

LEGO: A Lightweight and Efficient Multiple-Attribute Unlearning Framework for Recommender Systems

Fengyuan Yu
Zhejiang University
Hangzhou, China
fengyuanyu@zju.edu.cn

Junjie Fang
Hangzhou Dianzi University
Hangzhou, China
junjiefang@hdu.edu.cn

Yuyuan Li
Hangzhou Dianzi University
Hangzhou, China
y2li@hdu.edu.cn

Tao Wang
Midea Group
Foshan, China
tao.wang.seu@gmail.com

Xiaohua Feng
Zhejiang University
Hangzhou, China
fengxiaohua@zju.edu.cn

Chaochao Chen*
Zhejiang University
Hangzhou, China
zjuccc@zju.edu.cn

Abstract

With the growing demand for safeguarding sensitive user information in recommender systems, recommendation attribute unlearning is receiving increasing attention. Existing studies predominantly focus on single-attribute unlearning. However, privacy protection requirements in the real world often involve multiple sensitive attributes and are dynamic. Existing single-attribute unlearning methods cannot meet these real-world requirements due to i) **CH1**: the inability to handle multiple unlearning requests simultaneously, and ii) **CH2**: the lack of efficient adaptability to dynamic unlearning needs. To address these challenges, we propose LEGO, a lightweight and efficient multiple-attribute unlearning framework. Specifically, we divide the multiple-attribute unlearning process into two steps: i) *Embedding Calibration* removes information related to a specific attribute from user embedding, and ii) *Flexible Combination* combines these embeddings into a single embedding, protecting all sensitive attributes. We frame the unlearning process as a mutual information minimization problem, providing LEGO a theoretical guarantee of simultaneous unlearning, thereby addressing **CH1**. With the two-step framework, where *Embedding Calibration* can be performed in parallel and *Flexible Combination* is flexible and efficient, we address **CH2**. Extensive experiments on three real-world datasets across three representative recommendation models demonstrate the effectiveness and efficiency of our proposed framework. Our code and appendix are available at <https://github.com/anonymifish/lego-rec-multiple-attribute-unlearning>.

CCS Concepts

• **Information systems** → **Collaborative filtering**; • **Security and privacy** → **Human and societal aspects of security and privacy**.

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '25, Dublin, Ireland

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2035-2/2025/10
<https://doi.org/10.1145/3746027.3755604>

Keywords

Recommender System, Collaborative Filtering, Attribute Unlearning

ACM Reference Format:

Fengyuan Yu, Yuyuan Li, Xiaohua Feng, Junjie Fang, Tao Wang, and Chaochao Chen. 2025. LEGO: A Lightweight and Efficient Multiple-Attribute Unlearning Framework for Recommender Systems. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, October 27–31, 2025, Dublin, Ireland. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3746027.3755604>

1 Introduction

Modern recommender systems commonly use Collaborative Filtering (CF) algorithms to provide personalized recommendations [21, 27, 29, 35, 41, 42, 45, 49]. However, privacy concerns regarding personalized recommendations have increased, with increasing demand for protection against the misuse of sensitive user information. As a protective measure, the *Right to be Forgotten* requires recommendation platforms to allow users to withdraw individual data [3, 6, 34, 39]. *Recommendation unlearning* is an emerging approach for addressing these privacy concerns. One line of research, i.e., *input unlearning*, focuses on enabling the model to forget specific training data [26]. Another line of research, i.e., *attribute unlearning*, focuses on forgetting sensitive user attributes, which are not part of training data and cannot be unlearned through input unlearning [1, 12, 16, 25]. While input unlearning has been extensively studied, attribute unlearning remains comparatively underexplored. This paper aims to bridge this gap by focusing on attribute unlearning.

Most existing research on attribute unlearning can only handle single and static attributes [8, 13, 25]. However, in practice, unlearning requests usually involve multiple sensitive attributes and are dynamic: they may increase, decrease, or alter, as illustrated in Figure 1(a). The frequent changes in privacy protection requirements necessitate attribute unlearning to adapt flexibly and efficiently to these evolving demands.

In this paper, we identify that existing attribute unlearning methods cannot meet these requirements due to two key challenges: **CH1**: the inability to handle multiple unlearning requests simultaneously, and **CH2**: the lack of efficient adaptability to dynamic unlearning needs. Neither i) unlearning each attribute individually

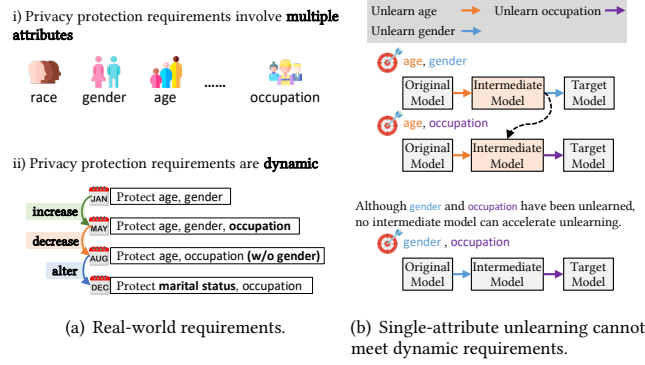


Figure 1: (a) Privacy protection requirements often involve multiple attributes and are dynamic: they may increase, decrease, and alter. (b) Single-attribute unlearning cannot meet dynamic privacy protection requirements. The dashed arrow indicates the storage of the intermediate model can accelerate sequential unlearning.

using single-attribute unlearning methods (i.e., sequential unlearning) nor ii) the only existing multiple-attribute unlearning method, AdvX [11], can address these two challenges. For **CH1**, the sequential unlearning approach may re-introduce previously unlearned attributes into the model while unlearning others, thereby degrading the effectiveness of unlearning. AdvX, which introduces an adversarial discriminator for each attribute, faces issues related to potential conflicts in optimization directions, which results in suboptimal unlearning effectiveness. For **CH2**, if the requirements change, the sequential unlearning approach needs to re-apply single-attribute unlearning to each sensitive attribute, even if many of them have already been unlearned. While saving intermediate models during unlearning alleviates this issue, it consumes considerable memory. Moreover, in many cases, as shown in Figure 1(b), even with intermediate models, the unlearning process cannot be accelerated. AdvX is also not adaptable to dynamic privacy protection requirements, as the training process must be re-executed each time the requirement changes.

To address the challenges above, we propose LEGO, a **Lightweight and Efficient multiple-attribute unlearninG Framework**. LEGO divides multiple-attribute unlearning process into two steps: *Embedding Calibration* and *Flexible Combination*. Firstly, embedding calibration removes information related to a specific attribute from user embedding. We achieve this by minimizing the mutual information between user embedding and the corresponding attribute. To preserve recommendation performance, we further introduce a parameter space constraint to ensure that, after calibration, embeddings do not deviate significantly from their original values. Secondly, flexible combination combines the unlearned embeddings into a single embedding, protecting all sensitive attributes that require protection through a weighted combination. Only the weights are optimized to ensure an efficient combination.

Our proposed two-step framework effectively addresses both challenges. Embedding calibration first unlearns a specific attribute,

and then flexible combination simultaneously unlearns all attributes by combining these embeddings. By leveraging the properties of mutual information and the parameter space constraint, we provide a theoretical guarantee for effective simultaneous unlearning of all attributes, addressing **CH1**. When a new requirement arises, embedding calibration can be performed in parallel to unlearn attributes not identified in previous requirements, and flexible combination can efficiently construct a new embedding that protects all sensitive attributes, thereby addressing **CH2**.

We summarize the main contributions of this paper as follows:

- We identify two key challenges of multiple-attribute unlearning in recommender systems (i.e., **CH1**: handling simultaneous unlearning requirements and **CH2**: adapting to dynamic needs). To tackle these challenges, we propose a multiple-attribute unlearning framework, named LEGO, which divides the multiple-attribute unlearning process into two steps: *Embedding Calibration* and *Flexible Combination*.
- To address **CH1**, *Embedding Calibration* first unlearns a specific attribute, and then *Flexible Combination* simultaneously unlearns all attributes by combining these embeddings with a theoretical guarantee of effectiveness.
- To address **CH2**, we propose a two-step framework, where *Embedding Calibration* can be performed in parallel to unlearn attributes, and *Flexible Combination* can efficiently construct a new embedding that protects all sensitive attributes.
- We conduct extensive experiments on three real-world datasets across three representative recommendation models. The results demonstrate that our method significantly outperforms existing baselines in terms of multiple-attribute unlearning effectiveness and efficiency.

2 Related Work

In this section, we review two major research lines of recommendation unlearning: traditional recommendation unlearning (input unlearning) and recommendation attribute unlearning.

2.1 Recommendation Unlearning

Machine unlearning aims to remove the influence of specific training data on a learned model (i.e., input unlearning) [32]. Existing machine unlearning methods can be categorized into two main approaches: i) Exact unlearning aims to remove the target data's influence as completely as if the model were retrained from scratch [4, 5]. ii) Approximate unlearning aims to estimate the influence of the target data and directly removes the influence through parameter manipulation [14, 15, 36, 43].

Following the partition-aggregation framework proposed by SISA (exact unlearning) [4], subsequent studies achieve exact unlearning tailored for recommender systems [6, 22, 23]. Approximate unlearning has also been explored in the context of recommendation [24, 48]. A benchmark has been proposed to comprehensively evaluate various recommendation unlearning methods [7].

2.2 Recommendation Attribute Unlearning

Due to the information extraction capabilities of recommender systems, sensitive attributes such as gender, race, and age of users can be encoded into user embeddings. However, since these attributes

are not explicitly represented in the training data, input unlearning (even exact unlearning or retraining from scratch) cannot effectively address attribute unlearning.

Existing research on recommendation attribute unlearning predominantly focuses on single-attribute unlearning. Ganhör et al. [13] is the first to address the attribute unlearning problem in recommender systems. They employ adversarial training during model training on a VAE-based recommendation model, MultVAE [28], to achieve attribute unlearning. Li et al. [25] explore post-training attribute unlearning by directly manipulating model parameters after the training process. This work focuses on the attributes with binary labels; in a later work, Chen et al. [8] extend the method to handle multiple-label attributes. The only work addressing multiple-attribute unlearning, AdvX [11], extends the approach of Adv [13] by introducing an additional attack discriminator for each attribute. However, these methods fail to meet real-world dynamic privacy protection requirements due to two key challenges: i) the inability to handle multiple unlearning requests simultaneously and ii) the lack of efficient adaptability to dynamic unlearning needs.

3 Preliminaries

3.1 Recommendation Model

Among recommendation models, CF is a well-established algorithm for generating personalized recommendations by analyzing collaborative information between users and items [37]. Let $\mathcal{U} = \{u_1, \dots, u_N\}$ and $\mathcal{V} = \{v_1, \dots, v_M\}$ denote the user and item set, respectively. In general, many existing CF approaches optimize users' latent representations, a.k.a., user embedding, during training to generate personalized recommendations. We denote user embedding of the model as $[\theta_1^\top, \dots, \theta_N^\top] = \mathbf{U} \in \mathbb{R}^{N \times d}$, where $\theta_i \in \mathbb{R}^d$ represents the transpose of the embedding of user u_i (d is the dimension of latent space). We denote the set of attributes as $\mathcal{A} = \{A_1, A_2, \dots\}$, where $A_i = \{c_1^i, \dots, c_{p_i}^i\}$ represents a sensitive attribute, and each c_j^i denotes a possible value of attribute A_i . We denote the value of attribute i for user u_j as $a_j^i, a_j^i \in A_i$.

3.2 Attacking Setting

Following the settings in the previous research [8, 25, 44, 47], the attack process in the attribute unlearning problem of recommender systems is also referred to as the Attribute Inference Attack (AIA) [1, 20], which is divided into three main stages: exposure, training, and attack. We adopt the assumption of a gray-box attack during the exposure stage, meaning that not all model parameters are exposed to the attacker; only the embeddings of users and some of their associated attribute information are revealed. In the training stage, it is assumed that the attacker trains the attack model on the shadow dataset [33], as assuming the attacker possesses the entire dataset is overly idealistic and impractical. In the context of multiple attribute unlearning, we assume that during the training stage, the attacker trains a separate attack model for each sensitive attribute. The attack process is framed as a classification task, where the attack model takes users' embedding as input and the attributes as labels. In the inference phase, the attacker utilizes their attack model to make predictions.

3.3 Mutual Information Estimation

In our framework, we employ Mutual Information (MI) minimization to achieve attribute unlearning because there is a natural link between MI and classification accuracy [10, 30, 31, 46]. MI $I(\mathbf{x}; \mathbf{y})$ is a fundamental measure of the dependence between two random variables, which represents the reduction in the uncertainty of $\mathbf{x}$ due to the knowledge of $\mathbf{y}$. If the MI between user embedding and the sensitive attribute is zero, the embedding carries no useful information for predicting the attribute. In this case, the optimal classifier would be one that randomly guesses the attribute based on its distribution in the sample.

Mathematically, the definition of MI between variables $\mathbf{x}$ and $\mathbf{y}$ is the relative entropy between the joint distribution and the product distribution $p(\mathbf{x})p(\mathbf{y})$:

$$I(\mathbf{x}; \mathbf{y}) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right]. \quad (1)$$

Calculating the exact value of MI is challenging, as it requires closed-form expression for the density functions and a tractable log-density ratio between the joint and marginal distributions [2, 40]. To estimate MI, previous work [9] derives CLUB, a contrastive log-ratio upper bound for MI. With the conditional distribution $p(\mathbf{y} | \mathbf{x})$, MI contrastive log-ratio upper bound is defined as:

$$I_{\text{CLUB}}(\mathbf{x}; \mathbf{y}) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log p(\mathbf{y} | \mathbf{x})] - \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\mathbf{y})} [\log p(\mathbf{y} | \mathbf{x})]. \quad (2)$$

When the conditional distributions $p(\mathbf{y} | \mathbf{x})$ or $p(\mathbf{x} | \mathbf{y})$ are unavailable, CLUB uses a variational distribution $q_\phi(\mathbf{y} | \mathbf{x})$ with parameter ϕ to approximate $p(\mathbf{y} | \mathbf{x})$. A variational CLUB term (vCLUB) is defined as follows:

$$I_{\text{vCLUB}}(\mathbf{x}; \mathbf{y}) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log q_\phi(\mathbf{y} | \mathbf{x})] - \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\mathbf{y})} [\log q_\phi(\mathbf{y} | \mathbf{x})]. \quad (3)$$

vCLUB no longer guarantees an upper bound of $I(\mathbf{x}; \mathbf{y})$ using the variational approximation $q_\phi(\mathbf{y} | \mathbf{x})$. However, with a good variational approximation $q_\phi(\mathbf{y} | \mathbf{x})$, vCLUB can still hold an upper bound on MI. Denote $q_\phi(\mathbf{x}, \mathbf{y}) = q_\phi(\mathbf{y} | \mathbf{x})p(\mathbf{x})$, CLUB proves that vCLUB remains a MI upper bound if

$$KL(p(\mathbf{x}, \mathbf{y}) || q_\phi(\mathbf{x}, \mathbf{y})) \leq KL(p(\mathbf{x})p(\mathbf{y}) || q_\phi(\mathbf{x}, \mathbf{y})). \quad (4)$$

This inequality suggests that vCLUB remains a MI upper bound if the variational joint distribution $q_\phi(\mathbf{x}, \mathbf{y})$ is "closer" to $p(\mathbf{x}, \mathbf{y})$ than to $p(\mathbf{x})p(\mathbf{y})$. Therefore, minimizing $KL(p(\mathbf{x}, \mathbf{y}) || q_\phi(\mathbf{x}, \mathbf{y}))$ helps satisfy the condition for vCLUB to remain an upper bound on MI. This KL divergence can be minimized by maximizing the log-likelihood of $q_\phi(\mathbf{y} | \mathbf{x})$, because of the following equation:

$$\begin{aligned} & \min_{\phi} KL(p(\mathbf{x}, \mathbf{y}) || q_\phi(\mathbf{x}, \mathbf{y})) \\ &= \min_{\phi} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log(p(\mathbf{y} | \mathbf{x})p(\mathbf{x})) - \log(q_\phi(\mathbf{y} | \mathbf{x})p(\mathbf{x}))] \\ &= \min_{\phi} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log p(\mathbf{y} | \mathbf{x})] - \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log q_\phi(\mathbf{y} | \mathbf{x})]. \end{aligned} \quad (5)$$

The first term of Eq. (5) is independent of the parameter ϕ . Therefore, this minimization problem is equivalent to maximizing the second term. Thus, given samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^B$, maximizing the

log-likelihood function

$$\mathcal{L}(\phi) = \frac{1}{B} \sum_{i=1}^B \log q_{\phi}(\mathbf{y}_i | \mathbf{x}_i), \quad (6)$$

which leads to a better variational approximation.

In general, MI minimization aims to reduce the correlation between two variables $\mathbf{x}$ and $\mathbf{y}$ by selecting an optimal parameter σ if the joint variational distribution $p_{\sigma}(\mathbf{x}, \mathbf{y})$. With vCLUB, MI can be minimized through an alternative optimization approach. In each training iteration, vCLUB first optimizes ϕ by maximizing the log-likelihood $\mathcal{L}(\phi)$ with sampled data points to obtain a better variational approximation. Then, it estimates the upper bound of MI as follows:

$$\begin{aligned} \hat{\mathcal{I}}_{\text{vCLUB}} &= \frac{1}{B^2} \sum_{i=1}^B \sum_{j=1}^B [\log q_{\phi}(\mathbf{y}_i | \mathbf{x}_i) - \log q_{\phi}(\mathbf{y}_j | \mathbf{x}_i)] \\ &= \frac{1}{B} \sum_{i=1}^B \left[\log q_{\phi}(\mathbf{y}_i | \mathbf{x}_i) - \frac{1}{B} \sum_{j=1}^B \log q_{\phi}(\mathbf{y}_j | \mathbf{x}_i) \right]. \end{aligned} \quad (7)$$

with samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^B$. After that, the gradient descent is used to optimize σ .

4 Methodology

In this section, we first introduce our proposed multiple-attribute unlearning framework LEGO, which decomposes the task of multiple-attribute unlearning into two steps: *Embedding Calibration* and *Flexible Combination*. Next, we provide a detailed explanation of these two steps.

4.1 Overview of LEGO

To meet the dynamic privacy protection requirements in multiple-attribute unlearning in recommender systems, LEGO performs parallelizable single-attribute unlearning and then combines the unlearned embeddings based on the specific privacy protection requirements. Figure 2 presents an overview of our proposed LEGO. After training the recommender system, the user embedding $\mathbf{U}_0$ of the CF model encode sensitive user information, potentially exposing them to adversaries. We denote the sensitive attributes set that needs to be protected under the new privacy protection requirement as $\mathcal{A}_r = \{\mathbf{A}_1, \dots, \mathbf{A}_k\}$.

Embedding calibration. The embedding calibration step modifies the user embedding $\mathbf{U}_0$ to unlearn a single sensitive attribute $\mathbf{A}_t$, thereby preventing adversaries from inferring sensitive user information from the embedding while preserving recommendation performance. After embedding calibration, we obtain k distinct embeddings $\mathbf{U}_1^*, \dots, \mathbf{U}_k^*$, each unlearning the corresponding sensitive attribute $\mathbf{A}_1, \dots, \mathbf{A}_k$, respectively. Although these embeddings protect the unlearned attributes, they may still leak other sensitive user attributes.

Flexible combination. In this step, embeddings $\mathbf{U}_i^*, i = 1, \dots, k$ are combined to form $\mathbf{U}^* = \alpha_1 \cdot \mathbf{U}_1^* + \dots + \alpha_k \cdot \mathbf{U}_k^*$. The combination step optimizes only the combination weights $\alpha = [\alpha_1, \dots, \alpha_k]$, ensuring both flexibility and efficiency. After the flexible combination, the embedding $\mathbf{U}^*$ protects all the privacy information that requires

protection. The combined embedding $\mathbf{U}^*$ then replaces the original user embedding $\mathbf{U}_0$.

LEGO can meet dynamic requirements. When a new privacy protection requirement arises: i) If the new requirement includes new attributes, the embedding calibration step in LEGO can be performed in parallel. ii) If no new attributes exist, the embedding calibration does not need to be performed again, as embeddings that have already unlearned a specific attribute can be leveraged. iii) LEGO can swiftly construct a new embedding by combining embeddings that have unlearned a specific attribute, thus meeting the new privacy protection requirement.

LEGO can unlearn multiple attributes simultaneously. LEGO provides a theoretical guarantee for simultaneously protecting multiple sensitive attributes. In embedding calibration, we define our unlearning objective as an MI minimization optimization problem with a parameter space constraint. We minimize MI to prevent adversaries from inferring sensitive user information, while the parameter space constraint preserves recommendation performance. In flexible combination, we optimize the combination weights by minimizing the MI between the combined embedding and sensitive attributes. There are several other methods to prevent adversaries from inferring sensitive user information. Two of the most widely used approaches are distribution alignment (employed in D2DFR) and adversarial training (used in AdvX). However, these two objectives are not suitable for the two-step approach of LEGO. The distribution alignment method requires computing the centers of distributions for each attribute. However, these distributions may differ significantly from one another, thereby combining these embeddings could considerably degrade the recommendation performance of the model. Since the adversarial training method adversaries different objectives in the first step and is uninterpretable, we cannot guarantee that the combined embedding will effectively protect all sensitive attributes simultaneously. In contrast, the MI minimization objective ensures that the two-step approach's result does not deviate significantly from the optimal solution.

DEFINITION 1. Let $\mathbf{U}_0, \mathbf{U}_i^1, \mathbf{U}_i^2 \in \mathbb{R}^{N \times d}$ denotes user embeddings,

$$P_1 = \min_{\alpha^1 \in \Delta^{k-1}} \sum_{t=1}^k I \left(\sum_{i=1}^k \alpha_i^1 \cdot \mathbf{U}_i^1; \mathbf{A}_t \right),$$

$$P_2 = \min_{\alpha^2 \in \Delta^{k-1}, \mathbf{U}_i^2 \in \mathcal{B}_{\epsilon}(\mathbf{U}_0)} \sum_{t=1}^k I \left(\sum_{i=1}^k \alpha_i^2 \cdot \mathbf{U}_i^2; \mathbf{A}_t \right),$$

where Δ^{k-1} represents the $(k-1)$ -dimensional standard simplex, $\mathcal{B}_{\epsilon}(\mathbf{U}_0)$ represents the Euclidean ball of radius ϵ centered at $\mathbf{U}_0$.

THEOREM 1. Assume that $\mathbf{U}_i^1 = \arg \min_{\mathbf{U}_i \in \mathcal{B}_{\epsilon}(\mathbf{U}_0)} I(\mathbf{U}_i, \mathbf{A}_i)$ are constant matrices, and $\|\mathbf{U}_0\|_2 \leq C$ for some constant $C > 0$. Then, we have the bound $|P_1 - P_2| \leq 2kL(C + 2\epsilon)$, where L is the Lipschitz constant for MI.

PROOF. The proof can be found in Appendix B. $\square$

Theorem 1 shows that the gap between P_1 , the result of LEGO, and P_2 , the result of an end-to-end version of LEGO that unlearns multiple attributes simultaneously, is bounded by $2kL(C + 2\epsilon)$. This provides a theoretical guarantee that a linear combination of user

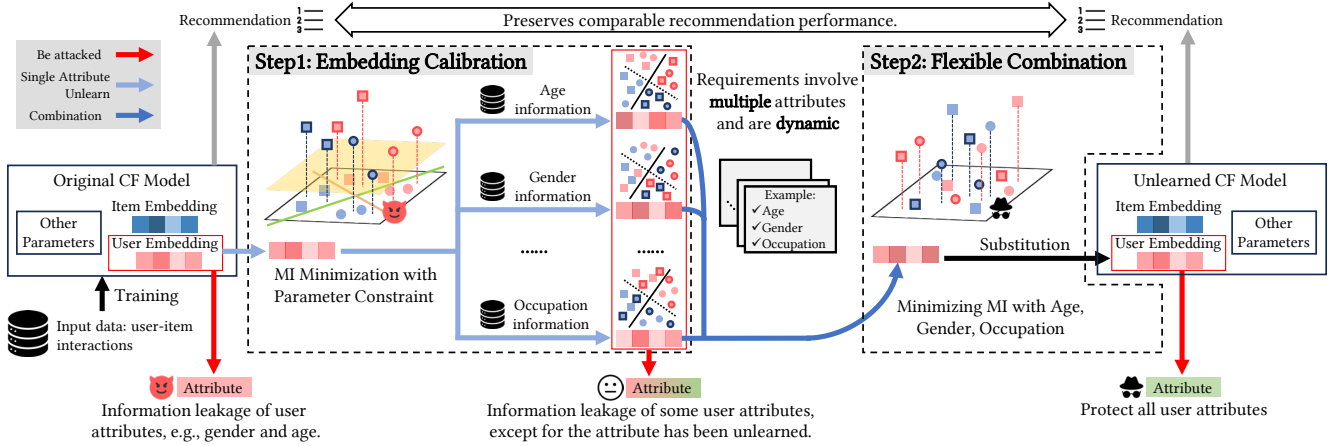


Figure 2: An overview of LEGO. Our proposed LEGO splits multiple-attribute unlearning into two steps: embedding calibration and flexible combination. To illustrate the goal of each step, we provide a sketch of the embedding distribution. In the sketch, the shape, color, and border of the data points represent age, gender, and occupation information, respectively. The lines and plane represent the decision boundaries of the classifier.

embeddings with one specific attribute information removed can lead to a user embedding in which all sensitive attribute information is unlearned, demonstrating that LEGO can protect multiple sensitive attributes simultaneously.

4.2 Embedding Calibration

In the embedding calibration step, we focus on two objectives in attribute unlearning: to protect a single sensitive attribute while preserving the recommendation performance.

Protecting a single sensitive attribute. To prevent the sensitive attribute A_t from being successfully classified by the attack model, we minimize the MI between the user embedding U_0 and A_t . This can be formalized as follows:

$$U_t^* = \arg \min_{U_t} I(U_t; A_t). \quad (8)$$

With a suitable variational distribution, vCLUB provides an upper bound for MI. Thus, by minimizing the vCLUB, we can effectively minimize the MI:

$$U_t^* = \arg \min_{U_t} I_{\text{vCLUB}}(U_t; A_t). \quad (9)$$

Specifically, we use a neural network parameterized by ϕ to model the variational distribution $q_\phi(A_t | u_t)$. With vCLUB, we minimize $I(U_t; A_t)$ by minimizing the following objectives through alternating optimization of ϕ and U_t , as detailed in Section 3:

$$\begin{aligned} \phi &= \arg \max_{\phi} \mathbb{E}_{p(U_t, A_t)} \mathcal{L}(\phi), \\ U_t^* &= \arg \min_{U_t} \mathbb{E}_{p(U_t, A_t)} \hat{I}_{\text{vCLUB}}. \end{aligned} \quad (10)$$

Preserve recommendation performance. To preserve the recommendation performance, we apply a parameter space constraint $U_t \in \mathcal{B}_\epsilon(U_0)$ to ensure that, after calibration, the embeddings do not deviate significantly from the original ones, where ϵ is a hyperparameter that controls the maximum deviation between the

calibrated embedding U_t and the original embedding U_0 . Combining the optimization problem described in Eq. (10) with the parameter space constraint, we obtain a constraint optimization problem. Since the Euclidean projection operator $\text{proj}(\cdot)$ for the constraint has a closed-form solution, we add a projection operation after the alternative optimization algorithm to solve this constrained optimization problem. Specifically, after updating the embeddings using gradient descent, we apply a projection operation:

$$U_t = \begin{cases} U_t, & \text{if } \|U_t - U_0\|_2 \leq \epsilon, \\ \text{proj}(U_t) = U_0 + \frac{\epsilon}{\|U_t - U_0\|_2} (U_t - U_0), & \text{otherwise.} \end{cases} \quad (11)$$

4.3 Flexible Combination

In the flexible combination step, we combine the embeddings to obtain the combined embedding $U^* = U(\alpha) = \sum_{i=1}^k \alpha_i U_i^*$, where $\alpha = [\alpha_1, \dots, \alpha_k] \in \mathbb{R}^k$. To ensure that the combined embedding protects all sensitive attributes, we minimize MI between the combined embedding and all sensitive information:

$$\begin{aligned} \min_{\alpha} \sum_{i=1}^k I(U(\alpha); A_i), \\ \text{s.t. } \alpha_i > 0, i = 1, \dots, k, \quad \|\alpha\|_1 = 1. \end{aligned} \quad (12)$$

The constraint in this optimization problem prevents the trivial solution where $\alpha = \mathbf{0}$ and ensures the normalization of the weights. Similarly, we employ vCLUB and an alternative optimization algorithm to minimize MI. For each attribute A_t , a neural network parameterized by ϕ_t is utilized to model the variational distribution $q_{\phi_t}(A_t | U_t)$. To meet the constraint, we also use the projected gradient descent, where the projection operator is

$$\text{proj}(\alpha) = \text{softmax}(\alpha) = \left[\frac{\exp(\alpha_1)}{\sum_{j=1}^k \exp(\alpha_j)}, \dots, \frac{\exp(\alpha_k)}{\sum_{j=1}^k \exp(\alpha_j)} \right]. \quad (13)$$

By using softmax, we ensure that the projection operation adheres to the constraints while maintaining the stability and efficiency of the optimization process.

We summarize the complete procedure of LEGO in Algorithm 1.

Algorithm 1 LEGO

```

1: Input: User embedding  $U_0$ , user sensitive attributes  $\mathcal{A}_r$ , training epoch  $E_1$ ,  $E_2$ , batch size  $B$ , update step size  $\eta$ , parameter space constraint threshold  $\epsilon$ .
2: for  $t = 1$  to  $k$  do
3:   Initial:  $U_t^0 \leftarrow U_0$ , randomly initialize the parameters of  $\phi$ .
4:   for  $e = 0$  to  $E_1 - 1$  do
5:     Sample  $\{(\theta_{b_i}^\top, a_{b_i}^t)\}_{i=1}^B$  from  $p(U_t^0, A_t)$ .
6:     Update  $\phi$  by maximizing  $\mathcal{L}(\phi)$  as defined in Eq. (6).
7:     Compute MI estimation  $\hat{I}_{\text{vCLUB}}$  as defined in Eq. (7).
8:      $U_t^{e+1} \leftarrow U_t^e - \eta \cdot \nabla_{U_t^e} \hat{I}_{\text{vCLUB}}$ .
9:     Project  $U_t^{e+1}$  as defined in Eq. (11).
10:   end for
11:    $U_t^* \leftarrow U_t^{E_1}$ .
12: end for
13: Initial:  $\alpha_0 \leftarrow [\frac{1}{k}, \dots, \frac{1}{k}]$ , randomly initialize the parameters of  $\phi_1, \dots, \phi_k$ .
14: for  $e = 0$  to  $E_2 - 1$  do
15:   Sample  $\{(\theta_{b_i}^\top, a_{b_i}^1, \dots, a_{b_i}^k)\}_{i=1}^B$  from  $p(U(\alpha_0), A_1, \dots, A_k)$ .
16:   Update  $\phi_1, \dots, \phi_k$  by maximizing  $\mathcal{L}(\phi)$ .
17:   Compute MI estimation  $\hat{I}_{\text{vCLUB}}$ .
18:    $\alpha_{e+1} \leftarrow \alpha_e - \eta \cdot \nabla_{\alpha_e} \hat{I}_{\text{vCLUB}}$ .
19:   Project  $\alpha_{e+1}$  as defined in Eq. (13).
20: end for
21: return new user embedding  $U(\alpha_{E_2})$ .

```

5 Experiments

To comprehensively evaluate our proposed method, we conduct experiments on three benchmark datasets and three representative recommendation models. Specifically, we aim to answer the following Research Questions (RQs):

- **RQ1:** Can our method effectively unlearn multiple attributes simultaneously?
- **RQ2:** Does our method preserve the recommendation performance after unlearning?
- **RQ3:** Can our method meet dynamic privacy protection requirements? In other words, how efficient is our proposed approach?
- **RQ4:** What is the impact of key hyperparameters on both unlearning and recommendation performance in our proposed method?
- **RQ5:** What roles do the embedding calibration step and the flexible combination step play in our proposed LEGO?

In the Appendix C, we provide additional experimental results for further analysis.

5.1 Experimental Settings

Datasets. We conduct experiments on three publicly available real-world datasets, each containing user-item interaction data and user attribute information (e.g., age and gender).

- **MovieLens 100K (ML-100K)**¹: The MovieLens dataset is widely recognized as one of the most extensively used resources for recommender system research [17]. It contains user ratings for movies, as well as various user attributes such as gender, age, and occupation. Specifically, ML-100K subset includes 100,000 ratings from 1000 users on 1700 movies.
- **MovieLens 1M (ML-1M)**²: A version of MovieLens dataset that has 1 million ratings from 6000 users on 4000 movies.
- **KuaiSAR**³: KusiSAR is a large-scale, real-world dataset collected from Kuaishou, a leading short-video app in China with over 350 million daily active users [38]. For users, this dataset included two encrypted features for each user. In our experiments, we utilize KuaiSAR-small.

We provide details of dataset pre-processing and the statistics of the above datasets after pre-processing in Appendix A.

Recommendation Models. We validate the effectiveness of our proposed method across three representative and widely recognized recommendation models.

- **NCF:** Neural Collaborative Filtering (NCF) is a foundational collaborative filtering model that employs neural network architectures [19].
- **LightGCN:** Light Graph Convolution Network (LightGCN) is a State-Of-The-Art (SOTA) collaborative filtering model that optimizes recommendation performance through a simplified graph convolutional network design [18].
- **MultVAE:** MultVAE learns to recommend items by decoding the variational encoding of user interaction vectors and has shown superior performance compared to various deep neural network approaches [28].

Unlearning Methods. We compare our proposed method, LEGO, with the original model and three attribute unlearning methods.

- **Original:** This is the original model without attribute unlearning.
- **DP** [50]: This method protects user attributes by introducing noise perturbation to the user embedding during the model pre-diction process.
- **D2DFR** [8]: This method represents the latest SOTA single-attribute unlearning method, which is achieved through distribution alignment. To extend this method to multi-attribute unlearning, we adopt a sequential forgetting approach, where after forgetting one attribute, the method continues to forget the next attribute until all attributes have been forgotten.
- **AdvX** [11]: This is the only multiple-attribute unlearning method, which employs adversarial training to achieve attribute unlearning. While the original method is specifically designed for MultVAE, we extend it to other recommendation models.

We provide details of evaluation metrics, parameter settings, and hardware information in Appendix A.

5.2 Results and Discussions

5.2.1 Unlearning Performance (RQ1). The primary goal of attribute unlearning is to remove sensitive information from the recommendation model, preventing adversaries from inferring sensitive

¹<https://grouplens.org/datasets/movielens/100k/>

²<https://grouplens.org/datasets/movielens/1m/>

³<https://kuaisar.github.io/>

Table 1: Results of recommendation performance (HR@10 and NDCG@10) and unlearning performance (i.e., the performance of attackers: BAcc and F1). Except for Original, the best results are highlighted in bold. We run all models 10 times and report the average results and standard deviation. Results are expressed as percentages (%).

Dataset	Attributes	Method	NCF				LightGCN				MultVAE			
			HR@10 ↑	NDCG@10 ↑	BAcc ↓	F1 ↓	HR@10 ↑	NDCG@10 ↑	BAcc ↓	F1 ↓	HR@10 ↑	NDCG@10 ↑	BAcc ↓	F1 ↓
ML-100K	Gender	Original	15.67±0.32	8.37±0.21	65.61±0.74	66.62±0.60	16.07±0.81	8.60±0.17	62.96±1.73	63.43±1.35	16.40±1.01	8.63±0.32	65.64±1.69	65.72±1.11
		DP	5.20±0.28	2.40±0.28	53.61±0.70	52.50±0.46	8.03±1.26	4.17±0.45	61.62±1.16	62.15±1.03	14.00±0.85	6.80±0.14	63.11±2.40	63.05±2.68
		D2DFR	16.07±0.15	8.23±0.06	47.42±5.57	52.13±1.36	16.23±0.15	8.53±0.06	55.71±1.36	54.22±1.03	16.27±0.76	8.30±0.26	59.63±3.18	58.97±0.68
		AdvX	16.53±1.02	8.63±0.59	55.05±7.81	60.19±3.42	12.30±0.17	6.10±0.17	54.80±18.30	60.21±5.57	10.53±0.40	5.37±0.15	44.15±2.42	48.05±1.20
		LEGO	15.20±0.20	7.87±0.12	48.18±7.38	47.27±2.10	16.00±0.10	8.33±0.06	49.09±2.51	48.84±1.00	15.83±0.59	7.87±0.15	49.30±3.35	49.98±0.46
	Gender, Age	Original	15.67±0.32	8.37±0.21	62.75±0.46	63.26±0.44	16.07±0.81	8.60±0.17	60.30±0.69	60.54±0.49	16.40±1.01	8.63±0.32	62.21±1.21	62.25
		DP	5.20±0.28	2.40±0.28	44.78±0.57	44.22±0.72	8.03±1.26	4.17±0.45	55.97±0.52	56.24±0.47	14.00±0.85	6.80±0.14	58.38±1.06	58.36±1.18
		D2DFR	15.63±0.15	8.30±0.10	48.57±0.47	49.84±0.70	16.23±0.12	8.50±0.10	52.65±1.20	51.69±0.36	16.17±0.06	8.33±0.23	54.72±0.17	54.59±0.49
		AdvX	16.23±0.55	8.30±0.17	54.19±1.07	54.42±1.68	12.53±0.46	6.33±0.12	48.20±2.06	49.32±0.29	7.67±0.07	3.80±0.06	50.04±1.58	50.41±0.17
		LEGO	15.70±0.00	8.30±0.00	46.74±0.73	46.54±1.08	16.50±0.00	8.40±0.00	36.11±0.84	35.57±0.63	16.00±0.78	8.07±0.42	38.97±0.29	39.81±1.04
	Gender, Age, Occupation	Original	15.67±0.32	8.37±0.21	42.67±0.11	42.97±0.03	16.07±0.81	8.60±0.17	41.78±0.78	42.16±0.63	16.40±1.01	8.63±0.32	42.83±0.92	42.93±0.58
		DP	5.20±0.28	2.40±0.28	30.76±0.23	30.27±0.40	8.03±1.26	4.17±0.45	38.98±0.37	39.51±0.39	14.00±0.85	6.80±0.14	40.73±0.71	40.28±0.71
		D2DFR	15.57±0.12	8.23±0.06	33.96±0.66	33.65±0.40	16.37±0.06	8.50±0.00	35.01±0.55	35.30±0.81	15.83±0.40	8.27±0.38	37.69±0.79	38.15±0.39
		AdvX	15.83±0.58	8.40±0.35	33.99±3.60	35.71±1.34	11.97±0.31	6.07±0.06	34.37±7.63	41.67±0.05	6.25±2.90	3.50±0.99	28.01±0.93	27.83±1.49
		LEGO	16.00±0.00	8.03±0.06	32.30±0.28	33.22±0.22	16.60±0.00	8.60±0.00	28.19±0.99	27.59±0.98	16.33±0.50	8.40±0.20	30.85±1.67	30.88±0.98
ML-1M	Gender	Original	8.20±0.26	4.10±0.17	76.12±0.20	75.60±0.13	9.10±0.10	4.60±0.10	71.28±0.04	70.41±0.70	9.20±0.26	4.40±0.20	72.10±1.26	71.18±1.08
		DP	2.20±0.10	0.97±0.06	55.30±1.01	55.32±0.57	3.13±0.06	1.53±0.06	66.26±0.21	65.98±0.24	7.67±0.26	3.70±0.00	68.46±0.58	68.06±0.58
		D2DFR	8.50±0.00	4.20±0.00	51.28±6.52	50.64±0.17	5.90±0.00	3.00±0.00	49.03±5.97	55.41±1.64	8.03±0.12	4.03±0.06	47.86±7.44	54.82±2.38
		AdvX	8.60±0.26	4.27±0.15	55.21±3.10	61.00±2.87	4.57±0.06	2.20±0.00	64.17±6.76	68.49±2.23	4.67±0.15	2.37±0.06	63.65±1.93	64.94±0.45
		LEGO	8.07±0.06	3.97±0.06	47.48±0.63	44.83±0.23	8.70±0.00	4.50±0.00	51.20±2.21	49.56±0.18	8.97±0.46	4.40±0.10	53.21±0.42	52.17±0.80
	Gender, Age	Original	8.20±0.26	4.10±0.17	71.88±0.13	71.62±0.16	9.10±0.10	4.60±0.10	67.45±0.26	67.01±0.46	9.20±0.26	4.40±0.20	68.10±0.38	67.64±0.23
		DP	2.20±0.10	0.97±0.06	47.76±0.61	47.77±0.48	3.13±0.06	1.53±0.06	60.27±0.49	60.13±0.54	7.67±0.23	3.70±0.00	64.50±0.68	64.30±0.45
		D2DFR	8.40±0.00	4.17±0.06	61.09±0.31	60.98±0.16	5.90±0.00	3.00±0.00	44.66±6.25	54.31±1.42	7.93±0.15	3.97±0.06	50.22±2.41	53.90±1.15
		AdvX	8.33±0.31	4.10±0.10	46.65±6.92	49.62±1.44	4.63±0.06	2.30±0.00	61.04±0.74	61.26±0.76	4.43±0.23	2.13±0.06	40.98±9.58	41.74±0.17
		LEGO	8.10±0.00	4.00±0.00	56.34±0.20	55.68±0.29	8.60±0.00	4.40±0.00	50.80±0.44	50.13±0.19	8.80±0.00	4.27±0.06	53.09±0.34	52.29±0.42
	Gender, Age, Occupation	Original	8.20±0.26	4.10±0.17	51.25±0.45	51.07±0.50	9.10±0.10	4.60±0.10	48.32±0.36	48.07±0.55	9.20±0.26	4.40±0.20	49.41±0.53	49.17±0.47
		DP	2.20±0.10	0.97±0.06	33.46±0.50	33.43±0.38	3.13±0.06	1.53±0.06	43.26±0.49	43.11±0.51	7.67±0.23	3.70±0.00	45.67±0.46	45.54±0.34
		D2DFR	8.40±0.00	4.10±0.00	42.20±0.14	41.89±0.17	6.00±0.00	3.00±0.00	43.91±0.16	43.48±0.25	7.97±0.15	4.00±0.00	45.39±0.54	45.56±0.39
		AdvX	8.30±0.36	4.00±0.20	44.32±0.60	44.33±0.50	4.50±0.00	2.20±0.00	39.66±0.32	39.77±0.38	4.57±0.21	2.30±0.20	26.28±5.59	29.26±0.06
		LEGO	8.23±0.06	4.00±0.00	41.55±0.10	41.02±0.02	8.80±0.00	4.50±0.00	38.97±0.28	38.32±0.63	9.07±0.42	4.40±0.17	39.55±0.45	39.12±0.31
KuaiSAR	Feat1	Original	1.87±0.12	0.93±0.06	14.87±0.66	14.88±0.66	3.43±0.06	1.80±0.00	13.30±1.40	13.30±1.41	3.30±0.00	1.67±0.06	13.59±1.33	13.59±1.32
		DP	1.33±0.06	0.70±0.00	13.45±1.31	13.45±1.32	1.27±0.06	0.67±0.06	12.37±0.85	12.37±0.84	2.73±0.06	1.40±0.00	13.52±0.15	13.51±0.14
		D2DFR	2.00±0.00	1.00±0.00	14.08±1.18	14.08±1.20	3.53±0.06	1.80±0.00	12.98±0.70	12.97±0.70	3.30±0.00	1.63±0.06	13.26±0.65	13.26±0.65
		AdvX	1.53±0.25	0.73±0.15	14.69±0.84	14.75±0.84	2.47±0.06	1.27±0.06	12.87±0.53	12.87±0.55	1.33±0.25	0.70±0.17	12.54±0.04	12.50±0.00
		LEGO	2.00±0.00	1.00±0.00	14.41±0.04	14.43±0.04	3.50±0.00	1.80±0.00	11.57±0.16	11.56±0.17	3.30±0.10	1.67±0.06	12.59±0.72	12.58±0.73
	Feat1, Feat2	Original	1.87±0.12	0.93±0.06	24.01±0.66	24.01±0.65	3.43±0.06	1.80±0.00	23.67±0.37	23.68±0.36	3.30±0.00	1.67±0.06	24.01±1.09	24.01±1.08
		DP	1.33±0.06	0.70±0.00	21.77±1.40	21.78±1.40	1.27±0.06	0.67±0.06	21.49±0.82	21.48±0.82	2.73±0.06	1.40±0.00	24.03±0.30	24.02±0.29
		D2DFR	2.00±0.00	1.00±0.00	22.16±0.98	22.16±0.98	3.50±0.00	1.80±0.00	19.33±0.76	19.33±0.77	3.30±0.10	1.67±0.06	23.78±0.79	23.76±0.79
		AdvX	1.83±0.15	0.87±0.06	24.83±0.73	24.88±0.71	1.93±0.06	1.03±0.06	23.21±0.57	23.25±0.50	1.20±0.70	0.57±0.32	23.78±0.50	23.75±0.47
		LEGO	2.00±0.00	1.00±0.00	23.99±0.44	23.67±0.43	3.50±0.00	1.80±0.00	18.47±0.12	18.73±0.11	2.23±0.06	1.10±0.00	22.97±0.78	22.97±0.78

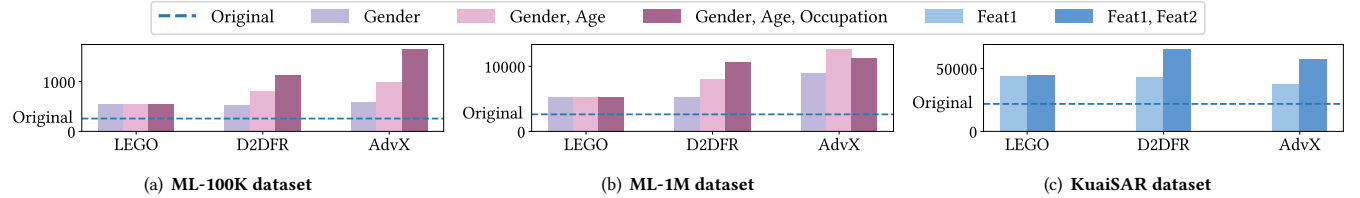


Figure 3: Results of efficiency in adapting to dynamic requirements. We present the running time of compared methods on NCF model across three datasets. We run all models 10 times and report the average results in seconds (s). The dashed line represents the training time of the original recommendation model.

user attributes. To comprehensively evaluate the unlearning performance of LEGO, we report two metrics, F1 score and BAcc, in Table 1. DP, D2DFR, AdvX, and LEGO reduce the BAcc by an average of 12.77%, 20.84%, 18.37%, and 24.31%, respectively, compared to the original model. These results demonstrate that LEGO effectively removes sensitive information from the recommendation model. Specifically, D2DFR reduces the BAcc on one, two, and three attributes by an average of 25.08%, 22.58%, and 13.79%, respectively,

indicating that D2DFR is less effective at removing multiple attributes simultaneously. AdvX reduces the BAcc on MultVAE by an average of 24.75%, and on NCF and LightGCN by 19.61% and 13.04%, respectively, highlighting that AdvX lacks of generalizability across different recommendation models.

5.2.2 Recommendation Performance (RQ2). While unlearning sensitive user attributes, the impact on recommendation performance should be minimized to ensure the utility of the recommender

system. We use HR and NDCG to evaluate recommendation performance after unlearning, truncating the rank list at 10 for both metrics. As shown in Table 1, unlearning methods do affect recommendation performance to varying degrees. DP, D2DFR, AdvX, and LEGO reduce NDCG@10 by 48.17%, 5.64%, 30.30%, and 3.43%, respectively, on average. The results demonstrate that LEGO effectively preserves recommendation performance. Specifically, D2DFR decreases NDCG@10 on one, two, and three attributes by an average of 5.52%, 5.48%, and 6.72%, respectively. In contrast, LEGO reduces NDCG@10 on one, two, and three attributes by an average of 3.92%, 4.08%, and 1.99%, respectively. This indicates that while the sequential unlearning methods degrade model recommendation performance, LEGO does not have the same effect.

5.2.3 Efficiency in Adapting to Dynamic Requirements (RQ3). We evaluate the efficiency of these unlearning methods in adapting to dynamic privacy protection requirements based on their running time. Since the recommendation model does not affect the overall trend, We conduct experiments on all three datasets using the NCF model, with the total running time reported in seconds. For better comparison, we indicate the original recommendation model training time with a dashed line. DP only adds noise to the inference process, so its running time is the same as the original training time. As shown in Figure 3, we observe that compared to AdvX, our proposed LEGO significantly reduces running time across all three datasets. This is because AdvX employs a time-consuming adversarial training approach during its training process to achieve attribute unlearning. D2DFR’s running time is directly proportional to the number of attributes. Specifically, our proposed LEGO achieves nearly the same efficiency in multiple attributes unlearning as in single-attribute unlearning. These results demonstrate that LEGO can effectively meet dynamic privacy protection requirements.

5.2.4 Parameter Sensitivity (RQ4). We investigate the hyperparameter ϵ , which controls the maximum deviation of the unlearned embedding from the original embedding. This parameter trades off unlearning effectiveness and recommendation performance. Since the total norm is related to the number of users N , we control ϵ/N to be 0, 0.1, 0.2, 0.3, 0.4, and ∞ (without any constraint). In the experiment, we report the results of unlearning all user attributes recorded in the dataset, while fixing the number of training iterations at 2000 to ensure convergence. As shown in Figure 4, particularly in Figure 4(b), as ϵ/N increases, both NDCG@10 and BAcc decrease. This occurs because a looser constraint allows for more extensive calibration of the embedding to improve attribute unlearning effectiveness, but it degrades recommendation performance. In Figure 4(b), LightGCN’s NDCG@10 increases as the ϵ/N increases. This is because our method may unintentionally reduce negative biases, potentially leading to unexpected improvements in recommendation performance. This phenomenon has been consistently observed in prior work [8, 25]. As shown in Figure 4(a), in the ML-100K dataset, recommendation performance remains robust to changes in ϵ/N , as it is a relatively small dataset. Due to space constraints, the full set of hyperparameter sensitivity results are provided in the Appendix C.

Table 2: Results of ablation studies on two steps (Step 1: D2DFR-FC, Step 2: EC-AC).

	HR@10 (%) $\uparrow$	NDCG@10 (%) $\uparrow$	BAcc (%) $\downarrow$	F1 (%) $\downarrow$
D2DFR-FC	7.56 $\pm$ 0.04	3.24 $\pm$ 0.03	47.65 $\pm$ 1.00	47.39 $\pm$ 0.80
EC-AC	8.19 $\pm$ 0.06	3.96 $\pm$ 0.01	45.57 $\pm$ 0.06	44.35 $\pm$ 0.05
LEGO	8.23 $\pm$ 0.06	4.00 $\pm$ 0.00	41.55 $\pm$ 0.10	41.02 $\pm$ 0.02

5.2.5 Ablation Study (RQ5). Table 2 presents the results of an ablation study conducted using NCF on the ML-1M dataset. We sequentially remove the embedding calibration step and the flexible combination step to assess their impact on the unlearning (F1 and BAcc) and recommendation (HR and NDCG) performance. Initially, when we replace the embedding calibration step with D2DFR to unlearn a single attribute (D2DFR-FC), we observe a significant increase in F1 score and BAcc and a significant decrease in HR and NDCG. This indicates that without the embedding calibration step using MI minimization, the flexible combination step cannot guarantee unlearning effectiveness. Subsequently, we remove the flexible combination step and combine the embeddings by averaging them (EC-AC). Although the model performed well in recommendation performance, there is a noticeable decline in unlearning performance. This suggests that combining the embeddings by averaging them may result in suboptimal weights, thereby reducing unlearning performance. The results of the ablation study clearly demonstrate the critical roles of both steps in our proposed LEGO.

6 Conclusion

In this paper, we investigate multiple-attribute unlearning in recommender systems, aiming to simultaneously remove multiple sensitive attributes while efficiently adapting to dynamic privacy protection requirements. To the best of our knowledge, we are the first to identify the dynamic privacy protection requirements that often involve multiple sensitive attributes and evolve over time and across regions. Existing single-attribute unlearning methods fail to meet these requirements due to two key challenges: i) **CH1**: the inability to handle multiple unlearning requests simultaneously, and ii) **CH2**: the lack of adaptability to dynamic unlearning needs. To address these challenges, we propose LEGO, which decomposes multiple-attribute unlearning into two steps: *Embedding Calibration* and *Flexible Combination*. We conduct extensive experiments on three real-world datasets and three representative recommendation models to evaluate the effectiveness and efficiency of our proposed method. The results demonstrate that LEGO achieves performance comparable to baseline methods in single-attribute unlearning and outperforms them in multiple-attribute unlearning while preserving recommendation performance. Furthermore, our method proves to be highly efficient in adapting to dynamic privacy protection requirements. Note that all existing work focuses on discrete attributes or uses binning to transform continuous attributes into discrete ones, yet continuous attributes are prevalent in the real world.

Acknowledgments

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China (No. LZQ25F020002), Hangzhou Key Scientific Research Plan (No. 2024SZD1A28), and the National Natural Science Foundation of China (No. 62402148).

References

- [1] Ghazaleh Beigi, Ahmadrza Mosallanezhad, Ruocheng Guo, Hamidreza Alviri, 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*. 34–42.
- [2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *International conference on machine learning*. PMLR, 531–540.
- [3] Rob Bonta. 2022. California consumer privacy act (CCPA). Retrieved from State of California Department of Justice: <https://oag.ca.gov/privacy/ccpa> (2022).
- [4] Lucas Bourtole, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning.. In *IEEE Symposium on Security and Privacy (SP)*. 141–159.
- [5] Yinzhi Cao and Junfeng Yang. 2015. Towards Making Systems Forget with Machine Unlearning. *2015 IEEE Symposium on Security and Privacy* (2015).
- [6] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation Unlearning. In *Proceedings of the ACM Web Conference 2022*. ACM, 2768–2777.
- [7] Chaochao Chen, Jiaming Zhang, Yizhao Zhang, Li Zhang, Lingjuan Lyu, Yuyuan Li, Biao Gong, and Chenggang Yan. 2024. CURE4Rec: A Benchmark for Recommendation Unlearning with Deeper Influence. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [8] Chaochao Chen, Yizhao Zhang, Yuyuan Li, Jun Wang, Lianying Qi, Xiaolong Xu, Xiaolin Zheng, and Jianwei Yin. 2024. Post-Training Attribute Unlearning in Recommender Systems. *ACM Transactions on Information Systems* (2024).
- [9] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. CLUB: A Contrastive Log-ratio Upper Bound of Mutual Information.. In *International Conference on Machine Learning (ICML)*. 1779–1788.
- [10] Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.
- [11] Gustavo Escobedo, Christian Ganhör, Stefan Brandl, Mirjam Augstein, and Markus Schedl. 2024. Simultaneous Unlearning of Multiple Protected User Attributes From Variational Autoencoder Recommenders Using Adversarial Training. In *Advances in Bias and Fairness in Information Retrieval - 5th International Workshop, BIAS 2024*. 91–102.
- [12] Xiaohua Feng, Yuyuan Li, Fengyuan Yu, Li Zhang, Chaochao Chen, and Xiaolin Zheng. 2025. Plug and Play: Enabling Pluggable Attribute Unlearning in Recommender Systems. In *Proceedings of the ACM on Web Conference 2025*.
- [13] Christian Ganhör, David Penz, Navid Reksasz, 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, 2142–2147.
- [14] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks.. In *Computer Vision and Pattern Recognition (CVPR)*. 9301–9309.
- [15] Chuan Guo, T. Goldstein, Awni Y. Hannun, and L. Maaten. 2019. Certified Data Removal from Machine Learning Models. In *International Conference on Machine Learning*. Vol. abs/1911.03030.
- [16] 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 e-prints* (2022), arXiv–2202.
- [17] Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2016).
- [18] 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, 639–648.
- [19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.
- [20] 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 Security 18)*. 513–529.
- [21] Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. *Recommender systems handbook* (2021), 91–142.
- [22] Yuyuan Li, Chaochao Chen, Yizhao Zhang, Weiming Liu, Lingjuan Lyu, Xiaolin Zheng, Dan Meng, and Jun Wang. 2023. UltraRE: Enhancing RecEraser for Recommendation Unlearning via Error Decomposition.. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- [23] 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.
- [24] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Biao Gong, Jun Wang, and Linxun Chen. 2023. Selective and collaborative influence function for efficient recommendation unlearning. *Expert Systems with Applications* 234 (2023), 121025.
- [25] 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, 984–994.
- [26] Yuyuan Li, Xiaohua Feng, Chaochao Chen, and Qiang Yang. 2024. A Survey on Recommendation Unlearning: Fundamentals, Taxonomy, Evaluation, and Open Questions. *arXiv preprint arXiv:2412.12836* (2024).
- [27] Y. Li, Y. Shan, Y. Liu, H. Wang, W. Wang, Y. Wang, and R. Li. 2025. Personalized Federated Recommendation for Cold-Start Users via Adaptive Knowledge Fusion. In *Proceedings of the ACM Web Conference 2025*. 2700–2709.
- [28] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*. ACM Press, 689–698.
- [29] Sibe Liu, Yuanzhe Zhang, Xiang Li, Yunbo Liu, Chengwei Feng, and Hao Yang. 2025. Gated Multimodal Graph Learning for Personalized Recommendation. *INNO-PRESS: Journal of Emerging Applied AI* 1, 1 (2025).
- [30] Yunbo Liu, Xukui Qin, Yifan Gao, Xiang Li, and Chengwei Feng. 2025. SE-Transformer: A Hybrid Attention-Based Architecture for Robust Human Activity Recognition. *INNO-PRESS: Journal of Emerging Applied AI* 1, 1 (2025).
- [31] Sascha Meyen. 2016. *Relation between classification accuracy and mutual information in equally weighted classification tasks*. Ph. D. Dissertation. University of Hamburg Hamburg, Germany.
- [32] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A Survey of Machine Unlearning. *arXiv abs/2209.02299* (2022).
- [33] 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 preprint arXiv:1806.01246* (2018).
- [34] Teresa Scassa. 2020. Data Protection and the Internet: Canada. *Data Protection in the Internet* (2020), 55–76.
- [35] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*. Springer, 291–324.
- [36] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and A. Suresh. 2021. Remember What You Want to Forget: Algorithms for Machine Unlearning. In *Neural Information Processing Systems*, Vol. abs/2103.03279.
- [37] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 1–45.
- [38] Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Dewei Leng, Yanan Niu, Yang Song, Xiao Zhang, and Jun Xu. 2023. KuaiSAR: A Unified Search And Recommendation Dataset.. In *International Conference on Information and Knowledge Management*.
- [39] 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.
- [40] Chao Wang, Chuanhao Nie, and Yunbo Liu. 2025. Evaluating Supervised Learning Models for Fraud Detection: A Comparative Study of Classical and Deep Architectures on Imbalanced Transaction Data. *preprint arXiv:2505.22521* (2025).
- [41] H. Wang, Y. Jia, M. Zhang, Q. Hu, H. Ren, P. Sun, Y. Wen, and T. Zhang. 2024. Feddse: Distribution-aware Sub-model Extraction for Federated Learning over Resource-constrained Devices. In *Proceedings of the ACM Web Conference 2024*. 2902–2913. doi:10.1145/3587102.3638960
- [42] H. Wang, Y. Li, W. Xu, R. Li, Y. Zhan, and Z. Zeng. 2023. DAFKD: Domain-Aware Federated Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20412–20421.
- [43] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2023. Machine Unlearning of Features and Labels.. In *Network and Distributed System Security Symposium (NDSS)*.
- [44] Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, Yanjie Fu, and Meng Wang. 2020. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 679–688.
- [45] Y. Wu, S. Zhou, M. Yang, L. Wang, H. Chang, W. Zhu, X. Hu, X. Zhou, and X. Yang. 2025. Unlearning Concepts in Diffusion Model via Concept Domain Correction and Concept Preserving Gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 8496–8504.
- [46] Fan Zhang, Gongguan Chen, Hua Wang, and Caiming Zhang. 2024. CF-DAN: Facial-expression recognition based on cross-fusion dual-attention network. *Computational Visual Media* 10, 3 (2024), 593–608.
- [47] 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 Web Conference 2021*. 3002–3014.

- [48] Yang Zhang, Zhiyu Hu, Yimeng Bai, Jiancan Wu, Qifan Wang, and Fuli Feng. 2025. Recommendation Unlearning via Influence Function. *ACM Transactions on Recommender Systems* 3, 2 (2025), 1–23.
- [49] Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie GU, and Wenwu Zhu. 2022. On the Convergence of Stochastic Multi-Objective Gradient Manipulation and Beyond. In *Advances in Neural Information Processing Systems*, Vol. 35. 38103–38115.
- [50] Xue Zhu and Yuqing Sun. 2016. Differential privacy for collaborative filtering recommender algorithm. In *Proceedings of the 2016 ACM on international workshop on security and privacy analytics*. 9–16.

A Experimental Details

Table 3: Statistics of datasets after pre-processing.

Dataset	Attribute	Category #	User #	Item #	Rating #	Sparsity
ML-100K	Gender	2	943	1,682	100,000	93.695%
	Age	3				
	Occupation	21				
ML-1M	Gender	2	6,040	3,706	1,000,209	95.531%
	Age	3				
	Occupation	21				
KuaiSAR	Feat1	8	25,473	284,996	4,619,183	99.936%
	Feat2	3				

Dataset pre-processing. For ML-100K and ML-1M, we retrain only users who have interacted with at least five items. For KuaiSAR, we retain only users who have interacted with at least five items and items that have received at least five user interactions. To assess recommendation performance, the most recent interaction items for each user (sorted by interaction timestamp) are retained for testing. For ML-100K and ML-1M, the available gender attribute is restricted to male and female categories. The age attribute is divided into three groups: under 28 years old, between 28 and 40, and over 40 for ML-100K, and under 25, between 25 and 35, and over 35 for ML-1M. For KuaiSAR, we use anonymized one-hot encoded categories of users as the target attributes. We summarize the statistics of the above datasets after pre-processing in Table 3.

It is worth noting that we do not use the LFM-2B dataset, which has been widely used in previous work, because it is not available for download due to license issues.

Evaluation Metrics. We specify the evaluation metrics of unlearning effectiveness and recommendation performance as follows.

As mentioned in Section 3, the attack process is considered a classification task, where the attack model takes user embeddings as input and the attributes as labels. Following [8, 25?], we build a Multilayer Perceptron (MLP) [?] as the adversarial classifier, since MLP demonstrates the best performance as the attacker, as shown in [25]. The dimension of MLP’s hidden layer is set to 100, with a softmax layer as the output layer. We set the L2 regularization weight to 1.0, the initial learning rate to 1e-2, and the maximum number of iterations to 500, leaving the other hyperparameters at their default values in scikit-learn 1.4.2. We train the MLP using 80% of the users and test it on the remaining 20%. To evaluate the effectiveness of attribute unlearning, we use two widely adopted classification metrics: the micro-averaged F1 score (F1) and Balanced Accuracy (BAcc). Lower values of F1 and BAcc indicate greater effectiveness of attribute unlearning. We report the results of the attack using five-fold cross-validation. For the results of the multiple-attribute attack, we report the average F1 and BAcc across all attributes.

To assess recommendation performance, we use leave-one-out testing [? ? ?]. We use Hit Ratio at rank K (HR@K) and Normalized Discounted Cumulative Gain at rank K (NDCG@K) as metrics to evaluate recommendation performance. HR@K measures whether

the test item is in the top-K list, while NDCG@K is a position-aware ranking metric that gives higher scores to hits that occur at higher ranks. In our experiment, the entire negative item set is used to compute HR@K and NDCG@K. Note that we compare the recommendation performance of several methods under the condition of achieving optimal unlearning effectiveness.

Training parameters. For model-specific parameters in the recommendation models, we follow the settings provided in the respective original papers. Specifically, we use the Adam optimizer with a learning rate of 1e-3 and set the embedding dimension to 32 for NCF and LightGCN, and 200 for MultVAE. The number of epochs is set to 20, 100, and 20 for NCF, LightGCN, MultVAE, respectively.

Details of Applying Single-Attribute Methods for Multi-Attribute Unlearning. We apply single-attribute unlearning methods sequentially by removing one attribute at a time. For example, to unlearn Gender, Age, and Occupation on MovieLens using D2DFR, we first apply D2DFR to remove Gender, obtaining an intermediate model $\mathcal{M}_1$; then apply D2DFR on $\mathcal{M}_1$ to unlearn Age, resulting in $\mathcal{M}_2$; finally, apply D2DFR again on $\mathcal{M}_2$ to unlearn Occupation, yielding the final unlearned model.

Hyperparameters. To obtain the optimal performance for all methods, we use grid search to tune the hyperparameters. In D2DFR, we set the trade-off coefficient 1e-6. In AdvX, we set the gradient scaling coefficient to be 600. In our proposed LEGO, we set ϵ/N to 0.5. We use the Adam optimizer with a learning rate of 1e-3 to optimize the embeddings during the embedding calibration step. We construct a two-layer MLP as the variational distribution, with a hidden layer dimension of 100 and a softmax output layer. The learning rate of the MLP is set to 1e-4, and the training is run for 2000 iterations to ensure convergence.

Hardware information. All models and algorithms are implemented using Python 3.9 and Pytorch 2.3.0. The experiments are conducted on a server running Ubuntu 22.04, equipped with 256GB of RAM and an NVIDIA GeForce RTX 4090 GPU.

B Proof of Theorem 1

PROOF. For clarity of notation, let us define the combined embedding and the MI between the combined embedding and the sensitive attribute as follows:

$$U^{q_1}(\alpha^{q_2}) = \sum_{i=1}^k (\alpha_i^{q_2} \cdot U_i^{q_1}),$$

$$I_t^{(q_1, q_2)} = I(U^{q_1}(\alpha^{q_2}); A_t).$$

Using the triangle inequality, we can split the $|P_1 - P_2|$ into two terms:

$$|P_1 - P_2| = \left| \sum_{t=1}^k (I_t^{(1,1)} - I_t^{(2,2)}) \right|$$

$$\leq \left| \sum_{t=1}^k (I_t^{(1,1)} - I_t^{(2,1)}) \right| + \left| \sum_{t=1}^k (I_t^{(2,1)} - I_t^{(2,2)}) \right|.$$

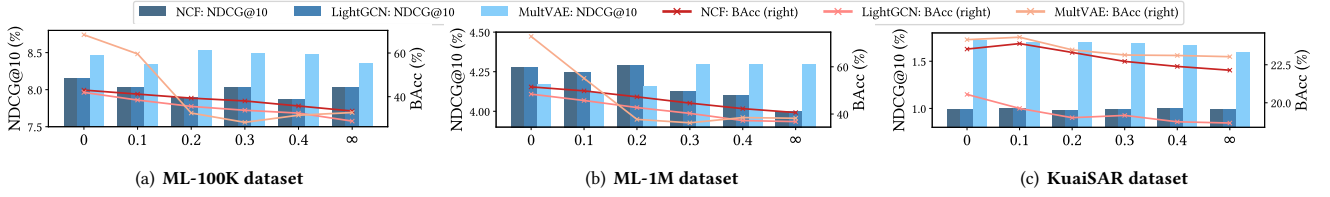


Figure 4: Effect of hyperparameter ϵ . We conduct experiments on all three models across three datasets. We use BAcc and NDCG@10 to represent the performance of unlearning and recommendation respectively. We report the results of unlearning all user attributes recorded in the dataset.

Applying the Lipschitz continuity of MI with respect to its first argument gives us the bounds of these two terms:

$$\begin{aligned} \left| \sum_{t=1}^k (I_t^{(1,1)} - I_t^{(2,1)}) \right| &\leq \sum_{t=1}^k L \left(\sum_{i=1}^k |\alpha_i^1| \cdot \|U_i^1 - U_i^2\|_2 \right) \\ &\leq \sum_{t=1}^k 2L\epsilon \left(\sum_{i=1}^k |\alpha_i^1| \right) = 2kL\epsilon, \\ \left| \sum_{t=1}^k (I_t^{(2,1)} - I_t^{(2,2)}) \right| &\leq \sum_{t=1}^k L \left\| \sum_{i=1}^k (\alpha_i^1 - \alpha_i^2) \cdot U_i^2 \right\|_2 \\ &\leq 2kL(C + \epsilon), \end{aligned}$$

where $L > 0$ is the Lipschitz constant. Combining these two inequality, the total gap is bounded by:

$$|P_1 - P_2| \leq 2kL\epsilon + 2kL(C + \epsilon) = 2kL(C + 2\epsilon).$$

□

C Additional Experimental Results

C.1 Unlearning Correlated Attributes

Table 4: Results of evaluating the impact of unlearning one attribute on the inference performance of a correlated attribute. The experiment is conducted on the LightGCN model using the ML-1M dataset.

Attribute	Gender F1	Gender BAcc	Occupation F1	Occupation BAcc
Original	70.41	71.28	10.29	11.38
Gender	49.56	51.20	9.05	9.36
Occupation	68.34	70.72	4.65	4.68
Gender, Occupation	55.63	55.09	5.68	5.84

Additionally, we conduct an experiment on the LightGCN model using the ML-1M dataset to evaluate how unlearning one attribute affects the unlearning performance of a correlated attribute (e.g., gender and occupation), with results shown in the Table 4. We observe that unlearning one attribute slightly reduces F1 and BAcc on a correlated attribute, but the values remain higher than those after

LEGO. These results indicate that single-attribute unlearning provides some unintended privacy protection on correlated attributes, but LEGO remains necessary for effective multi-attribute unlearning, as it achieves lower AIA accuracy overall.

C.2 Empirical Validation of the Theoretical Bound

Table 5: Results of evaluating the empirical tightness of the theoretical bound in Theorem 1 across datasets and models.

Dataset	MI	NCF	LightGCN	MultVAE
ML-100K	P_1	0.4850	0.7478	0.8180
	P_2	0.4665	0.7239	0.7883
ML-1M	P_1	0.5040	0.7858	0.8655
	P_2	0.4843	0.7535	0.8502
KuaiSAR	P_1	0.0240	0.0043	0.0199
	P_2	0.0209	0.0041	0.0114

Theorem 1 provides a theoretical guarantee that a linear combination of user embeddings with one specific attribute information removed can lead to a user embedding in which all sensitive attribute information is unlearned, demonstrating that LEGO can protect multiple sensitive attributes simultaneously. While the bound is theoretically derived, its practical tightness and generalization across datasets and models are crucial for real-world applicability. Since MI cannot be computed directly in our setting, we follow prior work and use the variational upper bound estimated by vCLUB as a proxy. We empirically evaluate the values of P_1 and P_2 across three datasets and three model architectures.

As shown in Table 5, the gap between the theoretical bound and the estimated MI remains small across all settings, indicating that the bound is empirically tight.

C.3 Sensitivity to ϵ

Parameter sensitivity results with respect to ϵ are shown in Figure 4.