### Definition 2 (Pareto Stationary Point for AU).

For the unlearning and recommendation knowledge retention objectives, if there exists a scalar  $\alpha_u \in [0, 1]$  such that

$$\alpha_u \nabla_{\mathbf{em}'_i} (\mathcal{L}^{AU}(\mathbf{em}'_i)) + (1 - \alpha_u) \nabla_{\mathbf{em}'_i} (\mathcal{L}^R(\mathbf{em}'_i)) = 0, \quad (23)$$

then  $\mathbf{em}'_i$  is a Pareto stationary point.

Any solution meeting the conditions is termed a Pareto stationary point, and each Pareto optimal point is a Pareto stationary point. To locate these points, we reformulate the dual-objective unlearning problem as the following optimization:

$$\min_{\alpha_u} \left\| \alpha_u \nabla_{\mathbf{em}'_i} (\mathcal{L}^{AU}) + (1 - \alpha_u) \nabla_{\mathbf{em}'_i} (\mathcal{L}^R) \right\|_2^2, \quad (24)$$

where  $\alpha_u \in [0, 1]$ . According to the literature [24], [50], the solution of this equation is either a Pareto stationary point if zero or provides an optimization direction for both unlearning and recommendation knowledge retention if non-zero. The optimal  $\alpha_u$  for Eq. (24) can be solved analytically as a convex quadratic problem with linear constraints. Our goal is to minimize the squared norm of this combined gradient. Expanding the squared norm, taking the derivative with respect to  $\alpha_u$  and setting it to zero, we obtain:

$$2\alpha_u \left\| \nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^R) \right\|_2^2 + 2(\nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^R))^T \nabla_{\mathbf{em}'_i}(\mathcal{L}^R) = 0. \quad (25)$$

Given fixed gradient values per update, we can derive the one-dimensional quadratic solution for  $\alpha_u$  as:

$$\hat{\alpha}_u = \left[ \frac{(\nabla_{\mathbf{em}'_i}(\mathcal{L}^R) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}))^T \nabla_{\mathbf{em}'_i}(\mathcal{L}^R)}{\|\nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^R)\|_2^2} \right]_{+,1}, \quad (26)$$

where  $[.]_{+,1}^T$  denotes clipping to  $[0, 1]$  for  $\mathcal{L}^R = \mathcal{L}^{Rec}$ , i.e.,  $[x]_{+,1}^T = \max(\min(x, 1), 0)$ . For  $\mathcal{L}^R = \mathcal{L}^{Reg}$ , to avoid stagnation of gradient updates caused by regularization, the value is clipped to  $[0.1, 1]$ , i.e.,  $[x]_{+,1}^T = \max(\min(x, 1), 0.1)$ .

We can reframe the combined gradient in Eq. (24) as  $\alpha_u (\nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^R)) + \nabla_{\mathbf{em}'_i}(\mathcal{L}^R)$ . This reconstruction highlights the role of  $\alpha_u$  in balancing the gradient difference  $\nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^R)$  with the recommendation gradient  $\nabla_{\mathbf{em}'_i}(\mathcal{L}^R)$ . Our goal is to adjust  $\alpha_u$  to modulate both the magnitude and the sign of the gradient difference, such that the norm of the sum gradient is minimized. This means we aim for the sum vector to be nearly orthogonal to  $\nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU}) - \nabla_{\mathbf{em}'_i}(\mathcal{L}^R)$ . Thus,  $\alpha_u$  is the dot product divided by the magnitude of the gradient difference, which represents the proportion of the projection distance of the recommendation gradient along the direction of the gradient difference. To ensure that  $\alpha_u$  does not overly favor one objective, thereby avoiding the sacrifice of one objective in favor of the other, we clip  $\alpha_u$  to a specific range, thereby maintaining a balance between the two objectives.

By adjusting  $\hat{\alpha}_u$  dynamically and continuously during the AU updates, a balance between the objectives is achieved. With the weight determined, we optimize the total loss function  $\mathcal{L}^{All}$  using gradient descent, which is proven to converge to a point in the Pareto set [50]. AttrCloak is designed to fine-tune only the entire user representation layer to quickly safeguard user privacy attributes, including  $f_\theta = \{f_{\varphi_u}; f_p\}$ , with item embeddings  $f_{\varphi_v}$  and scoring layers  $s_\psi$  remaining fixed during AU. Although  $f_{\varphi_v}$  and  $s_\psi$  are involved in computing the  $\mathcal{L}^R$ , they do not participate in our multi-task optimization and are therefore not classified as “task-specific” layers. This mechanism is referred to as *parameter self-sharing*.

**Summary.** AttrCloak serves as a complementary solution for RSs, offering adaptability to diverse training methods. It effectively safeguards sensitive attributes against AIAs. To enable AU even in the absence of interaction data, we support two post-training unlearning scenarios, as outlined in Algorithm 1:

---

**Algorithm 1** Attribute Unlearning in RS with AttrCloak

---

**Input:** Pre-trained model with user embeddings  $\mathbf{em}_i$ , sensitive attributes  $\mathcal{A}$ , learning rate  $\eta$ , interaction matrix  $\mathbf{R}$  (optional).  
**Output:** Fine-tuned user embeddings  $\mathbf{em}'_i$  after unlearning.  
1: Initialize  $\mathbf{em}'_i = \mathbf{em}_i = f_p(f_{\varphi_u}(u_i))$ .  
2: **while** not converged **do**  
3:   **for all** attribute  $a_j \in \mathcal{A}$  **do**  
4:     Estimate class distributions  $\mathcal{N}(\mu_{j,k}, \Sigma_{j,k})$  for  $|U_{a_j}|$  classes using Eq. (13), (14).  
5:     Compute global distribution  $\mathcal{N}(\mu_{global}, \Sigma_{global})$  using Eq. (15), (16).  
6:     Compute loss  $\mathcal{L}_j^{AU}$  using Eq. (18).  
7:   **end for**  
8:   Compute total attribute unlearning loss  $\mathcal{L}^{AU}$  as in Eq. (19).  
9:   **if** interaction data  $\mathbf{R}$  is available **then**  
10:     Compute recommendation loss  $\mathcal{L}^{Rec}$  using Eq. (20).  
11:     Set  $\mathcal{L}^R = \mathcal{L}^{Rec}$ .   ▷ **Data-Dependent (DD) Protection**  
12:   **else**  
13:     Compute regularization loss  $\mathcal{L}^{Reg}$  defined in Eq. (21).  
14:     Set  $\mathcal{L}^R = \mathcal{L}^{Reg}$ .   ▷ **Data-Free (DF) Protection**  
15:   **end if**  
16:   Define dual-objective loss  $\mathcal{L}^{All}$  as in Eq. (22).  
17:   Compute gradients  $\nabla_{\mathbf{em}'_i}(\mathcal{L}^{AU})$  and  $\nabla_{\mathbf{em}'_i}(\mathcal{L}^R)$ .  
18:   Solve optimization using Eq. (26) to find optimal  $\alpha_u, \alpha_r$ .  
19:   Update user embeddings using GD:  $\mathbf{em}'_i: \mathbf{em}'_i \leftarrow \mathbf{em}'_i - \eta \nabla_{\mathbf{em}'_i} \mathcal{L}^{All}$ , as well as  $(\varphi_u, p) \leftarrow (\varphi_u, p) - \eta \nabla_{(\varphi_u, p)} \mathcal{L}^{All}$ .  
20: **end while**  
21: **return**  $\mathbf{em}'_i$ .

---

- **i) Data-Dependent Protection:** When interaction data is available, AU is guided by a combined loss function:  $\mathcal{L}^{All} = \alpha_u \cdot \mathcal{L}^{AU} + \alpha_r \cdot \mathcal{L}^{Rec}$ .  $\mathcal{L}^{AU}$  uses a variational upper bound on mutual information, minimizing the KL-divergence of  $em$  between inter-class and global distributions to decorrelate sensitive attributes. To preserve recommendation performance, the RS training loss is set as  $\mathcal{L}^{Rec}$ . The dual-objective optimization framework adjusts  $\alpha_u$  and  $\alpha_r$  through parameter self-sharing to balance the two objectives.
- **ii) Data-Free Protection:** When interaction data is unavailable,  $\mathcal{L}^{Rec}$  cannot be calculated. To mitigate recommendation performance reduction, we use a regularization loss  $\mathcal{L}^{Reg}$ , with the combined loss given by  $\mathcal{L}^{All} = \alpha_u \cdot \mathcal{L}^{AU} + \alpha_r \cdot \mathcal{L}^{Reg}$ . Similarly,  $\alpha_u$  and  $\alpha_r$  are optimized within a dual-objective parameter self-sharing framework to achieve a Pareto-optimal solution between the two goals of AU.

## V. THEORETICAL ANALYSIS

### A. Privacy Analysis.

Our core theoretical insight is that attribute privacy can be protected by minimizing the mutual information  $I(\mathbf{em}'_i; a_j)$ , which quantifies how much information about the sensitive attribute  $a_j \in \mathcal{A}$  is revealed by the embedding  $\mathbf{em}'_i$ . Due to the upper bound for mutual information capturing the maximum potential privacy leakage stemming from user embeddings, we minimize it to ensure embeddings contain no statistically discernible traces of  $a_j$ . Mutual information is commonly represented in terms of conditional entropy as follows:

$$I(\mathbf{em}'_i; a_j) = \mathbb{E}_{p(a_j, \mathbf{em}'_i)} \left[ \log \frac{p(\mathbf{em}'_i | a_j)}{p(\mathbf{em}'_i)} \right]. \quad (27)$$

By substituting the conditional distribution with a variational distribution  $q(\mathbf{em}'_i)$  in Eq. (27), we obtain:

$$\begin{aligned}\mathcal{I}(\mathbf{em}'_i; a_j) &= \mathbb{E}_{p(a_j, \mathbf{em}'_i)} \left[ \log \frac{p(\mathbf{em}'_i|a_j)q(\mathbf{em}'_i)}{p(\mathbf{em}'_i)q(\mathbf{em}'_i)} \right] \\ &= \mathbb{E}_{p(a_j, \mathbf{em}'_i)} \left[ \log \frac{p(\mathbf{em}'_i|a_j)}{q(\mathbf{em}'_i)} \right] - \mathbb{E}_{p(\mathbf{em}'_i)} \left[ \frac{p(\mathbf{em}'_i)}{q(\mathbf{em}'_i)} \right] \\ &= \mathbb{E}_{p(a_j)} [KL(p(\mathbf{em}'_i|a_j)||q(\mathbf{em}'_i))] - KL(p(\mathbf{em}'_i)||q(\mathbf{em}'_i)).\end{aligned}\quad (28)$$

$KL(\cdot|\cdot)$  denotes the Kullback-Leibler divergence between the conditional distribution of the subclass embeddings and the marginal distribution of all embeddings. Since KL divergence is non-negative, we derive an upper bound as below:

$$\mathcal{I}(\mathbf{em}'_i; a_j) \leq \mathbb{E}_{p(a_j)} [KL(p(\mathbf{em}'_i|a_j)||q(\mathbf{em}'_i))]. \quad (29)$$

By choosing an appropriate  $q(\mathbf{em}'_i)$  close to  $p(\mathbf{em}'_i)$ , we can effectively control the magnitude of mutual information. To incorporate each sensitive attribute  $a_j$  into the upper bound expression, we calculate the subclass distributions within this bound. Assuming  $a_j$  has  $k$  subclasses, denoted as  $a_j^1, a_j^2, \dots, a_j^k$ , each subclass's conditional distribution for  $\mathbf{em}'_i$  is  $p(\mathbf{em}'_i|a_j)$ , so the upper bound can be expressed as:

$$\mathcal{I}(\mathbf{em}'_i; a_j) \leq \sum_{i=1}^k p(a_j^i) KL(p(\mathbf{em}'_i|a_j^i)||p(\mathbf{em}'_i)), \quad (30)$$

where  $p(\mathbf{em}'_i|a_j^i)$  is the conditional probability given the class  $a_j^i$ , and  $p(\mathbf{em}'_i)$  is the marginal probability, typically given by:

$$p(\mathbf{em}'_i) = \sum_{i=1}^k p(a_j^i)p(\mathbf{em}'_i|a_j^i). \quad (31)$$

$p(a_j^i)$  is the prior probability of subclass  $a_j^i$ . For sensitive attributes like gender, we compute  $p(a_j^i)$  as follows:

$$p(a_j^i) = \frac{|S_{a_j=i}|}{|S_{a_j}|}, \quad (32)$$

where  $|S_{a_j=i}|$  is the number of users in attribute class  $i$ . This satisfies  $\sum_{i=1}^k p(a_j^i) = 1$ . Consequently, the upper bound of mutual information simplifies to:

$$\sum_{i=1}^k \frac{|S_{a_j=i}|}{|S_{a_j}|} KL(p(\mathbf{em}'_i|a_j^i)||p(\mathbf{em}'_i)). \quad (33)$$

This upper bound aligns with the loss objective function in Eq. (12). To reduce computational complexity, we typically approximate the conditional distribution of embeddings in high-dimensional space as a Gaussian. For each subclass  $a_j^i$  of  $\mathbf{em}'_i$ , we use a Gaussian distribution  $\mathcal{N}(\mu_{j,i}, \Sigma_{j,i})$ , where  $\mu_{j,i}$  and  $\Sigma_{j,i}$  represent the mean and covariance matrix, respectively. Similarly, the global embedding distribution  $p(\mathbf{em}'_i)$  can be approximated by a Gaussian distribution with mean  $\mu_{\text{global}}$  and covariance matrix  $\Sigma_{\text{global}}$ . Ultimately, the upper bound of mutual information can be minimized as follows:

$$\mathcal{L}_j^a = \sum_{i=1}^k \frac{|S_{a_j=i}|}{|S_{a_j}|} D_{\text{KL}}(\mathcal{N}(\mu_{j,i}, \Sigma_{j,i})||\mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})). \quad (34)$$

where  $k = |U_{a_j}|$  is the number of subclasses for the privacy attribute  $a_j$ , consistent with the conclusions in Eq. (18). Intuitively, this loss function encourages the embeddings of

TABLE III: 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%

different attribute subclasses to align with the global distribution, thereby reducing the distinguishability of attributes and defending against attribute inference attacks.

### B. Complexity Analysis.

We analyze the complexity of AttrCloak by breaking down each component and focusing on the key computational costs.

**Computing Privacy Attribute Loss.** The main computational cost for calculating  $\mathcal{L}^{AU}$  arises from computing the covariance matrix, which has a complexity of  $O(|U| \cdot d^2)$ . The Kullback-Leibler (KL) divergence calculation for the distributions of  $|U_{a_j}|$  sensitive attribute classes has a complexity of  $O(|U_{a_j}| \cdot d^3)$ . Thus, the total complexity for computing  $\mathcal{L}^{AU}$  for all attributes is  $O(\sum_{a_j \in \mathcal{A}} |U_{a_j}| \cdot d^3 + |\mathcal{A}| \cdot |U| \cdot d^2)$ . Since the embedding dimension  $d$  is typically small, the computation of privacy attribute loss remains manageable.

**Computing Recommendation Knowledge Retention Loss.** For computing  $\mathcal{L}^{Rec}$ , in the Data-Dependent (DD) scenario, the complexity is  $O(|U| \cdot |B| \cdot d)$ , where  $B$  is the batch size. In the Data-Free (DF) Protection scenario, the complexity of the regularization loss  $\mathcal{L}^{Reg}$  is  $O(|U| \cdot d)$ .

**Dual-Objective Optimization.** The dual-objective optimization method involves gradient calculations for  $\mathcal{L}^{AU}$  and  $\mathcal{L}^R$ , with a complexity of  $O(|U| \cdot d)$ .

To sum up, given that  $|U|$  and  $|V|$  are typically much larger than  $|\mathcal{A}|$ ,  $|U_{a_j}|$ , and  $d$ , the dominant term of the overall complexity for each iteration in the DD scenario is  $O(|U| \cdot |B| \cdot d)$ , corresponding to the recommendation loss during the training phase. The unlearning loss  $\mathcal{L}^{AU}$  does not significantly affect complexity. For the DF scenario, the complexity for each iteration is primarily determined by  $d$ , which is much smaller than  $|U|$  and is fixed in advance. Note that since user embeddings must always be updated, the complexity of any privacy-preserving algorithm cannot be lower than  $O(|U| \cdot d)$ . This demonstrates that the overall loss of AttrCloak-DF has low complexity, allowing for quick computation, and  $|V|$  does not affect the method's scalability.

## VI. EVALUATIONS

### A. Evaluation Setup.

**Testbed.** We implement AttrCloak using Python 3.8 and PyTorch 2.2.0, and run experiments on NVIDIA A100 GPUs with a 26-core CPU and 256GB RAM on Ubuntu 20.04 LTS.

**Datasets.** The experiments were conducted on four datasets as shown in Table III: MovieLens-100K (ML-100K) and MovieLens-1M (ML-1M) [40] for movie ratings, ModCloth [70] for clothing sales, and Last.FM-1K [36] for music

TABLE IV: 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. The Respective Optimal Methods are Highlighted with a Gray Background under Both Data Environments.

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	0.7626	0.4496	0.2281	0.7955	0.4164	0.1995	0.7648	0.7513	0.6040	0.5876	
	<b>DD</b>	AttrCloak-DD	0.5822	0.1989	0.0995	0.5608	0.1692	0.1024	0.5566	0.5485	0.3531	0.3493
		RAP [6]	0.9310	0.8143	0.6034	0.9917	0.9611	0.8079	0.9903	0.9660	0.8789	0.9281
	<b>DF</b>	BlurMe [26]	0.6300	0.3170	0.1525	0.6641	0.2957	0.1482	0.6606	0.5813	0.4023	0.4161
		LDP-SH [22]	0.6698	0.2692	0.1472	0.6854	0.3171	0.1217	0.6289	0.5977	0.4351	0.3783
	<b>DF</b>	AttrCloak-DF	0.5995	0.2401	0.1419	0.6203	0.2503	0.0772	0.5663	0.5612	0.1892	0.3279
		U2U-R [2]	0.9987	0.9947	0.9788	0.9999	0.9992	0.9999	0.9999	0.9937	0.9823	0.9987
		D2D-R [2]	0.6538	0.2798	0.1989	0.7113	0.3180	0.1551	0.6386	0.5927	0.3405	0.3960
MLP Attacker	Original	0.7414	0.3289	0.2454	0.7243	0.3526	0.1660	0.7653	0.6179	0.5359	0.5612	
	<b>DD</b>	AttrCloak-DD	0.5928	0.1724	0.0358	0.5217	0.1656	0.0406	0.5216	0.5549	0.1501	0.1803
		RAP [6]	0.6286	0.2056	0.0995	0.5235	0.1829	0.0257	0.5610	0.5549	0.1197	0.2421
	<b>DF</b>	BlurMe [26]	0.6605	0.2162	0.0902	0.6177	0.1573	0.0803	0.5637	0.5776	0.2245	0.2711
		LDP-SH [22]	0.6976	0.2586	0.0690	0.6475	0.1838	0.0828	0.6787	0.5422	0.3266	0.2699
	<b>DF</b>	AttrCloak-DF	0.6088	0.1950	0.0528	0.6169	0.1821	0.0555	0.5589	0.5132	0.0567	0.2106
		U2U-R [2]	0.6300	0.2003	0.1056	0.6036	0.2036	0.0927	0.5640	0.5536	0.3783	0.4288
		D2D-R [2]	0.6340	0.2162	0.0531	0.6537	0.2045	0.0348	0.5645	0.4918	0.2686	0.2863
Random Attacker		0.5000	0.1429	0.0476	0.5000	0.1429	0.0476	0.5000	0.5000	0.1469	0.1667	

listening behavior. These datasets include user-item interactions and user attributes like gender and age, making them suitable for AU research. The “Age” attribute is categorized into seven age groups, as done in MovieLens-1M, and location labels are based on continent tags from the Last.FM-1K.

**Recommendation Model and Hyperparameters.** Unless otherwise specified, we adopt the NCF model from [71] as our backbone RS model. The scoring function is defined as the dot product of user and item representations, with both embedding dimensions set to 128 and initialized using the Xavier uniform distribution. Neural CF Layers include two MLP hidden layers with the same embedding dimension. Learning rate is set to  $10^{-4}$ . We employ the BPR loss [44] as the primary objective function  $\mathcal{L}^{Rec}$ , with the optimizer set to Adam and a batch size of 512. For the LightGCN [72] model used in robustness experiments (§ VI-D), we implement three graph convolutional layers for message passing. The embeddings from all layers are aggregated using mean pooling to generate the final representations. To prevent excessive smoothing, we enhance the BPR loss with adaptive L2 regularization.

**Attacker Setting.** For selecting the attribute inference model for user embedding attacker, we utilize easily implementable and powerful machine or deep learning models, including a three-layer MLP model [73] and the XGBoost model [74]. The shadow dataset consists of embeddings with the same sensitive attribute distribution, but with only 20% of the original number of embeddings. Both threat models are employed as private attribute classifiers and trained on shadow datasets.

**Baselines.** We compare AttrCloak with the original user embedding and existing defenses against AIAs:

- **Original:** This is the user embedding before unlearning.
- **RAP [6]:** An in-training attribute protection method, RAP employs adversarial training with two components: a Bayesian personalized recommender and a private attribute

inference attacker. It is formulated as a minimax game, where the recommender minimizes the recommendation loss, while the attacker maximizes the attack gain.

- **BlurMe [26]:** This method uses a blurring mechanism to add fake perturbations to user interaction profiles. It reduces the inference of privacy attributes by selecting new items that are negatively correlated with the user’s actual attributes and appending their average ratings to the user profile.
- **LDP-SH [22]:** This method requires retraining and introduces noise to user-item interactions based on  $\epsilon$ -differential privacy before training, thereby protecting private attributes.
- **U2U-R and D2D-R [2], [75]:** The data-free AU methods exclude training data, using user-to-user (U2U) loss and distribution-to-distribution (D2D) distance loss as attribute distinguishability losses with regularization loss as recommendation loss to achieve AU, respectively. It treats AU as a single-objective optimization with fixed weights.

For AttrCloak, this paper introduces two versions of AttrCloak: **AttrCloak-DD** and **AttrCloak-DF**. The key difference lies in whether user-item interaction data is required during the unlearning process, corresponding to data-dependent (**DD**) and data-free (**DF**) data settings, respectively.

**Evaluation Metrics.** In evaluating recommendation performance, we employ metrics widely used in RS, reporting recommendation utility by calculating the average Hit Ratio (HR) [37] and Normalized Discounted Cumulative Gain (NDCG) [38] across the ranked item lists of all test users. We truncate the ranked lists for both metrics at positions 5 and 10. Specifically, HR@K measures the percentage of relevant items that appear in the top K recommendations, while NDCG@K evaluates both the relevance and ranking quality of the recommended items, penalizing misordered items. Higher HR and NDCG values indicate better recommendation performance in terms of relevance and ranking quality. For privacy-preserving

TABLE V: Utility Results of Recommendation Performance.

Datasets	Methods	Utility Metrics			
		NDCG@5	NDCG@10	HR@5	HR@10
MovieLens-100K	Original	0.8116	0.7721	0.7979	0.7479
	DD	AttrCloak-DD RAP BlurMe LDP-SH	0.8172 0.5819 0.8053 0.6183	0.7769 0.5590 0.7685 0.5838	0.8034 0.5784 0.7941 0.6174
		AttrCloak-DF U2U-R D2D-R	0.7728 0.6952 0.6685	0.7531 0.6735 0.6439	0.7661 0.6895 0.6614
		AttrCloak-DF U2U-R D2D-R	0.7728 0.6952 0.6685	0.7531 0.6735 0.6439	0.7661 0.6895 0.6614
		AttrCloak-DF U2U-R D2D-R	0.7728 0.6952 0.6685	0.7531 0.6735 0.6439	0.7661 0.6895 0.6614
	DF	Original	0.6473	0.6095	0.6324
		AttrCloak-DD RAP BlurMe LDP-SH	0.6554 0.6518 0.6468 0.4907	0.6176 0.6224 0.6089 0.4691	0.6404 0.6285 0.5979 0.4928
		AttrCloak-DF U2U-R D2D-R	0.6523 0.6750 0.6318	0.6217 0.6243 0.6017	0.6277 0.5904 0.5981
		AttrCloak-DF U2U-R D2D-R	0.6523 0.6750 0.6318	0.6217 0.6243 0.6017	0.6277 0.5904 0.5981
		AttrCloak-DF U2U-R D2D-R	0.6523 0.6750 0.6318	0.6217 0.6243 0.6017	0.6277 0.5904 0.5981
MovieLens-1M	Original	0.7297	0.7483	0.7678	0.8203
	DD	AttrCloak-DD RAP BlurMe LDP-SH	0.7928 0.3842 0.7168 0.5647	0.8093 0.3860 0.7345 0.6004	0.8292 0.3956 0.7512 0.6262
		AttrCloak-DF U2U-R D2D-R	0.5470 0.5084 0.5082	0.5523 0.5053 0.5053	0.5589 0.5162 0.5164
		AttrCloak-DF U2U-R D2D-R	0.5470 0.5084 0.5082	0.5523 0.5053 0.5053	0.5589 0.5162 0.5164
		AttrCloak-DF U2U-R D2D-R	0.5470 0.5084 0.5082	0.5523 0.5053 0.5053	0.5589 0.5162 0.5164
	DF	Original	0.6696	0.6318	0.6556
		AttrCloak-DD RAP BlurMe LDP-SH	0.6655 0.5693 0.6452 0.6113	0.6281 0.5297 0.6008 0.5703	0.6498 0.5556 0.5694 0.5943
		AttrCloak-DF U2U-R D2D-R	0.6060 0.5135 0.5657	0.5707 0.4788 0.5313	0.5888 0.4972 0.5498
		AttrCloak-DF U2U-R D2D-R	0.6060 0.5135 0.5657	0.5707 0.4788 0.5313	0.5888 0.4972 0.5498
		AttrCloak-DF U2U-R D2D-R	0.6060 0.5135 0.5657	0.5707 0.4788 0.5313	0.5888 0.4972 0.5498

performance evaluation, we assess information leakage in user embeddings using the accuracy of attribute classifiers. The AIA accuracy indicates the degree to which sensitive attributes (e.g., gender, age) can be correctly predicted from the embeddings. The AIA's goal is to achieve high attack accuracy. Our goal is to protect against AIAs, where scores closer to those of a random attacker indicate better privacy preservation. This is because excessively low accuracy could trigger the "Streisand Effect" [14], [31], [76], inadvertently leading to privacy exposure. We also compared the runtime required to implement privacy protection measures until the model converges to demonstrate the efficiency.

### B. Overall Performance.

1) **Attribute Unlearning Performance:** The classification accuracy of AIAs across various datasets reflects the effectiveness of AttrCloak in attribute unlearning, as shown in Table IV. Sensitive attributes include user gender, age group, and occupation in MovieLens-100K and MovieLens-1M, body type in ModCloth, and gender, age group, and country location in Last.FM-1K. In unprotected systems, XGBoost and MLP attacks improve classification accuracy by 28.65% and 23.44% on average compared to random attackers, respectively, highlighting significant privacy risks in user embeddings. In data-dependent (DD) scenarios, AttrCloak-DD reduces MLP attack accuracy by 20.66% on average, outperforming RAP, BlurMe, and LDP-SH by 2.68%, 4.86%, and 7.83%, respectively. For XGBoost attackers, AttrCloak-DD decreases attack efficiency

TABLE VI: Running Time Consumption of Different Unlearning Methods Under Different Data Environments.

	Time(s)	ML-100K	ML-1M	ModCloth	Last.FM-1K
DD	AttrCloak-DD	235.78	1322.18	97.05	9580.37
	RAP	580.35	4317.50	672.63	76398.79
	BlurMe	678.69	4266.94	506.78	196695.70
	LDP-SH	414.76	2972.77	474.38	51290.93
DF	AttrCloak-DF	12.81	39.20	167.52	10.03
	U2U-R	8.98	111.36	169.03	88.34
	D2D-R	7.45	56.98	88.54	6.38

by 15.18%, while RAP instead increases XGBoost attack performance by 38.34%, failing to protect attribute privacy. This suggests that adversarial training methods targeting MLP adversarial models might inadvertently increase privacy leakage in machine learning models. BlurMe and LDP-SH provide smaller reductions of 7.71% and 7.89%, respectively. In data-free (DF) scenarios, although most methods perform worse than in DD scenarios, AttrCloak-DF demonstrates superior performance. Against MLP attackers, it reduces attack efficiency by 18.08%, compared to 12.78% and 15.89% reductions achieved by U2U-R and D2D-R, respectively. Against attackers with XGBoost, U2U-R significantly increases the attack efficiency by 49.06% due to its adjustment of user embeddings into a needle-shaped distribution, while AttrCloak-DF and D2D-R reduce it by 14.65% and 7.54%, respectively. In summary, AttrCloak-DD and AttrCloak-DF demonstrate superior unlearning performance compared to baselines within their respective data environment settings.

2) **Recommendation Performance:** The evaluation of recommendation performance based on NDCG and HR is summarized in Table V. Generally, attribute unlearning tends to impact recommendation performance to varying degrees, sometimes even causing catastrophic performance degradation. Specifically, in the data-free (DF) centralized protection scenario, U2U-R results in average decreases of 11.65% and 12.00% in NDCG@5 and NDCG@10, and 12.60% and 13.06% in HR@5 and HR@10, respectively. Similarly, D2D-R leads to average reductions of 12.10% and 11.99% in NDCG@5 and NDCG@10, and 13.20% and 13.13% in HR@5 and HR@10. In contrast, AttrCloak-DF achieves better performance retention by balancing these two objectives through dual-objective optimization, with average reductions of 7.00% and 6.60% in NDCG@5 and NDCG@10, and 7.81% and 7.52% in HR@5 and HR@10. The effectiveness of AttrCloak in balancing unlearning and recommendation performance is even more pronounced in the data-dependent (DD) scenario. AttrCloak-DD outperforms the original model in most cases, achieving average improvements of 1.82% and 1.76% in NDCG@5 and NDCG@10, and 1.73% and 1.52% in HR@5 and HR@10. This is likely due to AttrCloak-DD's dynamic mitigation of data distribution bias for different attributes, coupled with its dual-objective optimization, resulting in a more balanced data distribution. In comparison, RAP shows the largest performance drops, with average declines exceeding 16% across all metrics. LDP-SH also suffers substantial per-

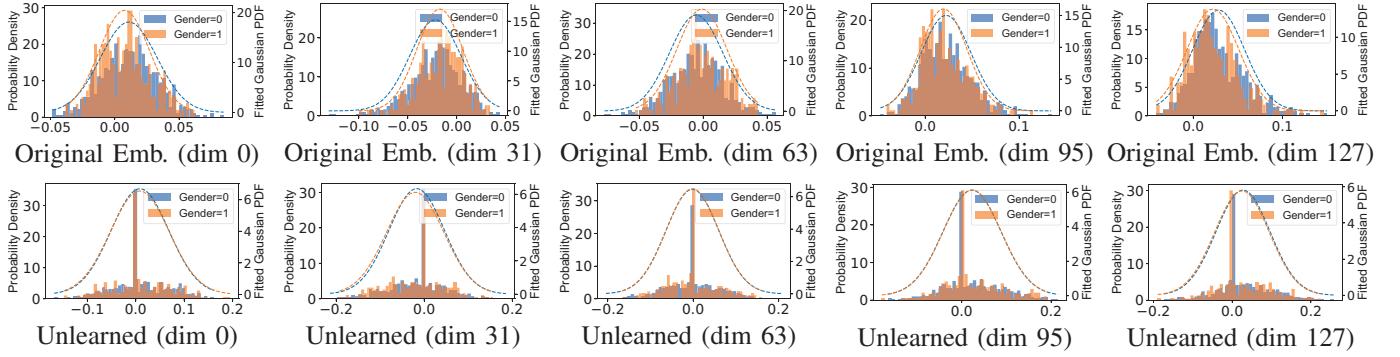


Fig. 4: Comparison of the Distribution of User Embeddings Before and After Attribute Unlearning (Gender) on the MovieLens-100K Dataset. Each Plot Represents a Selected Dimension of Embeddings; 0 and 1 Represent Female and Male, Respectively.

formance losses, with average reductions of over 10% across the four metrics. BlurMe performs relatively better, with an average reduction of 1.39% across the metrics. In summary, the dual-objective optimization of AttrCloak demonstrates superior performance retention compared to baseline methods.

3) **Unlearning Efficiency:** We recorded the runtime of AU methods to compare their efficiency, as shown in Table VI. In the data-dependent (DD) scenario, AttrCloak-DD, as a post-training method, demonstrates higher efficiency compared to in-training privacy protection methods such as RAP, BlurMe, and LDP-SH. Specifically, AttrCloak achieves greater efficiency than RAP by leveraging the pre-trained global model for fine-tuning without requiring additional MLP-based adversarial training modules. BlurMe, on the other hand, suffers from reduced efficiency due to the need for item sampling with opposing attributes, a limitation that becomes more pronounced in large-scale datasets. In the data-free (DF) scenario, the regularization method significantly accelerates the unlearning process. While AttrCloak-DF incurs slightly higher runtime costs compared to D2D-R due to its dual-objective optimization overhead, its runtime is comparable to U2U-R. Considering the trade-off between unlearning efficacy and recommendation performance retention, as well as the significant acceleration over in-training methods, we assume this runtime cost is acceptable. Overall, AttrCloak provides new insights into achieving efficient AU for privacy.

4) **Embedding Visualization:** To intuitively analyze the unlearning results, we visualize histograms of user embedding distributions before and after unlearning. Fig. 4 shows the partial dimensionality visualizations for the gender attribute in MovieLens-100K. In the embedded probability density histogram, gender is color-coded, with dashed Gaussian curves fitted to each attribute's embedding. The objective of AU is to make the different categories of sensitive attributes indistinguishable in the embeddings. Before AU, the embeddings for different categories show distinct clusters, indicating a correlation between the embeddings and sensitive attributes. After AU, the embedding distributions for different categories become more overlapping, with reduced distinguishability, which makes it more difficult for AIA attackers to infer sensitive attributes. For example, in the dimension 0, the

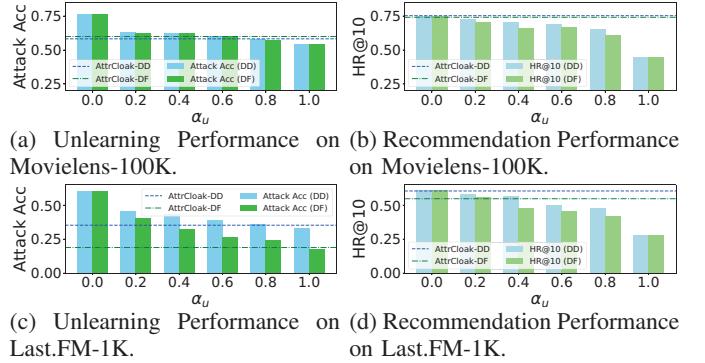


Fig. 5: Performance Comparison of Fixed Weight Parameter  $\alpha_u$  Optimization and AttrCloak Dual-Objective Optimization.

overlap area increased by 19.2% after unlearning, and the Kullback-Leibler (KL) divergence decreased by 95.9%.

### C. Ablation Study.

To assess the effectiveness of our parameter self-sharing-based dual-objective optimization method, we conducted ablation studies by replacing the dynamic dual-objective weighted optimization in Eq. (22) with fixed parameter thresholds. Specifically,  $\alpha_u$  was set to six fixed values:  $\alpha_u \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ , with  $\alpha_r = 1 - \alpha_u$  accordingly.  $\alpha_u = 0.0$  considers only the recommendation loss  $\mathcal{L}^R$ , while  $\alpha_u = 1.0$  focuses solely on the attribute unlearning loss  $\mathcal{L}^{AU}$ .

Ablation results are shown in Fig. 5. For instance, in gender AU on MovieLens-100K, Fig. 5a indicates that when  $\alpha_u \geq 0.8$ , the attack accuracy drops below that of AttrCloak-DD and AttrCloak-DF, reflecting improved privacy protection. However, Fig. 5b demonstrates a significant degradation in recommendation performance at these settings, failing to meet service quality requirements. Notably, fixed parameters  $\alpha_u = 0.2$  or  $0.4$  perform comparably to AttrCloak in some cases (e.g., 4.80% higher attack accuracy and 2.30% reduced recommendation performance for  $\alpha_u = 0.2$  in DD scenarios), but their generalization across datasets is inconsistent. For example, on Last.FM-1K (Fig. 5c-5d),  $\alpha_u = 0.2$  shows a

TABLE VII: Attribute Unlearning and Recommendation Utility Results with Another Recommendation Model (LightGCN).

Metrics		XGBoost Attacker Acc.			MLP Attacker Acc.			Recommendation Utility				
		Gender	Age	Occupation	Gender	Age	Occupation	NDCG@5	NDCG@10	HR@5	HR@10	
MovieLens-100K	DD	Original	0.7029	0.3422	0.1923	0.7520	0.3382	0.2255	0.8125	0.7847	0.7997	0.7657
		AttrCloak-DD	0.5557	0.1499	0.0849	0.5530	0.1645	0.0836	0.8218	0.7944	0.8017	0.7768
		RAP	0.8302	0.6525	0.4748	0.5477	0.1790	0.0838	0.6019	0.5726	0.5870	0.5471
		BlurMe	0.6008	0.2294	0.1300	0.6061	0.2255	0.0902	0.8136	0.7875	0.8000	0.7794
	DF	LDP-SH	0.6565	0.1923	0.1194	0.6260	0.2719	0.1326	0.6391	0.6144	0.6251	0.4962
		AttrCloak-DF	0.5066	0.1698	0.1154	0.5782	0.2069	0.0716	0.7470	0.7203	0.7343	0.7018
		U2U-R	0.9151	0.6989	0.8422	0.5875	0.2149	0.0796	0.6606	0.6255	0.6476	0.6033
		D2D-R	0.6273	0.2228	0.1658	0.6153	0.2480	0.0822	0.6371	0.6150	0.6359	0.6036
Random Attacker		0.5000	0.1429	0.0476	0.5000	0.1429	0.0476	0.5000	0.5000	0.1469	0.1667	

21.61% drop in privacy performance in Data-Free scenarios, highlighting the limitations of manual parameter tuning.

#### D. Backbone Robustness Study.

To demonstrate AttrCloak robustness across different backbone models, we conducted experiments using GNN-based RS model. Specifically, we adopted LightGCN [72], a state-of-the-art collaborative filtering model. The experimental results on the MovieLens-100K dataset for attribute unlearning and recommendation performance are presented in Table. VII. In the data-dependent (DD) scenario, AttrCloak-DD achieves an average reduction of 14.90% in MLP attacker accuracy, outperforming RAP, BlurMe, and LDP-SH by 0.31%, 4.02%, and 7.64%, respectively. Similarly, it reduces XGBoost attacker accuracy by 17.15%, surpassing RAP, BlurMe, and LDP-SH by 38.90%, 5.66%, and 5.92%. Regarding recommendation performance, AttrCloak-DD achieves slight improvements in four metrics, while RAP and LDP-SH cause catastrophic drops of 18.27–20.06%. In the data-free (DF) scenario, AttrCloak-DF shows stronger attack resistance, outperforming U2U-R by 55.48% and 0.84% against MLP and XGBoost attackers, respectively, and exceeding D2D-R by 7.47% and 2.96%. For recommendation utility, AttrCloak-DF retains performance better than U2U-R and D2D-R, with average improvements of 9.15% and 10.30% across the four metrics. In summary, AttrCloak exhibits strong effectiveness on the LightGCN backbone, highlighting its robustness across different RS models.

#### E. Scalability Study

We also evaluated the scalability across different embedding sizes, ranging from 32 to 512 dimensions, using the ModCloth dataset and the Body Shape sensitive attribute. The results in Fig. 6 indicate that both AttrCloak maintained strong attribute privacy protection, with the attacker’s accuracy reduced to near-random levels across all embedding sizes. As the embedding dimension increased, the model’s recommendation performance improved, and AttrCloak-DD even outperformed the original model after AU, which suggests that it helps mitigate underlying data distribution biases. On the other hand, AttrCloak-DF exhibited some performance degradation after AU, which is within an acceptable range and is expected given the absence of original training data.

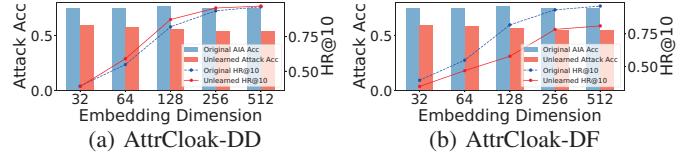


Fig. 6: Scalability of Privacy and Recommendation Performance with Different Embedding Dimensions.

## VII. CONCLUSIONS

This paper presents AttrCloak, a flexible post-training attribute unlearning framework for safeguarding user privacy in RS. AttrCloak effectively mitigates the risk of Attribute Inference Attacks by ensuring attribute indistinguishability through mutual information minimization while preserving recommendation quality using recommendation or regularization loss. By leveraging a dual-objective optimization method based on parameter self-sharing within a MTL framework, AttrCloak achieves a balance between privacy protection and recommendation utility without altering the RS training architecture. Its compatibility with both data-dependent and data-free scenarios further enhances its adaptability to dynamic data environments. Experimental evaluations on real-world datasets confirm that AttrCloak outperforms existing methods, delivering superior privacy protection and recommendation performance with minimal time overhead.

## VIII. ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (2023YFE0205700), National Natural Science Foundation of China (62341410, 62302348, 62472327), Science and Technology Development Fund, Macao S.A.R (FDCT) project (0078/2023/AMJ), UM projects (SKL-IOTSC-2024-2026, CPG2024-00022-IOTSC), the State Key Laboratory of Internet of Things for Smart City (University of Macau) Open Research Project (SKL-IoTSC(UM)/ORP05/2025) and Key R&D Program of Hubei Province (2023BAB077). Chuang Hu and Dazhao Cheng are the corresponding authors of this paper.

## REFERENCES

- [1] 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.
- [2] 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.
- [3] N. Z. Gong and B. Liu, “Attribute inference attacks in online social networks,” *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 1, pp. 1–30, 2018.
- [4] I. E. Olatunji, A. Hizber, O. Sihlovec, and M. Khosla, “Does black-box attribute inference attacks on graph neural networks constitute privacy risk?” *arXiv preprint arXiv:2306.00578*, 2023.
- [5] J. Jia and N. Z. Gong, “Attriguard: A practical defense against attribute inference attacks via adversarial machine learning,” in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 513–529.
- [6] 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.
- [7] S. Ruggieri, S. Hajian, F. Kamiran, and X. Zhang, “Anti-discrimination analysis using privacy attack strategies,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15–19, 2014. Proceedings, Part II 14*. Springer, 2014, pp. 694–710.
- [8] K. Li, G. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai, “Adversarial privacy-preserving graph embedding against inference attack,” *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6904–6915, 2020.
- [9] T. Guo, S. Guo, J. Zhang, W. Xu, and J. Wang, “Efficient attribute unlearning: Towards selective removal of input attributes from feature representations,” *arXiv preprint arXiv:2202.13295*, 2022.
- [10] J. Leyesen, “Exploring unlearning methods to ensure the privacy, security, and usability of recommender systems,” in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1300–1304.
- [11] W. Liu, J. Wan, X. Wang, W. Zhang, D. Zhang, and H. Li, “Forgetting fast in recommender systems,” *arXiv preprint arXiv:2208.06875*, 2022.
- [12] C. Chen, F. Sun, M. Zhang, and B. Ding, “Recommendation unlearning,” in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2768–2777.
- [13] 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.
- [14] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli, “Zero-shot machine unlearning,” *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2345–2354, 2023.
- [15] T. Chen, S. Zhang, and M. Zhou, “Score forgetting distillation: A swift, data-free method for machine unlearning in diffusion models,” *arXiv preprint arXiv:2409.11219*, 2024.
- [16] W. Wang, C. Zhang, Z. Tian, and S. Yu, “Machine unlearning via representation forgetting with parameter self-sharing,” *IEEE Transactions on Information Forensics and Security*, 2023.
- [17] C. Fan, J. Liu, A. Hero, and S. Liu, “Challenging forgets: Unveiling the worst-case forget sets in machine unlearning,” in *European Conference on Computer Vision*, 2025, pp. 278–297.
- [18] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, “Fast yet effective machine unlearning,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [19] M. Jegorova, C. Kaul, C. Mayor, A. Q. O’Neil, A. Weir, R. Murray-Smith, and S. A. Tsafaris, “Survey: Leakage and privacy at inference time,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 9090–9108, 2022.
- [20] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, “Fairness constraints: A flexible approach for fair classification,” *Journal of Machine Learning Research*, vol. 20, no. 75, pp. 1–42, 2019.
- [21] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.
- [22] R. Bassily and A. Smith, “Local, private, efficient protocols for succinct histograms,” in *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, 2015, pp. 127–135.
- [23] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482–7491.
- [24] O. Sener and V. Koltun, “Multi-task learning as multi-objective optimization,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [25] W. Wang, Z. Tian, C. Zhang, A. Liu, and S. Yu, “Bfu: Bayesian federated unlearning with parameter self-sharing,” in *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, 2023, pp. 567–578.
- [26] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, “Blurme: Inferring and obfuscating user gender based on ratings,” in *Proceedings of the Sixth ACM Conference on Recommender Systems*, 2012, pp. 195–202.
- [27] T. Georgiou, A. El Abbadi, and X. Yan, “Privacy cyborg: Towards protecting the privacy of social media users,” in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017, pp. 1395–1396.
- [28] 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.
- [29] J. Xu, Z. Wu, C. Wang, and X. Jia, “Machine unlearning: Solutions and challenges,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.
- [30] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, “Machine unlearning,” in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 141–159.
- [31] M. Chen, W. Gao, G. Liu, K. Peng, and C. Wang, “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*, 2023, pp. 7766–7775.
- [32] H. Kim, S. Lee, and S. S. Woo, “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, no. 19, 2024, pp. 21241–21248.
- [33] J. Bonato, M. Cotogni, and L. Sabetta, “Is retain set all you need in machine unlearning? restoring performance of unlearned models with out-of-distribution images,” in *European Conference on Computer Vision*. Springer, 2025, pp. 1–19.
- [34] W. Yuan, H. Yin, F. Wu, S. Zhang, T. He, and H. Wang, “Federated unlearning for on-device recommendation,” in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 393–401.
- [35] Y. Li, C. Chen, X. Zheng, J. Liu, and J. Wang, “Making recommender systems forget: Learning and unlearning for erasable recommendation,” *Knowledge-Based Systems*, vol. 283, p. 111124, 2024.
- [36] Ó. Celma Herrada et al., *Music Recommendation and Discovery in the Long Tail*. Universitat Pompeu Fabra, 2009.
- [37] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.
- [38] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of ir techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [39] H. Zhang, X. Yuan, and S. Pan, “Unraveling privacy risks of individual fairness in graph neural networks,” in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1712–1725.
- [40] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.
- [41] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, “Graph neural networks in recommender systems: a survey,” *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.
- [42] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.
- [43] N. Talukder, M. Ouzzani, A. K. Elmagarmid, H. Elmeleegy, and M. Yakout, “Privometer: Privacy protection in social networks,” in *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)*. IEEE, 2010, pp. 266–269.
- [44] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, “MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models,” *arXiv preprint arXiv:1806.01246*, 2018.

- [45] J. Ma, Z. Zhao, J. Chen, A. Li, L. Hong, and E. H. Chi, “Snr: Sub-network routing for flexible parameter sharing in multi-task learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 216–223.
- [46] X. Sun, R. Panda, R. Feris, and K. Saenko, “Adashare: Learning what to share for efficient deep multi-task learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 8728–8740, 2020.
- [47] L. Duong, T. Cohn, S. Bird, and P. Cook, “Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (volume 2: short papers)*, 2015, pp. 845–850.
- [48] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, “Latent multi-task architecture learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4822–4829.
- [49] N. Pahari and K. Shimada, “Multi-task learning using bert with soft parameter sharing between layers,” in *2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS)*. IEEE, 2022, pp. 1–6.
- [50] J.-A. Désidéri, “Multiple-gradient descent algorithm (mgda) for multi-objective optimization,” *Comptes Rendus Mathematique*, vol. 350, no. 5-6, pp. 313–318, 2012.
- [51] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Traces and Emergence of Nonlinear Programming*. Springer, 2013, pp. 247–258.
- [52] J. Fliege and B. F. Svaiter, “Steepest descent methods for multicriteria optimization,” *Mathematical Methods of Operations Research*, vol. 51, pp. 479–494, 2000.
- [53] S. Schäffler, R. Schultz, and K. Weinzierl, “Stochastic method for the solution of unconstrained vector optimization problems,” *Journal of Optimization Theory and Applications*, vol. 114, pp. 209–222, 2002.
- [54] R. Mehta, S. Pal, V. Singh, and S. N. Ravi, “Deep unlearning via randomized conditionally independent hessians,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10422–10431.
- [55] A. Warnecke, L. Pirch, C. Wressnegger, and K. Rieck, “Machine unlearning of features and labels,” *arXiv preprint arXiv:2108.11577*, 2021.
- [56] A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh, “Remember what you want to forget: Algorithms for machine unlearning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 18075–18086, 2021.
- [57] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig, “The mutual information: Detecting and evaluating dependencies between variables,” *Bioinformatics*, vol. 18, no. suppl\_2, pp. S231–S240, 2002.
- [58] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [59] W. Huang and R. Y. Da Xu, “Gaussian process latent variable model factorization for context-aware recommender systems,” *Pattern Recognition Letters*, vol. 151, pp. 281–287, 2021.
- [60] T. Xiao and H. Shen, “Neural variational matrix factorization for collaborative filtering in recommendation systems,” *Applied Intelligence*, vol. 49, pp. 3558–3569, 2019.
- [61] L. Dos Santos, B. Piwowarski, and P. Gallinari, “Gaussian embeddings for collaborative filtering,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 1065–1068.
- [62] Z. Fan, Z. Liu, S. Wang, L. Zheng, and P. S. Yu, “Modeling sequences as distributions with uncertainty for sequential recommendation,” in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 2021, pp. 3019–3023.
- [63] S. He, K. Liu, G. Ji, and J. Zhao, “Learning to represent knowledge graphs with gaussian embedding,” in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, 2015, pp. 623–632.
- [64] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1235–1244.
- [65] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, “Deep matrix factorization models for recommender systems,” in *IJCAI*, vol. 17, 2017, pp. 3203–3209.
- [66] D. Jannach, Z. Karakaya, and F. Gedikli, “Accuracy improvements for multi-criteria recommender systems,” in *Proceedings of the 13th ACM Conference on Electronic Commerce*, 2012, pp. 674–689.
- [67] C. Gao, N. Li, T.-H. Lin, D. Lin, J. Zhang, Y. Li, and D. Jin, “Social recommendation with characterized regularization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 6, pp. 2921–2933, 2020.
- [68] F. Poirion, Q. Mercier, and J.-A. Désidéri, “Descent algorithm for nonsmooth stochastic multiobjective optimization,” *Computational Optimization and Applications*, vol. 68, pp. 317–331, 2017.
- [69] S. Peitz and M. Dellnitz, “Gradient-based multiobjective optimization with uncertainties,” in *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016 and the NEO Cities 2016 Workshop held on September 20-24, 2016 in Tlalnepantla, Mexico*. Springer, 2018, pp. 159–182.
- [70] M. Wan, J. Ni, R. Misra, and J. McAuley, “Addressing marketing bias in product recommendations,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 618–626.
- [71] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.
- [72] 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.
- [73] I. E. Olatunji, W. Nejdl, and M. Khosla, “Membership inference attack on graph neural networks,” in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2021, pp. 11–20.
- [74] S. Eidizadehkcheloo, B. Alipour Pijani, A. Imine, and M. Rusinowitch, “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*, 2021, pp. 338–354.
- [75] 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.
- [76] J. Foster, S. Schoepf, and A. Brintrup, “Fast machine unlearning without retraining through selective synaptic dampening,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 043–12 051.