41st IEEE International Conference on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

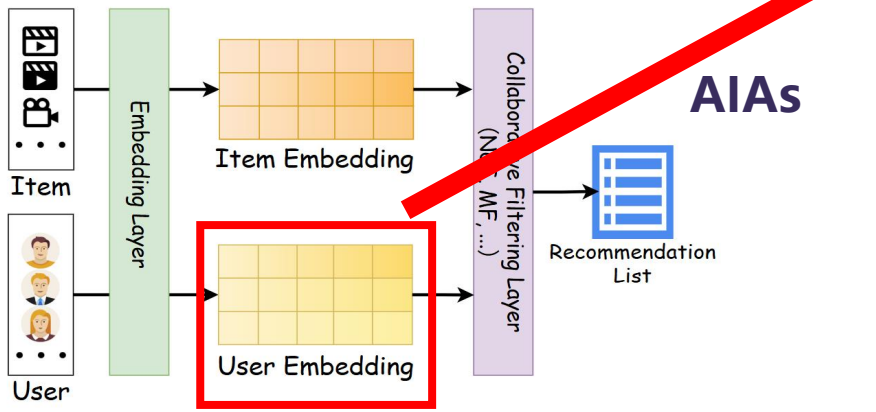
Defending against Attribute Inference Attacks in Post-Training of Recommendation Systems via Unlearning

Authors: Wenhan Wu, Yili Gong, Jiawei Jiang, Chuang Hu, Xiaobo Zhou and Dazhao Cheng

Affiliation: Wuhan University, University of Macau

Background: Attribute Inference Attacks (AIAs) in RS

Recommendation systems (RSs) are vulnerable to AIAs, where attackers leverage threat models to infer sensitive user attributes like age, gender from embeddings.



User Embeddings:

AIAs

Sensitive Attributes: **Gender** **Race** ...

Support

Unauthorized profiling and discriminatory practices

Background：Attribute Inference Attacks in RS

AIA Results on MovieLens-100K and MovieLens-1M dataset

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

The attack accuracy can reach up to 79.6%.

➢ Case 1：Netflix Prize Dataset De-anonymization

Although Netflix anonymized its user ratings dataset for a public competition, researchers successfully cross-referenced it with IMDb profiles and inferred private attributes.

NETFLIX

➢ Case 2：Attribute Inference in E-commerce Platforms

Some users voluntarily disclose attributes such as income level. Platforms then exploit these disclosures to infer the same attributes for users who choose not to reveal them.

E-COMMERCE

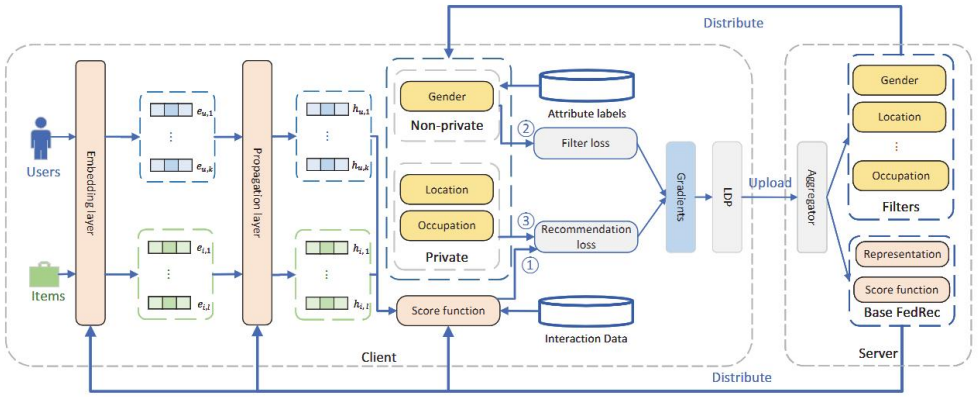
Background：Issues on In-training Methods

- In-training:
  - Adversarial training-based methods:
    Adding inference attacker model for adversarial.
  - Data modification-based methods:
    Adding a predefined number of dummy items to each user's profile.

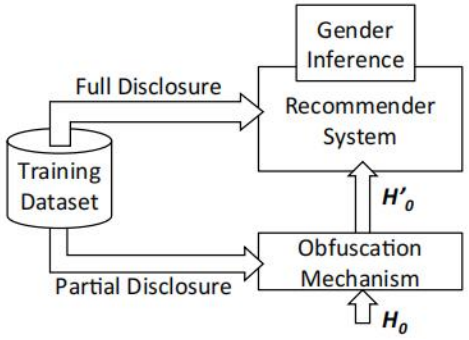
→ Predefined Sensitive Attributes

→ Extra modules required, Difficult to optimization

→ Distribution distortion, Reliance on raw data



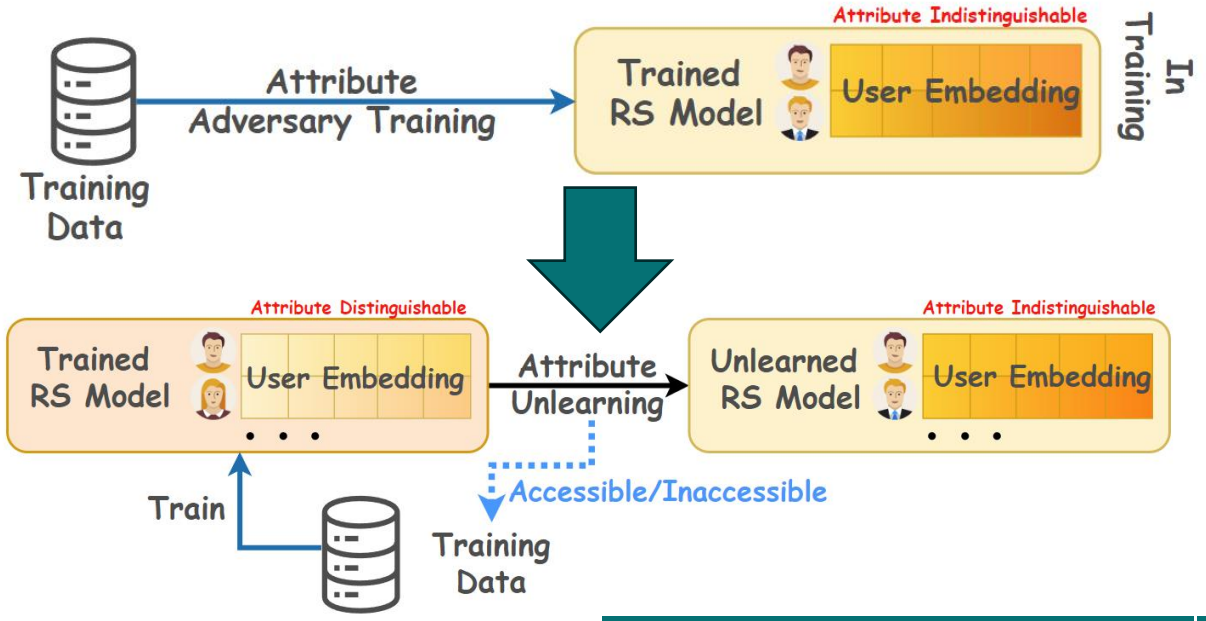
Adversarial training-based in-training methods



Data modification-based In-training methods

Performance?
Efficiency?
Flexibility?

Potential Solution: Attribute-wise Unlearning in RS



- A more flexible and efficient post-training approach.
- "unlearning" the attribute contributions or characteristics associated with user embeddings.

	Data-input unlearning	Attribute Unlearning
Unlearning Target	Input data (e.g. user-item interaction data)	Embedding attribute information
Target in Learning Phase	Used in learning phase	may not used during the training phase
"Gold Model"	Retrain model from scratch	No attribute privacy Exposure Under AIAs

Challenge 1： Applying Data-input RU to Attribute-wise?

➢ Applying data-input level unlearning to attribute-wise unlearning?

- data-input level RU cannot effectively decouple sensitive attributes from user embeddings to defend against AIAs, and even these attributes are not explicitly used as input during training.

Model Utility and AIA 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 accuracy of AIAs on the three private attributes showed no significant reduction after unlearning

Challenge 2： Balancing Performance and Privacy

➢ “Catastrophic Unlearning”

Due to overfitting to unlearning targets or conflicts in multi-task optimization (recommendation performance and privacy), unlearning methods often result in significant model accuracy loss, termed “catastrophic unlearning”

Preliminaries 1： Attribute Inference Attack Setting

➢ Attack Target

Framed as a classification task where the attackers train a model g to predict sensitive attributes z_i for the user embedding em_i

➢ Grey-box Attack Setting

The attackers do not have access to all model parameters but can access some user embedding vectors em_i and corresponding attribute information z_i. The threat model is trained on a shadow dataset D_{shadow}, which can be generated by sampling from the original user data within the same distribution.

➢ Threat Model

Train $\min_g \mathbb{E}_{(Em_{shadow},\ z_{shadow})}[\mathcal{L}_c(Z_{shadow},\ g(Em_{shadow})]$

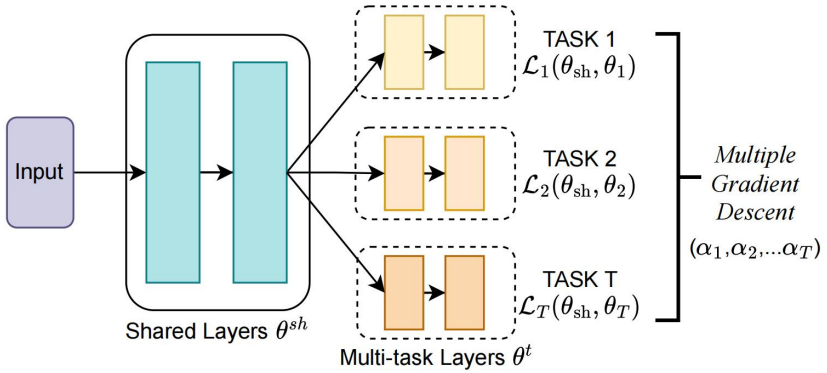
Attack $\dot{Z}_i = g(Em_i)$

Machine learning attacker MLP

Deep learning attacker XGBoost

Preliminaries 2: Parameter Sharing and GD in MTL

- To strike a balance between performance and privacy, we model the problem as a multi-task learning (MTL) task and adopt the hard parameter sharing approach.



Different tasks share the hidden layers of the model, while each task maintains its own independent output layer.

$$\min_{\theta_{sh}, \theta_1, \ldots, \theta_T} \sum_{t=1}^{T} \mathcal{L}_t(\theta_{sh}, \theta_t)$$

MTL (Hard Parameter Sharing)

Multi-objective Optimization Problem

- MTL can be achieved by minimizing a weighted empirical loss, which transforms into a multi-objective optimization problem, which is solved using gradient descent to reach a local optimum (Multiple Gradient Descent Algorithm (MGDA)).

KKT condition to achieve a *Pareto Stationary Point*

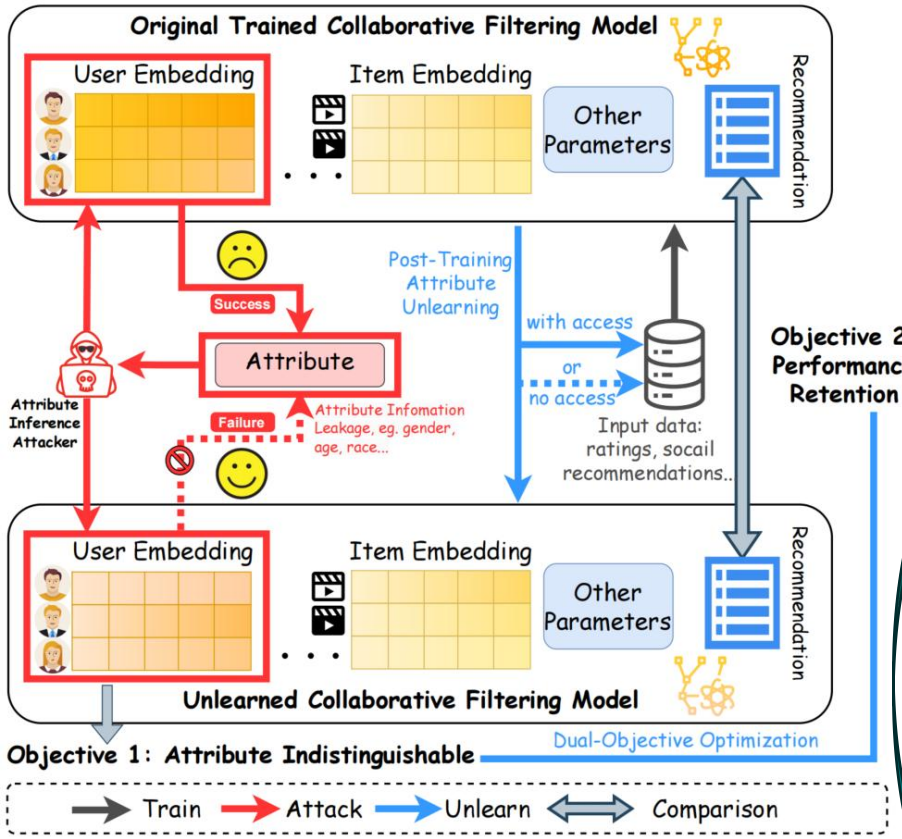
$$\begin{cases} \exists \alpha_1, \ldots, \alpha_T \geq 0, \text{ s.t.} \sum_{t=1}^{T} \alpha_t = 1 \text{ and } \sum_{t=1}^{T} \alpha_t \nabla_{\theta_{sh}} \mathcal{L}_t(\theta_{sh}, \theta_t) = 0 \\ \forall t, \nabla_{\theta_t} \mathcal{L}_t(\theta_{sh}, \theta_t) = 0 \end{cases}$$

- further represent the optimization problem as:

$$\min_{\alpha_1, \alpha_2, \ldots, \alpha_T} \left\| \sum_{t=1}^{T} \alpha_t \nabla_{(\theta_{sh}, \theta_t)} \mathcal{L}_t(\theta_{sh}, \theta_t) \right\|_2^2$$

subject to $\sum_{t=1}^{T} \alpha_t = 1$ and $\alpha_t \geq 0$ for all t. If the solution is 0, the KKT conditions are satisfied. Otherwise, the solution provides a descent direction that can improve all tasks.

Design：System Overview



- Objective 1: Attributes Indistinguishable

  Minimizing the upper bound of mutual information

  - Effectively removes the association between sensitive attributes and user embeddings.

$$\min_{em_i'} \sum_{a_j \in A} I(em_i', a_j)$$

$$\min_{em_i'} \sum_{a_j \in A} \left| \mathrm{Acc}\left(infer(em_i'), a_j\right) - \mathrm{Acc}\left(Random(em_i'), a_j\right) \right|$$

- Objective 2: Recommendation Knowledge Retention

  Minimizing the recommendation of regularization loss

  - Recommendation performance remains consistent post-unlearning.

$$\min_{em_i'} \mathrm{Dist}(M(em_i'), M(em_i))$$

- Objective 3: Dual-objective Optimization

  Under parameter-self sharing mechanism.

Design：Compositional Attribute Unlearning

➢ Private Attributes Information Loss

AttrCloak approximate the mutual information upper bound $\mathcal{L}_{i,j}^{AU}$ of attribute au_j ($|U_{au_j}|$ classes, for class $C_k \in U_{au_j}$, the number of users is $\left|S_{au_j=C_k}\right|$) using a variational upper bound based on the KL divergence:

$$\mathcal{L}_j^{AU} = I(em_i'; a_j) \leq \sum_{k=1}^{|U_{a_j}|} \frac{\left|S_{a_j=C_k}\right|}{\left|S_{a_j}\right|} D_{KL}(q_\phi(em_i'|Em_{au_j=C_k}) \mid p(em_i'))$$

- $D_{KL(q\|p)}$:Kullback–Leibler divergence variational distribution
- $q_\phi\left(em_i'\middle|Em_{au_j=C_k}\right)$: variational embedding distribution conditioned on the attribute $a_j = C_k$
- $p(em_i)$: prior represents the embedding distribution conditioned only on the input Em.

Design：Compositional Attribute Unlearning

➢ Private Attributes Information Loss

The probability distribution of embeddings for each class can be estimated, for instance, usually by fitting a Gaussian distribution

◆ Computing the user embedding distribution for each class (fitting a Gaussian distribution)：

$$\mu_{j,k} = \frac{1}{|S_{a_j=c_k}|}\sum_{em'_i \in S_{a_j=c_k}} em'_i, \qquad \Sigma_{j,k} = \frac{1}{|S_{a_j=c_k}|}\sum_{em'_i \in S_{a_j=c_k}} (em'_i - \mu_{j,k})\,(em'_i - \mu_{j,k})^T$$

$$\mu_{global} = \frac{1}{|S_{a_j}|}\sum_{em'_i \in S_{a_j}} em'_i, \qquad \Sigma_{global} = \frac{1}{|S_{a_j}|}\sum_{em'_i \in S_{a_j}} (em'_i - \mu_{global})\,(em'_i - \mu_{global})^T$$

◆ Calculating the KL divergence between each class embedding distribution and the global embedding distribution：

$$D_{KL}(N(\mu_{j,k}, \Sigma_{j,k})||N(\mu_{global}, \Sigma_{global}) = \frac{1}{2}[\log\frac{\det(\Sigma_{global})}{\det(\Sigma_{j,k})} - d + Tr(\Sigma_{global}^{-1}\Sigma_{j,k}) + (\mu_{global} - \mu_{j,k})^T \Sigma_{global}^{-1} (\mu_{global} - \mu_{j,k})]$$

◆ Calculating the total unlearning loss function $\mathcal{L}_j^{AU}$ for au_j as below:

$$\mathcal{L}_j^{AU} = \sum_{k=1}^{|U_{a_j}|} \frac{|S_{a_j=c_k}|}{|S_{a_j}|} D_{KL}(N(\mu_{j,k}, \Sigma_{j,k})||N(\mu_{global}, \Sigma_{global}), \qquad \mathcal{L}^{AU} = \frac{1}{|A|}\sum_{j=1}^{|A|} \mathcal{L}_j^{AU}$$

Design: Compositional Attribute Unlearning

➢ Data-Dependent Recommendation Knowledge Retention Loss

An intuitive approach is to directly use the typical recommendation loss function from the training phase (e.g., binary cross-entropy (BCE), or Bayesian personalized ranking (BPR) loss)

$$\mathcal{L}^{Rec} = \mathcal{L}_{BCE,BPR,\dots}(s_{\psi}(f_{\varphi,p}(u), f_{\varphi,p}(i)), R)$$

➢ Data-Free Recommendation Knowledge Loss

Since the interaction data may be inaccessible due to privacy restrictions or data modifications post training. In such scenarios, we propose the use of a L2-regularization $\mathcal{L}^{Reg}$ to help preserve recommendation performance.

$$\mathcal{L}^{Reg} = \sum_{i=1}^{|U|} ||em_i - em_i'||^2 = \sum_{i=1}^{|U|} \sum_{j=1}^{d} ||em_{i,j} - em_{i,j}'||^2$$

Combining the loss above, we get a dual-objective optimization problem, represented as follows:

$$\min_{em_i'} \mathcal{L}^{All} = \min_{em_i'} \alpha_u \mathcal{L}^{AU} + \alpha_r \mathcal{L}^{R}, \text{ where } \mathcal{L}^{R} = \mathcal{L}^{Rec} \text{ or } \mathcal{L}^{R} = \mathcal{L}^{Reg}$$

Design： Dual-objective Optimization with Parameter Self-sharing

➢ Why Parameter Self-sharing?

- ➢ To accelerate the training phase, the layers following the user embedding, as well as the item embedding, do not participate in multi-task optimization and are therefore not classified as "task-specific" layers θ_{sh} . This mechanism is referred to as parameter self-sharing.

➢ Pareto Stationary Point in Attribute Unlearning:

$$\alpha_u(\nabla_{em_i'}(\mathcal{L}^{AU}) + (1-\alpha_u)(\nabla_{em_i'}(\mathcal{L}^{R})) = 0, \alpha_u \in [0, 1]$$

reformulate as the following optimization

$$\min_{\alpha_u} \left\| \alpha_u(\nabla_{em_i'}(\mathcal{L}^{AU}) + (1-\alpha_u)(\nabla_{em_i'}(\mathcal{L}^{AU})) \right\|_2^2$$

solve the following optimization

$$\widehat{\alpha_u} = \left[\frac{(\nabla_{em_i'}(\mathcal{L}^{R}) - \nabla_{em_i'}(\mathcal{L}^{AU}))^T \nabla_{em_i'}(\mathcal{L}^{R})}{\left\| \nabla_{em_i'}(\mathcal{L}^{AU}) - \nabla_{em_i'}(\mathcal{L}^{R}) \right\|_2^2} \right]_{+,1}^{T},$$

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

By adjusting α_u dynamically and continuously during the AU updates, a balance between the performance and privacy objectives is achieved.

Design： Summary

Algorithm 1 Attribute Unlearning in RS with AttrCloak

Input: Pre-trained model with user embeddings $\mathbf{em}_i$, sensitive attributes $\mathcal{A}$, learning rate η, interaction matrix $\mathbf{R}$ (optional).
Output: Fine-tuned user embeddings $\mathbf{em}_i'$ after unlearning.

Initialize $\mathbf{em}_i' = \mathbf{em}_i = f_p(f_{\varphi_u}(u_i))$.
while not converged **do**
  for all attribute $a_j \in \mathcal{A}$ **do**
    Estimate class distributions $\mathcal{N}(\mu_{j,k}, \Sigma_{j,k})$ for $|U_{a_j}|$ classes using Eq. (13), (14).
    Compute global distribution $\mathcal{N}(\mu_{\text{global}}, \Sigma_{\text{global}})$ using Eq. (15), (16).
    Compute loss $\mathcal{L}_j^{AU}$ using Eq. (18).
  end for
  Compute total attribute unlearning loss $\mathcal{L}^{AU}$ as in Eq. (19).
  if interaction data $\mathbf{R}$ is available **then**
    Compute recommendation loss $\mathcal{L}^{Rec}$ using Eq. (20).
    Set $\mathcal{L}^R = \mathcal{L}^{Rec}$. ▷ **Data-Dependent (DD) Protection**
  else
    Compute regularization loss $\mathcal{L}^{Reg}$ defined in Eq. (21).
    Set $\mathcal{L}^R = \mathcal{L}^{Reg}$. ▷ **Data-Free (DF) Protection**
  end if
  Define dual-objective loss $\mathcal{L}^{All}$ as in Eq. (22).
  Compute gradients $\nabla_{\mathbf{em}_i'}(\mathcal{L}^{AU})$ and $\nabla_{\mathbf{em}_i'}(\mathcal{L}^R)$.
  Solve optimization using Eq. (26) to find optimal α_u, α_r.
  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 $(\varphi_u, p) \leftarrow (\varphi_u, p) - \eta \nabla_{(\varphi_u, p)} \mathcal{L}^{All}$.
end while
return $\mathbf{em}_i'$.

➢ Data-Dependent Protection

When interaction data is available, AU is guided by a combined loss:

$$\mathcal{L}^{All} = \alpha_u \mathcal{L}^{AU} + \alpha_r \mathcal{L}^{Rec}$$

➢ 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:

$$\mathcal{L}^{All} = \alpha_u \mathcal{L}^{AU} + \alpha_r \mathcal{L}^{Reg}$$

α_u and α_r are optimized within a dual-objective parameter self-sharing framework to achieve a Pareto-optimal solution between the two goals.

Evaluations: Attribute Unlearning Performance

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
	DD	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
		BlurMe [65]	0.6300	0.3170	0.1525	0.6641	0.2957	0.1482	0.6606	0.5813	0.4023	0.4161
		LDP-SH [28]	0.6698	0.2692	0.1472	0.6854	0.3171	0.1217	0.6289	0.5977	0.4351	0.3783
	DF	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
	DD	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
		BlurMe [65]	0.6605	0.2162	0.0902	0.6177	0.1573	0.0803	0.5637	0.5776	0.2245	0.2711
		LDP-SH [28]	0.6976	0.2586	0.0690	0.6475	0.1838	0.0828	0.6787	0.5422	0.3266	0.2699
	DF	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

- Privacy Metric: the accuracy of attribute classifiers of AIAs.

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.

- Results

Under different data availability conditions (data-dependent and data free), AttrCloak outperforms existing baseline methods, reducing the attack capability on various sensitive user attributes.

Evaluations: Recommendation Performance

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	0.8172	0.7769	0.8034	0.7524
		RAP	0.5819	0.5590	0.5784	0.5473
		BlurMe	0.8053	0.7685	0.7941	0.7464
		LDP-SH	0.6183	0.5838	0.6174	0.5695
	DF	AttrCloak-DF	0.7728	0.7531	0.7661	0.7409
		U2U-R	0.6952	0.6735	0.6895	0.6619
		D2D-R	0.6685	0.6439	0.6614	0.6294
MovieLens-1M		Original	0.6473	0.6095	0.6324	0.5853
	DD	AttrCloak-DD	0.6554	0.6176	0.6404	0.5934
		RAP	0.6518	0.6224	0.6285	0.5979
		BlurMe	0.6468	0.6089	0.6306	0.5840
		LDP-SH	0.4907	0.4691	0.4928	0.4609
	DF	AttrCloak-DF	0.6523	0.6217	0.6277	0.5958
		U2U-R	0.6750	0.6243	0.6467	0.5904
		D2D-R	0.6318	0.6017	0.5981	0.5668
ModCloth		Original	0.7297	0.7483	0.7678	0.8203
	DD	AttrCloak-DD	0.7928	0.8093	0.8292	0.8734
		RAP	0.3842	0.3860	0.3956	0.4148
		BlurMe	0.7168	0.7345	0.7512	0.8013
		LDP-SH	0.5647	0.6004	0.6262	0.7307
	DF	AttrCloak-DF	0.5470	0.5523	0.5589	0.5781
		U2U-R	0.5084	0.5053	0.5162	0.5321
		D2D-R	0.5082	0.5053	0.5164	0.5322
Last.FM-1K		Original	0.6696	0.6318	0.6556	0.6097
	DD	AttrCloak-DD	0.6655	0.6281	0.6498	0.6049
		RAP	0.5693	0.5297	0.5556	0.5062
		BlurMe	0.6452	0.6008	0.6106	0.5694
		LDP-SH	0.6113	0.5703	0.5943	0.5433
	DF	AttrCloak-DF	0.6060	0.5707	0.5888	0.5477
		U2U-R	0.5135	0.4788	0.4972	0.4564
		D2D-R	0.5657	0.5313	0.5498	0.5096

- **RS Performance Metric:**

  Reporting recommendation utility by calculating the average Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) across the ranked item lists of all test users. We truncate the ranked lists for both metrics at positions 5 and 10.

- **Results**

  In the DF scenario, AttrCloak outperforms both U2U-R and D2D-R in preserving recommendation performance. In the DD scenario, AttrCloak effectively mitigates attribute-induced data bias, even improves the original recommendation performance in most cases.

Evaluations: Unlearning Efficiency and Visualization

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

- Running Efficiency:
  - **DD:** As a post-training method, shows much higher efficiency (>5X) than in-training baselines.
  - **DF:** Slightly higher runtime costs compared to D2D-R due to the dual-objective optimization overhead.

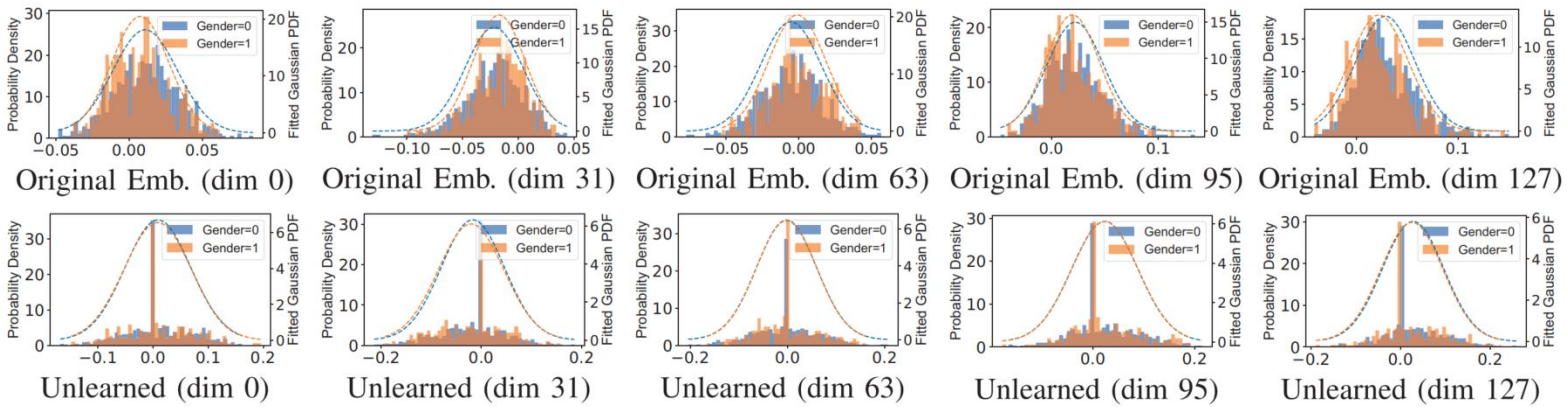
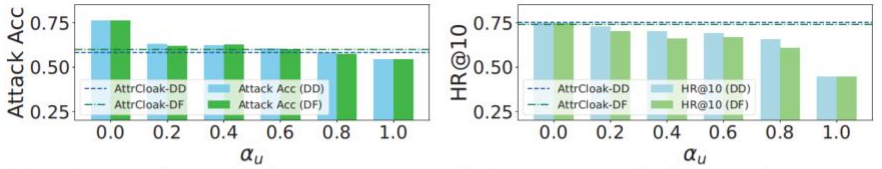


Fig. 4: Comparison of the Distribution of User Embeddings Before and After Attribute Unlearning (Gender) on the MovieLens-100K Dataset. Each Plot Represents a Selected Dimension of Embeddings; 0 and 1 Represent Female and Male, Respectively.

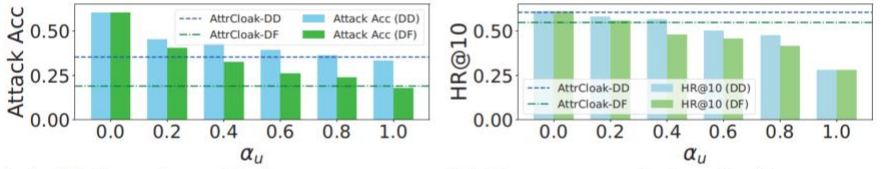
- Visualization:

The embedding distributions for different categories become more overlapping after unlearning, with reduced distinguishability. This makes it more difficult for AIA attackers to infer sensitive attributes

Evaluations: Ablation and Backbone Robustness Study



(a) Unlearning Performance on Movielens-100K. (b) Recommendation Performance on Movielens-100K.



(c) Unlearning Performance on Last.FM-1K. (d) Recommendation Performance on Last.FM-1K.

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

- Ablation Study: the impact of dual objective optimization.

Replacing the dynamic dual-objective weighted optimization α_u with fixed parameter values results in degraded performance or privacy, indicating the limitations of manually setting these values.

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	Original		0.7029	0.3422	0.1923	0.7520	0.3382	0.2255	0.8125	0.7847	0.7997	0.7657
	DD	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
		LDP-SH	0.6565	0.1923	0.1194	0.6260	0.2719	0.1326	0.6391	0.6144	0.6251	0.4962
	DF	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

- Backbone Robustness Results

Across different backbone models (LightGCN), AttrCloak still exhibits strong effectiveness, highlighting its robustness across different RS models

Conclusion

- We present the post-training AU framework, AttrCloak. It better handles flexible privacy requirements and adapts to real-world data environments.
- We reformulate AU as a dual-objective optimization problem with parameter self-sharing.
- We design a novel effective multi-component loss based on information theory.
- We implemented AttrCloak and evaluated its ability to balance privacy and recommendation utility on four real-world datasets.

Thank You!

Q & A

Email: wenhanwu@whu.edu.cn