Debiasing Learning for Membership Inference Attacks Against Recommender Systems

Zihan Wang*
Shandong University
zihanwang.sdu@gmail.com

Na Huang*
Shandong University
hn.z@mail.sdu.edu.cn

Fei Sun
Alibaba Group
ofey.sunfei@gmail.com

Pengjie Ren
Shandong University
jay.ren@outlook.com

Zhumin Chen
Shandong University
chenzhumin@sdu.edu.cn

Hengliang Luo
Meituan
luohengliang@meituan.com

Maarten de Rijke
University of Amsterdam
m.derijke@uva.nl

Zhaochun Ren†
Shandong University
zhaochun.ren@sdu.edu.cn

ABSTRACT

Learned recommender systems may inadvertently leak information about their training data, leading to privacy violations. We investigate privacy threats faced by recommender systems through the lens of membership inference. In such attacks, an adversary aims to infer whether a user's data is used to train the target recommender. To achieve this, previous work has used a shadow recommender to derive training data for the attack model, and then predicts the membership by calculating difference vectors between users' historical interactions and recommended items. State-of-the-art methods face two challenging problems: (i) training data for the attack model is biased due to the gap between shadow and target recommenders, and (ii) hidden states in recommenders are not observational, resulting in inaccurate estimations of difference vectors.

To address the above limitations, we propose a *Debiasing Learning for Membership Inference Attacks against recommender systems* (DL-MIA) framework that has four main components: (i) a difference vector generator, (ii) a disentangled encoder, (iii) a weight estimator, and (iv) an attack model. To mitigate the gap between recommenders, a variational auto-encoder (VAE) based disentangled encoder is devised to identify recommender invariant and specific features. To reduce the estimation bias, we design a weight estimator, assigning a truth-level score for each difference vector to indicate estimation accuracy. We evaluate DL-MIA against both general recommenders and sequential recommenders on three real-world datasets. Experimental results show that DL-MIA effectively alleviates training and estimation biases simultaneously, and achieves state-of-the-art attack performance.

*These authors contributed equally to this work.
†Corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539392

CCS CONCEPTS

• **Security and privacy**; • **Information systems** → **Recommender systems**;

KEYWORDS

Recommender system, Membership inference attack, Debiasing

ACM Reference Format:
Zihan Wang, Na Huang, Fei Sun, Pengjie Ren, Zhumin Chen, Hengliang Luo, Maarten de Rijke, and Zhaochun Ren. 2022. Debiasing Learning for Membership Inference Attacks Against Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539392

1 INTRODUCTION

The success of today's recommender systems is largely attributed to the increased availability of large-scale training data on users' private information (e.g., browsing and purchase history). Unfortunately, various studies show that recommender systems are vulnerable to attacks, leading to the leakage of their training data and severe privacy problems [4, 46].

In this paper, we study privacy threats faced by recommender systems through the lens of membership inference [46]. Specifically, membership inference attacks (MIAs) against recommender systems enable the adversary to infer whether a user's data is used to train the target recommender [53]. The main reason for the feasibility of MIA is overfitting, since the learned model tends to perform better on the training data [5]. Revealing the membership may cause serious harm, and leak sensitive information about specific individuals, such as shopping preferences, social relationships, and location information [9].

Existing MIA methods show promising performance in various domains, ranging from biomedical data [2, 13, 18] to mobility traces [37]. Despite the success, previous MIA methods [9, 29, 35, 42, 46, 52] cannot be directly applied to recommender systems, since they either require knowledge of the target model or use the predicted confidence scores of the classifier. In MIAs against recommender systems, the target recommenders are considered inaccessible, and only recommended items, rather than confidence

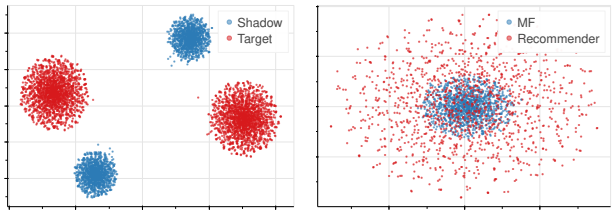
scores, are observational to the adversary [53]. In fact, this setting is prevalent in real-world scenarios.

In recent work, Zhang et al. [53] infer the membership of the target recommender based on the similarity between users' historical interactions and recommended items. The key idea here is, for users in the training set, their historical interactions tend to be more similar to output items of the recommender. Specifically, a shadow recommender is first established to simulate the target recommender and generate training data for the attack model. Then, difference vectors between users' historical interactions and recommended items are computed by factorizing the user-item rating matrix. On this basis, the attack model is able to predict the membership using difference vectors. This framework faces two challenging problems:

(1) **Training data for the attack model is biased.** As mentioned above, the algorithm and dataset used by the target recommender are inaccessible [53]. In that case, the adversary may construct a shadow recommender in a completely distinct manner, resulting in a biased training dataset for the attack model. In Figure 1(a), feature vectors from the MIA datasets generated by target and shadow recommenders (that use different methods) are visualized by the t-SNE algorithm [50], respectively. And there exist huge differences between the distributions of features obtained from the shadow recommender (blue) and target recommender (red). Besides, as mentioned in [53], the attack performance drops dramatically when target and shadow recommenders use different algorithms and datasets.
To mitigate the gap between recommender systems, we employ a variational auto-encoder (VAE) based encoder to disentangle features, and model recommender invariant and specific characteristics using two different distribution families.

(2) **The estimations of difference vectors are inaccurate.** In this attack, as explained above, the hidden states (e.g., user and item representations) in the target recommender are not available to the adversary. As a result, difference vectors between user historical interactions and recommended items may be estimated inaccurately for the target recommender, leading to incorrect membership predictions. For example, as demonstrated in Figure 1(b), the difference vectors generated by the target recommender (red) and matrix factorization (MF) (blue) are divergently distributed.
To reduce the influence of the estimation bias, we develop a weight estimator, and learn a truth-level score for each difference vector to indicate the estimation accuracy during training.

To address the above problems, we propose a framework, named *Debiasing Learning for Membership Inference Attacks against recommender systems* (DL-MIA), to simultaneously mitigate training data and estimation biases. As illustrated in Figure 2, DL-MIA has four main components: (i) a difference vector generator, (ii) a disentangled encoder, (iii) a weight estimator, and (iv) an attack model. During training, to simulate behavior of the target model, a shadow recommender is first constructed and learned on a shadow dataset. Then, the generator represents users' history interactions and recommended items by factorizing the user-item rating matrix, and calculates difference vectors. To mitigate the training data bias caused by the gap between target and shadow recommenders, the disentangled encoder is developed, and a variational auto-encoder (VAE) based on two distribution families is employed to identify recommender invariant and specific features. Next, to reduce the influence of the estimation bias, we establish a weight estimator, and assign a truth-level score for each difference vector. Finally, the disentangled and re-weighted difference vectors, as well as membership labels, are input for the multilayer perceptron (MLP) based attack model. In addition, to facilitate the model parameter update and weight learning, an alternating training strategy is applied among the disentangled encoder, weight estimator, and attack model.



(a) Training data bias. (b) Estimation bias.

Figure 1: Visualization results for the training data and estimation biases. (a) The bias between the MIA datasets generated by the shadow recommender (blue) and target recommender (red). (b) The bias between difference vectors generated using MF (blue) and hidden states in the recommender (red).

Our contributions can be summarized as follows: (i) To the best of our knowledge, ours is the first work to study debiasing learning for membership inference attacks against recommender systems. (ii) We develop a VAE based disentangled encoder to mitigate training data bias caused by the gap between shadow and target recommenders. (iii) We introduce truth-level scores, and learn the weight estimator with the alternating training strategy to alleviate the estimation bias of difference vectors. (iv) Experimental results show that DL-MIA achieves the state-of-the-art attack performance against both general and sequential recommender systems.

2 RELATED WORK

We survey related work along three dimensions: (i) membership inference attacks, (ii) general and sequential recommenders, and (iii) debiasing learning.

2.1 Membership inference attacks

Recently, membership inference attacks (MIAs) have achieved promising performance in various domains, such as biomedical data [2, 13, 18] and mobility traces [37]. The goal of membership inference attacks is to infer the membership of individual training samples for a target model. Shokri et al. [46] specify the first membership inference attack against machine learning models. The authors propose a general formulation of membership inference attack against machine learning models, and train multiple shadow models to simulate the target model's behavior. In that case, the training sets for multiple attack models (one for each class) are generated. Salem et al. [42] further relax several key assumptions from [46], including knowledge of the target model architecture and target dataset

distribution. Yeom et al. [52] explore the relationship between attack performance and overfitting, and propose the first decision-based attack. Nasr et al. [35] study membership inference attacks in both black-box and white-box settings. Instead of using output scores, several recent membership attacks [9, 29] assume only predicted hard labels of models are exposed, and demonstrate that label-only exposures are also vulnerable to membership leakage. In addition, Zhang et al. [53] investigate MIA against recommender systems, leveraging the differences between user history behaviors and output items from recommenders.

To mitigate the attacks, some defense mechanisms, including model stacking [42], dropout [42], adversarial training [34], differential privacy [9, 29], regularization [9, 29], and jointly maximizing privacy and prediction accuracy [19], have been proposed. To protect membership privacy of recommender systems, Zhang et al. [53] design a defense mechanism, named *Popularity Randomization*, and randomly recommend popular items to non-member users.

2.2 General and sequential recommenders

A generic recommender system aims to model users' preferences from their historical behavior. Early attempts on recommender systems, including matrix factorization (MF) [26, 27, 36, 41] and item-based neighborhood methods [20, 25, 31, 43], typically apply collaborative filtering (CF) on users' interaction histories. Recently, deep learning has been used to improve the performance of recommender systems by incorporating with auxiliary information [21, 23], or replacing the conventional matrix factorization [15, 45].

None of the above methods considers the order in users' behaviors or is designed for sequential recommendation. The earliest work on sequential recommendation, FPMC [39], utilizes Markov chain to capture the transition in behavior sequences. To further enhance the capability of modeling complex behavior, deep learning based models [16, 22, 48, 49, 55] are devised, including recurrent neural network based [16, 55], and attention based [22, 48] methods.

2.3 Debiasing learning

Bias is a critical issue in modern machine learning since trained models often fail to identify the proper representations for the target predictions [10]. To tackle the limitation, a large number of methods have been proposed to eliminate the biases. Specifically, to address selection bias [32] in datasets, propensity score [44, 51], ATOP [47], and data imputation [40] are utilized. Besides, debiasing strategies such as rebalancing [1], adversarial learning [3], and causal modeling [28] are proposed to mitigate unfairness [30] caused by algorithm and unbalanced data. In survey [6], seven types of biases with their definitions and characteristics are summarized and introduced in detail. However, the work listed does not consider the biases in MIA against recommender systems.

In this paper, we mainly focus on the membership inference attack (MIA) against recommender systems. To the best of our knowledge, ours is the first work to study debiasing learning for this task. The most closely related work is [53]. However, the previous MIA against recommender systems still face two challenging problems: (i) biased attack model training, (ii) inaccurate estimations of difference vectors. In our proposed DL-MIA, to mitigate the gap between target and shadow recommenders, the VAE based encoder is employed to model recommender invariant and specific features. In addition, to reduce the impacts of inaccurate estimations, the weight estimator is employed, and truth-level scores for difference vectors are calculated to facilitate the model update.

3 METHOD

We first formulate the membership inference attack (MIA) against recommender systems. Then, we give an overview of DL-MIA. Next, we explain DL-MIA's disentangled encoder and weight estimator. Finally, the learning algorithm is presented.

3.1 Problem formulation

Membership leakage in recommender systems happens when the adversary aims to determine whether a user's data is used to train the target recommender. Formally, given a user's data $\mathbf{x}$, a trained target recommender $\mathcal{M}_{target}$, and external knowledge of the adversary Ω, a membership inference attack model $\mathcal{A}$ can be defined as follows:

$$\mathcal{A} : \mathbf{x}, \mathcal{M}_{target}, \Omega \rightarrow \{0, 1\}, \tag{1}$$

where 0 means $\mathbf{x}$ is not a member of $\mathcal{M}_{target}$'s training dataset while 1 indicates $\mathbf{x}$ is a member. The attack model $\mathcal{A}$ is essentially a binary classifier.

Adversarial knowledge. In this attack, the adversary only has black-box access to the target recommender. Specifically, only the recommendations to users, and users' historical behaviors (e.g., ratings or interaction sequences) are observational. In that case, as explained in [53], the adversary can infer the membership using the similarity between users' historical behaviors and recommended items from the target model.

3.2 Model overview

Figure 2 shows the four main components of DL-MIA: (i) a difference vector generator, (ii) a disentangled encoder, (iii) a weight estimator, and (iv) an attack model. In this section, we give an overview of the DL-MIA framework.

3.2.1 Difference vector generator. Following Zhang et al. [53], to conduct MIA against the target recommender, a shadow recommender $\mathcal{M}_{shadow}$ is established, and difference vectors between users' historical behaviors and recommended items are calculated. To achieve this, we first factorize the user-item rating matrix to obtain item representations $\mathbf{M}^{item}$. Then, a shadow recommender $\mathcal{M}_{shadow}$ is established and trained to simulate the target recommender. Next, for the i-th user in $\mathcal{M}_{shadow}$, we project her/his interacted and recommended items into representations, denoted as $\mathbf{I}_{shadow,i}$ and $\mathbf{R}_{shadow,i}$, respectively. Finally, the difference vector for the i-th user is computed as:

$$\mathbf{f}^{diff}_{shadow,i} = \bar{\mathbf{I}}_{shadow,i} - \overline{\mathbf{R}}_{shadow,i}, \tag{2}$$

where $\bar{\mathbf{I}}_{shadow,i}$ and $\overline{\mathbf{R}}_{shadow,i}$ are the averages of item vectors in $\mathbf{I}_{shadow,i}$ and $\mathbf{R}_{shadow,i}$, respectively.

3.2.2 Disentangled encoder. To mitigate the training data bias, the disentangled encoder aims to identify features invariant and specific to shadow and target recommenders. Specifically, given generated

difference vector $\mathbf{f}^{diff}$, a VAE based encoder, composing two kinds of prior distributions, is employed to disentangle $\mathbf{f}^{diff}$ into the invariant feature $\mathbf{f}^{inv}$ and specific feature $\mathbf{f}^{spe}$. And the disentangled difference vector $\mathbf{f}^{dis}$ is obtained by concatenating $\mathbf{f}^{inv}$ and $\mathbf{f}^{spe}$, i.e., $\mathbf{f}^{dis} = [\mathbf{f}^{inv}; \mathbf{f}^{spe}]$.

3.2.3 Weight estimator. To further alleviate the influence of the estimation bias, the weight estimator assigns a truth-level score p to each disentangled difference vector $\mathbf{f}^{dis}$. To learn p, the estimation constraint is constructed. Moreover, to facilitate the model update and weight learning, an alternating training scheme is developed. In this way, the disentangled and reweighted difference vector $\mathbf{f}^{rew}$ is derived.

3.2.4 Attack model. For membership inference, a generic attack model $\mathcal{A}$ is essentially a binary classifier with the input of difference vectors. Following Zhang et al. [53], we adopt a MLP with 2 hidden layers for the attack model, i.e., $\mathcal{A} : \mathbf{y} = \text{MLP}(\mathbf{f}^{rew})$. The output $\mathbf{y} = (y_1, y_2)$ is a 2-dimensional vector indicating the probability of the user belonging to members (y_1) or non-members (y_2). And the binary cross-entropy loss is used to train the attack model:

$$\mathcal{L}_{BCE} = -\sum_{i=1}^{N_{shadow}} \left(y_i^* \log y_{i,1} + (1 - y_i^*) \log y_{i,2}\right), \tag{3}$$

where y_i^* is the ground truth label for i-th user, and N_{shadow} is the size of training data generated by the shadow recommender.

Since we use the difference vector generator and attack model of the same architecture as the previous work [53], only the disentangled encoder and weight estimator are explained in detailed in the following sections.

3.3 Disentangled encoder

Given the difference vector $\mathbf{f}^{diff}$ from the generator, the disentangled encoder aims to identify recommender invariant and specific features (i.e., $\mathbf{f}^{inv}$ and $\mathbf{f}^{spe}$). To achieve this, inspired by [8], we construct a variational auto-encoder (VAE) using Gaussian and von Mises Fisher (vMF) distributions to model recommender invariant and specific characteristics, respectively.

Specifically, in the encoder, we assume a difference vector is generated by conditioning on two independent latent variables: the recommender invariant feature $\mathbf{f}^{inv}$ and the recommender specific feature $\mathbf{f}^{spe}$. Thus, the joint probability in our model is computed as follows:

$$p_\theta(\mathbf{f}^{diff}, \mathbf{f}^{inv}, \mathbf{f}^{spe}) = p_\theta(\mathbf{f}^{inv}) p_\theta(\mathbf{f}^{spe}) p_\theta(\mathbf{f}^{diff} | \mathbf{f}^{inv}, \mathbf{f}^{spe}), \tag{4}$$

where $p_\theta(\mathbf{f}^{inv})$ and $p_\theta(\mathbf{f}^{spe})$ are the priors for $\mathbf{f}^{inv}$ and $\mathbf{f}^{spe}$, respectively. And $p_\theta(\mathbf{f}^{diff} | \mathbf{f}^{inv}, \mathbf{f}^{spe})$ denotes the likelihood. Following previous work [7, 54], we assume a factored posterior probability $q_\phi(\mathbf{f}^{inv}, \mathbf{f}^{spe} | \mathbf{f}^{diff}) = q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff}) q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff})$. Therefore, learning of our encoder maximizes an evidence lower bound on marginal log-likelihood:

$$\begin{aligned}
&\mathcal{L}_{ELBO} \\
&\stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\mathbf{f}^{inv}, \mathbf{f}^{spe}} \Big[\log p_\theta(\mathbf{f}^{diff} | \mathbf{f}^{spe}, \mathbf{f}^{inv}) - \log \frac{q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff})}{p_\theta(f^{spe})} \\
&\qquad - \log \frac{q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff})}{p_\theta(\mathbf{f}^{inv})} \Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{f}^{inv}, \mathbf{f}^{spe}} \big[\log p_\theta(\mathbf{f}^{diff} | \mathbf{f}^{spe}, \mathbf{f}^{inv}) \big] - KL(q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff}) \| p_\theta(\mathbf{f}^{spe})) \\
&\quad - KL\big(q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff}) \| p_\theta(\mathbf{f}^{inv})\big),
\end{aligned} \tag{5}$$

where $\mathbf{f}^{inv} \sim q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff})$ and $\mathbf{f}^{spe} \sim q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff})$. $q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff})$ and $q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff})$ are the posteriors. $KL(p \| q)$ denotes the KL divergence between the distribution p and q. In our disentangled encoder, two distribution families, i.e., the vMF and Gaussian distributions, are used to define the posteriors. Further details on the parameterization are provided below.

3.3.1 Gaussian Distribution. We assume that $q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff})$ follows a Gaussian distribution [38] $\mathcal{N}(\mu_\beta(\mathbf{f}^{diff}), \text{diag}(\sigma_\beta(\mathbf{f}^{diff})))$, and that the prior $p_\theta(\mathbf{f}^{inv})$ follows the standard distribution $\mathcal{N}(0, I)$, where I is an identity matrix. In our encoder, we only consider a diagonal covariance matrix, and thus the KL divergence term $KL(q_\phi(\mathbf{f}^{inv} | \mathbf{f}^{diff}) \| p_\theta(\mathbf{f}^{inv}))$ can also be obtained as follows:

$$\frac{1}{2}\Big(-\sum_i \log \sigma_{\beta i} + \sum_i \sigma_{\beta i} + \sum_i \mu_{\beta i}^2 - d\Big). \tag{6}$$

3.3.2 vMF Distribution. vMF can be recognized as a Gaussian distribution on a hypersphere with two parameters, μ, and κ. $\mu \in \mathbb{R}^m$ is a normalized vector (i.e., $\|\mu\|_2 = 1$) and defines the mean direction. $\kappa \in \mathbb{R}_{\geq 0}$ denotes the concentration parameter analogous to the variance in a Gaussian distribution.

In our encoder, we assume that $q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff})$ follows a vMF distribution $\text{vMF}(\mu_\alpha(\mathbf{f}^{diff}), \kappa_\alpha(\mathbf{f}^{diff}))$ and the prior $p_\theta(\mathbf{f}^{spe})$ follows the uniform distribution $\text{vMF}(\cdot, 0)$. The $KL(q_\phi(\mathbf{f}^{spe} | \mathbf{f}^{diff}) \| p_\theta(\mathbf{f}^{spe}))$ term in $\mathcal{L}_{ELBO}$ can then be computed in closed form:

$$\begin{aligned}
&\kappa_\alpha \frac{\mathcal{I}_{m/2}(\kappa_\alpha)}{\mathcal{I}_{m/2-1}(\kappa_\alpha)} + (m/2 - 1) \log \kappa_\alpha - (m/2) \log(2\pi) \\
&- \log \mathcal{I}_{m/2-1}(\kappa_\alpha) + \frac{m}{2} \log \pi + \log 2 - \log \Gamma(m/2),
\end{aligned} \tag{7}$$

where $\mathcal{I}_v$ is the modified Bessel function of the first kind at order v and $\Gamma(\cdot)$ is the Gamma function. Following Davidson et al. [11], we use an acceptance-rejection scheme to sample from the vMF distribution.

3.3.3 Reconstruction error. We assume the conditional likelihood distribution $p_\theta(\mathbf{f}^{diff} | \mathbf{f}^{spe}, \mathbf{f}^{inv})$ follows $\mathcal{N}(f([\mathbf{f}^{inv}; \mathbf{f}^{spe}]), \mathbf{I})$, where a MLP with 3 hidden layers is adopted for $f(\cdot)$. Thus, the reconstruction error (the first term) in $\mathcal{L}_{ELBO}$ can be rewritten as:

$$\mathop{\mathbb{E}}_{\mathbf{f}^{inv}, \mathbf{f}^{spe}} \left[-\frac{1}{2} \left\| f([\mathbf{f}^{inv}; \mathbf{f}^{spe}]) - \mathbf{f}^{diff} \right\|^2 \right]. \tag{8}$$

During training, we use a linear layer to produce μ_β, δ_β, μ_α, and κ_α. The difference vectors from both shadow and target recommenders are disentangled by the encoder. Note that membership labels in the target recommender are not exposed to DL-MIA. Through the encoder, the disentangled difference vector, i.e., $\mathbf{f}^{dis} = [\mathbf{f}^{inv}; \mathbf{f}^{spe}]$, is obtained to mitigate the gap between shadow and target recommender.

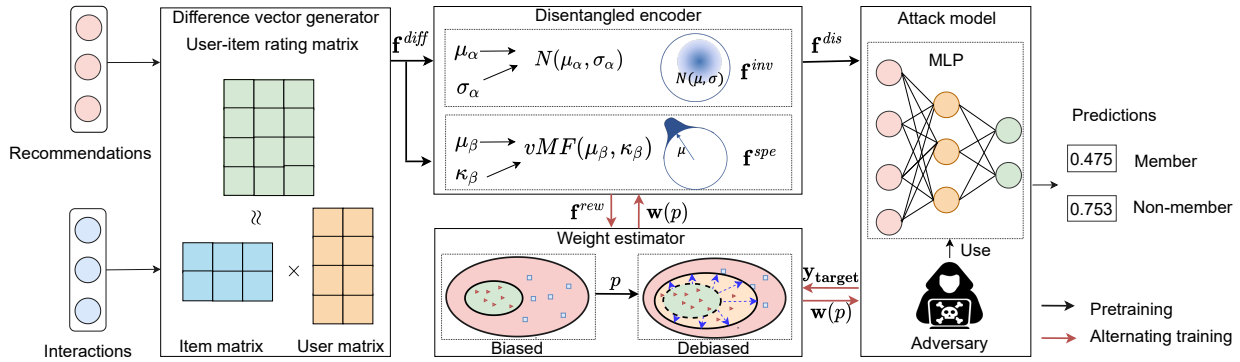


Figure 2: An overview of DL-MIA. DL-MIA has four main components: a difference vector generator, a disentangled encoder, a weight estimator, and an attack model.

3.4 Weight estimator

Given $\mathbf{f}^{dis}$ from the disentangled encoder, the weight estimator aims to alleviate the estimation bias of difference vectors. Specifically, we introduce the truth-level score to indicate the estimation accuracy. Then, we establish the estimation constraint, and assign a truth-level score for each difference vector. Moreover, to update model parameters and learn scores simultaneously, an alternating training strategy among the disentangled encoder, weight estimator, and attack model is adopted.

3.4.1 Truth-level score. As mentioned in Sec. 3.1, the hidden states in the target recommender, including item representations, are not observational to the adversary. As a result, difference vectors for recommenders may be computed inaccurately by MF in the generator. In the estimator, we write $\mathbf{f}'$ for the ground truth difference vector, and define the truth-level score p for $\mathbf{f}^{dis}$ as follows:

$$p = \frac{\delta\left(\mathcal{A}\left(\mathbf{f}'\right), y^*\right)}{\delta(\mathcal{A}(\mathbf{f}^{dis}), y^*)}, \tag{9}$$

where $\mathcal{A}(\cdot)$ denotes the attack model, and y^* is the membership label. $\delta(\cdot)$ is the error measure, for which we adopt the binary cross-entropy loss.

3.4.2 Alternating training. In the estimator, the truth-level score serves as the weighting parameter for the estimation bias. After debiasing by the truth-level score, the biased estimation should be equal to the unbiased estimation. Motivated by this, we can rewrite Eq. 9 as follows:

$$p \cdot \delta\left(\mathcal{A}\left(\mathbf{f}^{dis}\right), y^*\right) = \delta\left(\mathcal{A}\left(\mathbf{f}'\right), y^*\right). \tag{10}$$

On this basis, to compute the truth-level score p, the estimation constraint is established:

$$\mathcal{L}_{estimate} = \sum_j \lambda_j \cdot \sum_{i=1}^{N_j} \left\| p_{i,j} \cdot \delta\left(\mathcal{A}\left(\mathbf{f}^{dis}_{i,j}\right), y^*_{i,j}\right) - \delta\left(\mathcal{A}\left(\mathbf{f}'_{i,j}\right), y^*_{i,j}\right) \right\|^2, \tag{11}$$

where $j \in \{shadow, target\}$, and λ_j is the weight for the shadow or target recommender. Here, we set $\lambda_j = \frac{1}{N_j}$, where N_{shadow} and N_{target} are the size of training dataset generated by shadow recommender, and the test dataset for the target recommender, respectively.

However, membership labels $\mathbf{y}^*_{target}$ of the target recommender and the ground truth difference vector $\mathbf{f}'$ cannot be obtained directly. To address this issue and facilitate model update, we develop an alternating training strategy among the disentangled encoder, attack model, and weight estimator. Specifically, the re-weighted loss for the disentangled encoder and attack model is defined as follows:

$$\mathcal{L}_{reweight} = \mathcal{L}'_{BCE} + \mathcal{L}'_{ELBO}, \tag{12}$$

$$\mathcal{L}'_{BCE} = -\sum_{i=1}^{N_{shadow}} \mathbf{w}_{shadow,i}(\mathbf{p}) \cdot \left(y^*_i \log y_{i,1} + (1 - y^*_i) \log y_{i,2}\right),$$

$$\mathcal{L}'_{ELBO} = -\sum_j \sum_i^{N_j} \mathbf{w}_{j,i}(\mathbf{p}) \cdot \mathcal{L}_{ELBO,j,i}, \quad j \in \{shadow, target\},$$

where $\mathbf{w}_{shadow}(\mathbf{p})$ and $\mathbf{w}_{target}(\mathbf{p})$ are the data sample weights for shadow and target recommenders, obtained by applying a linear layer on the current truth-level scores $\mathbf{p}$. We compute the re-weighted and disentangled difference vector $\mathbf{f}^{rew} = \left[\mathbf{f}'^{inv}; \mathbf{f}'^{spe}\right]$ by minimizing $\mathcal{L}_{reweight}$, where $\mathbf{f}'^{inv}$ and $\mathbf{f}'^{spe}$ are the re-weighted invariant and specific vectors. Meanwhile, the trained attack model is able to predict membership labels for the target recommender, i.e., y_{target}. Next, we approximate $\mathbf{f}'$ and y^*_{target} by $\mathbf{f}^{rew}$ and y_{target}, and minimize $\mathcal{L}_{estimate}$ to refine the current truth-level scores $\mathbf{p}$. In this way, $\mathcal{L}_{reweight}$ and $\mathcal{L}_{estimate}$ are optimized in an alternating fashion.

3.5 Learning algorithm

The training process of DL-MIA contains two stages: (i) **Pretraining**. We first pretrain the disentangled encoder and attack model jointly by optimizing $\mathcal{L}_{BCE}$ (Eq. 3) and $\mathcal{L}_{ELBO}$ (Eq. 5). In this stage, the disentangled difference vector $\mathbf{f}^{dis}$ is computed, and inputted into the attack model for learning. (ii) **Alternating training**. After obtaining the disentangled difference vectors, we adopt the alternating training strategy to reduce the estimation bias. Specifically, the re-weighted loss $\mathcal{L}_{reweight}$ and the estimation constraint $\mathcal{L}_{estimate}$ are minimized iteratively. In such manner, the re-weighted difference vector $\mathbf{f}^{rew}$ is derived, and then used by the attack model to conduct membership inference on the target recommender. Sec. A.1 gives the detailed training algorithm of DL-MIA.

Table 1: Statistics of datasets. #Users, #Items, and #Interactions denote the number of users, items, and user-item interactions, respectively.

Dataset	#Users	#Items	#Interactions
MovieLens-1M	6,040	3,706	1,000,209
Amazon Digital Music	840,372	456,992	1,584,082
Amazon Beauty	1,210,271	249,274	2,023,070

4 EXPERIMENTS

Research questions. We aim to answer the following research questions: (RQ1) Does DL-MIA outperform the state-of-the-art attack methods? Is DL-MIA able to generalize to the sequential recommendation? (Sec. 5.1 and 5.2) (RQ2) How does the disentangled encoder and weight estimator contribute to the performance? (Sec. 6.1) (RQ3) What is the influence of the difference vector generator and defense mechanism? (Sec. 6.2 and 6.3) (RQ4) Is DL-MIA able to identify features invariant and specific to shadow and target recommenders, and alleviate the estimation bias? (Sec. 6.4)

Datasets. Following Zhang et al. [53], we evaluate the attack performance on two real-world datasets, MovieLens [14] and Amazon [33]. MovieLens is a widely used benchmark dataset for evaluating collaborative filtering algorithms. We use the version (MovieLens-1M) that includes 1 million user ratings for both general and sequential recommenders. Amazon is a series of datasets, consisting of large corpora of product reviews crawled from Amazon.com. Top-level product categories on Amazon are treated as separate datasets. Similar to Zhang et al. [53], we consider "Digital Music" for general recommenders, while "Beauty" is used for sequential recommenders, since the number of the interacting users per item in "Digital Music" is extremely low (less than 2). Table 1 summarizes the statistics of datasets.

Similar to [53], we further divide each dataset into three disjoint subsets. i.e., a shadow dataset, a target dataset, and a dataset for difference vector generation. We filter the target and shadow datasets to make sure the dataset for difference vector generation contains all the items. Then, target and shadow datasets are both randomly separated into two disjoint parts for members and non-members, respectively. For general recommenders, we remove users with less than 20 interactions. For sequential recommenders, we filter out users and items with less than 5 interaction records.

Recommender systems. Following [53], we evaluate membership inference attacks against three general recommenders: (i) Item-based collaborative filtering (ItemBase) [43], (ii) latent factor model (LFM) [36], and (iii) neural collaborative filtering (NCF) [15]. To investigate the generality of our proposed model, we also implement attack models on three sequential recommendation methods in our experiments, including GRU4Rec [16], Caser [49], and BERT4Rec [48].

Experimental settings. Table 4 shows the notation for our experimental settings. Note that not all possible settings are listed due to space limitations. In the experiments, there are two kinds of combinations (i.e., 2-letter and 4-letter combinations) for experimental settings. For the 2-letter combinations, the first letter indicates the dataset, and the second letter denotes the recommendation algorithm. For example, for general recommenders, "AI" denotes that the recommender is implemented by ItemBase and trained on Amazon Digital Music. For the 4-letter combinations, the first two letters represent the dataset and algorithm used by the shadow recommender, and the last two letters denote the dataset and algorithm used by the target recommender. For instance, for sequential recommenders, "ABMC" means the adversary establishes a shadow recommender using BERT4REC on Amazon Beauty to attack a target recommender using Caser on MovieLens-1M.

Baseline. We compare the proposed DL-MIA with the biased baseline (Biased) [53], which is the first work studying the membership inference attack against recommender systems. Previous MIA methods [9, 29, 35, 42, 46, 52] are not considered in our experiments, since they cannot be directly applied to recommender systems.

Evaluation metric. We adopt the area under the ROC curve (AUC) as the evaluation metric. AUC signifies the probability that the positive sample's score is higher than the negative sample's score, illustrating the classification model's ability to rank samples. For example, if the attack model infers the membership with random guessing, the AUC is close to 0.5.

Implementation details. For the attack model, we build a MLP with 2 hidden layers. And the first layer has 32 units and the second layer has 8 units. We employ the ReLU activation function, and use the softmax function as the output layer. For optimizers, we employ Adam with a learning rate of 0.001 for the disentangled encoder and SGD with a learning rate of 0.01 and a momentum of 0.7 for the attack model. During training, we first pretrain the attack model and the disentangled encoder jointly for 200 epochs. Then, truth-level scores and model parameters are alternatively updated for every 10 epochs. The whole alternative training is conducted for 100 epochs. Following [53], we consider the top 100 recommendations for members, and recommend the most popular items to non-members. Table 5 lists detailed parameter settings.

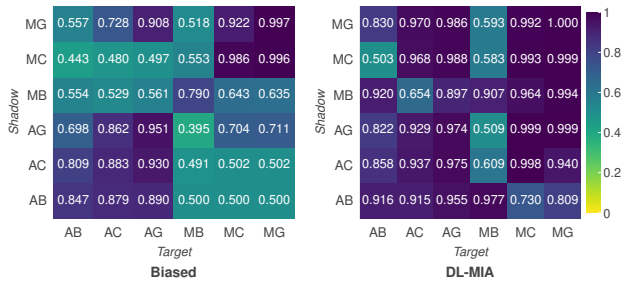
5 EXPERIMENTAL RESULTS

For RQ1, we evaluate the attack performance of our proposed DL-MIA over general and sequential recommender systems.

5.1 Attack performance over general recommenders (RQ1)

Figure 3(a) shows the experimental outcomes for the attack performance over general recommender systems. Based on the experimental results, we have the following observations: (i) Membership inference attack against general recommender systems is challenging, and for the biased baseline, AUC scores are less than 0.7 in most settings (more than 60%). In contrast, our proposed DL-MIA is able to effectively infer the membership of the target recommenders, and AUC scores are over 0.8 for more than 80% experimental settings. (ii) The proposed DL-MIA consistently outperforms the biased baseline in all the settings. For example, for the "MLAI" setting, the AUC score of DL-MIA is 0.980, while that of the biased baseline attack is 0.608. That is, identifying features invariant and specific to recommenders, and computing the truth-level scores for difference vectors substantially enhance the attack performance. (iii) Similar to the conclusions mentioned in [53], with knowledge of the algorithm and dataset distribution used by the target recommender, the adversary is capable of conducting a strong attack, and AUC

(a) General recommenders



(b) Sequential recommenders

Figure 3: Attack performance (AUC) over general recommenders (a) and sequential recommenders (b).

scores of DL-MIA at the back-diagonal in Figure 3(a) are the highest in most cases. In summary, the proposed DL-MIA can effectively infer the membership for the target recommender. Identifying recommender invariant and specific features, as well as considering estimation accuracy of difference vectors, are beneficial for the membership inference attack.

5.2 Attack performance over sequential recommenders (RQ1)

To investigate the generality of DL-MIA, we report the performance of membership inference attacks against sequential recommenders. Based on the results in Figure 3(b), we arrive at the following insights: (i) Even with ordered user historical behaviors, DL-MIA can still accurately calculate the difference vectors, and infer membership. The AUC scores of DL-MIA are over 0.8 for more than 80% settings. (ii) DL-MIA surpasses the biased baseline, and achieves better attack performance over both general and sequential recommenders, demonstrating the effectiveness and strong generalizability of our proposed framework. In summary, the DL-MIA framework cannot only effectively conduct membership inference attacks against general recommenders, but also attains the best AUC scores over sequential recommenders.

6 ANALYSIS

In this section, we take a closer look at DL-MIA to analyze its performance. We examine how the disentangled encoder, and the weight estimator contribute to the performance. The influence of the difference vector generator and defense mechanism is also investigated. In addition, we conduct case studies to study whether DL-MIA is able to recognize recommender invariant and specific features, and mitigate the estimation bias.

Table 2: Ablation studies over general and sequential recommenders. …(β-VAE) and …(FactorVAE) are two model-variants of DL-MIA with widely-used VAE models.

Model	General				Sequential			
	AIMI	ALML	AIML	ALMI	AGMG	MGAG	MGMC	AGMC
DL-MIA	**0.999**	**0.995**	0.993	0.998	**0.999**	**0.986**	0.992	**0.999**
-Reweight	0.553	0.518	0.531	0.526	0.723	0.915	0.936	0.707
Biased [53]	0.525	0.510	0.524	0.510	0.711	0.908	0.922	0.704
…(β-VAE)	0.920	0.993	0.995	0.996	0.999	0.983	0.993	0.996
…(FactorVAE)	0.999	0.987	**0.999**	**0.999**	0.999	0.982	**0.994**	0.999

6.1 Ablation studies (RQ2)

We conduct ablation studies over both general and sequential recommenders. The results are shown in Table 2. AUC scores are adopted here to evaluate the attack performance. When only employing the difference vector generator and attack model, our framework is reduced to the biased baseline. In that case, AUC scores over all the settings suffer a dramatic drop. In the "-Reweight" setting, the disentangled encoder and attack model are trained jointly whereas the alternative training is removed. Compared to the biased baseline, identifying features invariant and specific to shadow and target recommenders considerably alleviates the training data bias, and AUC scores are consistently improved over both general and sequential recommenders. Meanwhile, DL-MIA further enhances the attack performance by reducing the estimation bias of difference vectors. In a nutshell, both the disentangled encoder and weight estimator contribute to the improvements in attack performance.

In Tabe 2, we also consider two model variants, DL-MIA (β-VAE) and DL-MIA (FactorVAE), with two widely-used VAE models β-VAE [17] and FactorVAE [24], respectively. Based on the results in Table 2, we observe that DL-MIA (β-VAE), and DL-MIA (FactorVAE) both achieve a similar attack performance as DL-MIA. That is, even with a different VAE based encoder, our proposed framework is still able to perform effective membership inference.

6.2 Influence of difference vector generator (RQ3)

Table 3 shows the attack performance (AUC) with two kinds of difference vector generators, "MF" and "BERT". "MF" generates the difference vectors by factorizing user-item matrices (explained in Sec. 3.2). "BERT" employs the tiny-sized BERT [12] to embed item descriptions, and takes the [CLS] vectors as item representations. Since some item descriptions are missing, we do not consider experimental settings using the Amazon Digital Music dataset. Based on the results in Table 3, we find that DL-MIA performs better than the biased baseline over both generators, indicating the effectiveness of our framework.

6.3 Influence of defense mechanism (RQ3)

Following Zhang et al. [53], we investigate the influence of defense mechanism, and apply the countermeasure named *Popularity*

Table 3: Influence of the difference vector generator over general and sequential recommenders. "Gen." is short for "Generator."

Gen.	Model	General				Sequential			
		MIMI	MIMN	MLMI	MNML	AGMG	MGAG	MGMC	AGMC
MF	Biased [53]	0.998	0.706	0.931	0.914	0.711	0.908	0.922	0.704
	DL-MIA	**1.000**	**0.959**	**0.997**	**0.987**	**0.999**	**0.986**	**0.992**	**0.999**
BERT	Biased [53]	0.979	0.575	0.505	0.706	0.517	0.584	0.922	0.621
	DL-MIA	**0.980**	**0.830**	**0.971**	**0.770**	**0.873**	**0.863**	**0.979**	**0.870**

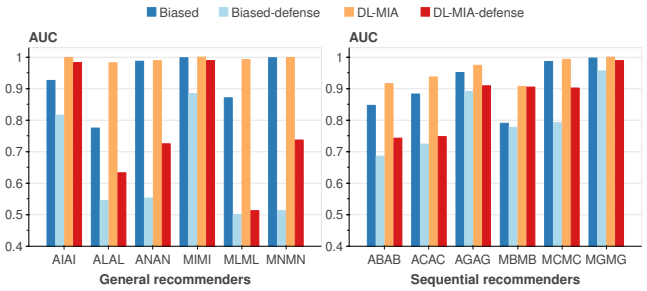


Figure 4: Influence of the defense mechanism. "Biased-defense" and "DL-MIA-defense" denote Biased [53] and DL-MIA with the defense mechanism, respectively.

Randomization to the attack frameworks. Figure 4 shows the attack performance (AUC) before and after deploying the defense mechanism. With the defense mechanism, the attack performance for both the biased baseline and DL-MIA consistently decreases over all the settings. Meanwhile, compared to the biased baseline, our proposed DL-MIA achieves higher AUC scores, and shows a stronger robustness to the countermeasure.

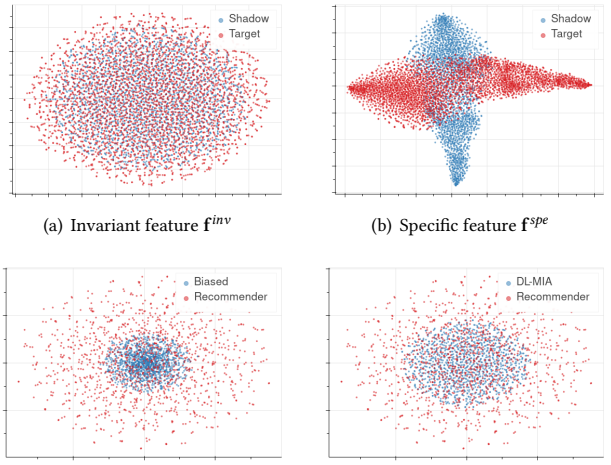
6.4 Case studies (RQ4)

Figure 5 shows visualization results of the "ANML" setting for the general recommender system by t-SNE [50]. Points in Figure 5(a) and Figure 5(b) stand for the recommender invariant and specific features, respectively. We can see that invariant features from the shadow recommender (red) and target recommender (blue) are distributed similarly, whereas specific features are scattered divergently. That is, by employing the disentangled encoder, DL-MIA is able to mitigate the gap between recommenders.

In addition, Figure 5(c) and Figure 5(d) demonstrate the visualization results of difference vectors before and after debiasing, respectively. Based on the results, we conclude that DL-MIA effectively reduce the gap between difference vectors generated by the attack model (red) and recommender (blue), and alleviate the influence of the estimation bias.

7 CONCLUSION AND FUTURE WORK

In this paper, we investigate the membership inference attack against recommender systems. Previously published methods faces two challenging problems: (i) the biased attack model training



(a) Invariant feature $\mathbf{f}^{inv}$ (b) Specific feature $\mathbf{f}^{spe}$

(c) Biased difference vector (d) Debiased difference vector

Figure 5: Visualization results of the "ANML" setting for the general recommender.

caused by the gap between target and shadow recommenders, (ii) and inaccurate estimation of difference vectors since hidden states in recommenders are inaccessible. To handle these problems, we propose a novel framework named DL-MIA. To mitigate the gap between target and shadow recommenders, the VAE based encoder is devised to identify recommender invariant and specific features. And to alleviate the estimation bias, the weight estimator is constructed, and truth-level scores for difference vectors are computed to facilitate the model update. We evaluate DL-MIA against both general recommenders and sequential recommenders on two real-world datasets. Experimental results demonstrate that the DL-MIA framework is able to effectively alleviate training and estimation biases, and shows a strong generality.

In future work, we intend to incorporate more kinds of disentangled methods and explore other types of biases in the membership inference attack against recommender systems.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China with grant No. 2020YFB1406704, the Natural Science Foundation of China (61902219, 61972234, 62072279, 62102234), the Natural Science Foundation of Shandong Province (ZR2021QF129), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), Shandong University multidisciplinary research and innovation team of young scholars (No. 2020QNQT017), Meituan, the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing Fair Ranking Schemes. In *SIGMOD*. 1259–1276.

[2] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. 2016. Membership Privacy in MicroRNA-based Studies. In *CCS*. 319–330.

[3] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-Aware Recommendation with Private-Attribute Protection using Adversarial Learning. In *WSDM*. 34–42.

[4] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *USENIX*. 267–284.

[5] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *CCS*. 343–362.

[6] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *CoRR* abs/2010.03240 (2020).

[7] Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. Variational Sequential Labelers for Semi-Supervised Learning. In *EMNLP*. 215–226.

[8] Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A Multi-Task Approach for Disentangling Syntax and Semantics in Sentence Representations. In *NAACL-HLT*. 2453–2464.

[9] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. 2021. Label-Only Membership Inference Attacks. In *ICML*, Vol. 139. 1964–1974.

[10] Sanghyeok Chu, Dongwan Kim, and Bohyung Han. 2021. Learning Debiased and Disentangled Representations for Semantic Segmentation. In *NeurIPS*. 8355–8366.

[11] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. 2018. Hyperspherical Variational Auto-Encoders. In *UAI*. 856–865.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[13] Inken Hagestedt, Yang Zhang, Mathias Humbert, Pascal Berrang, Haixu Tang, XiaoFeng Wang, and Michael Backes. 2019. MBeacon: Privacy-Preserving Beacons for DNA Methylation Data. In *NDSS*.

[14] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

[17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.

[18] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS genetics*. 4, 8 (2008), e1000167.

[19] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *CCS*. 259–274.

[20] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: factored item similarity models for top-N recommender systems. In *KDD*. 659–667.

[21] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian J. McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *ICDM*. 207–216.

[22] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. 197–206.

[23] Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *RecSys*. 233–240.

[24] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by Factorising. In *ICML*, Vol. 80. 2654–2663.

[25] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*. 426–434.

[26] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. 77–118.

[27] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*. 42, 8 (2009), 30–37.

[28] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *NeurIPS*. 4066–4076.

[29] Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In *CCS*. 880–895.

[30] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the Volume: Preference Bias Amplification in Collaborative Recommendation. In *RecSys*, Vol. 2440.

[31] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Comput.* 7, 1 (2003), 76–80.

[32] Benjamin M. Marlin, Richard S. Zemel, Sam T. Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *UAI*. 267–275.

[33] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. 43–52.

[34] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *CCS*. 634–646.

[35] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *SP*. 1021–1035.

[36] Huseyin Polat and Wenliang Du. 2005. SVD-based collaborative filtering with privacy. In *SAC*. 791–795.

[37] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2018. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In *NDSS*.

[38] Chen Qian, Fuli Feng, Lijie Wen, and Tat-Seng Chua. 2021. Conceptualized and Contextualized Gaussian Embedding. In *AAAI*. 13683–13691.

[39] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*. 811–820.

[40] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *SIGIR*. 309–318.

[41] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NeurIPS*. 1257–1264.

[42] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *NDSS*.

[43] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[44] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *ICML*, Vol. 48. 1670–1679.

[45] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *WWW*. 111–112.

[46] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *SP*. 3–18.

[47] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *KDD*. 713–722.

[48] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*. 1441–1450.

[49] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. 565–573.

[50] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* 9, 86 (2008), 2579–2605.

[51] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating Selection Biases in Recommender Systems with a Few Unbiased Ratings. In *WSDM*. 427–435.

[52] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *CSF*. 268–282.

[53] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhumin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership Inference Attacks Against Recommender Systems. In *CCS*. 864–879.

[54] Chunting Zhou and Graham Neubig. 2017. Multi-space Variational Encoder-Decoders for Semi-supervised Labeled Sequence Transduction. In *ACL*. 310–320.

[55] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. In *AAAI*. 5941–5948.

Table 4: Notation for different settings. "Rec." is short for "Recommender." "∗" stands for any recommendation algorithm or dataset.

Rec.	Notation	Description
General	A∗	Trained on the Amazon Digital Music dataset.
	M∗	Trained on the MovieLens-1M dataset.
	∗I	Implemented by the ItemBase algorithm.
	∗L	Implemented by the LFM algorithm.
	∗N	Implemented by the NCF algorithm.
	AI	The recommender is implemented by Item algorithm on the Amazon Digital Music dataset.
	AIMN	The shadow recommender is implemented by the ItemBase algorithm on the Amazon Digital Music dataset, and the target recommender is implemented by NCF algorithm on the MovieLens-1M dataset.
Sequential	A∗	Trained on the Amazon Beauty datset.
	M∗	Trained on the MovieLens-1M dataset.
	∗B	Implemented by BERT4REC algorithm.
	∗C	Implemented by Caser algorithm.
	∗G	Implemented by GRU4REC algorithm.
	AB	The recommender is implemented by BERT4REC on the Amazon Beauty dataset.
	ABMC	The shadow recommender is implemented by BERT4REC on the Amazon Beauty dataset, and the target recommender is implemented by Caser on the MovieLens-1M dataset.

Table 5: Parameter settings of different recommender systems.

Baseline	Settings
ItemBase	–
LFM	Embed.-size=100, SGD optimizer, learning-rate=0.01
NCF	Embed.-size=8, batch-size=256, Adam optimizer, hidden-size=64, 32, 16, learning-rate=0.001
BERT4REC	batch-size=128, Adam optimizer, dropout=0.1 hidden-size=256, learning-rate=0.001
Caser	Embed.-size=50, batch-size=512, Adam optimizer, dropout=0.5 learning-rate=0.001
GRU4REC	batch-size=50, Adagrad optimizer, dropout=0.5, hidden-size=100, learning-rate=0.01, momentum=0

A APPENDIX

A.1 Learning algorithm of DL-MIA

Algorithm 1 gives the detailed learning algorithm of DL-MIA. Specifically, given the target recommender, we first establish the shadow recommender $\mathcal{M}_{shadow}$, calculate the difference vector $\mathbf{f}^{diff}$ by MF, and initialize model parameter Θ for the disentangled encoder and attack model (line 1–3). Then, to mitigate the gap between the shadow and target recommenders, we train the disentangled encoder and attack model by jointly optimizing $\mathcal{L}_{BCE}$ and $\mathcal{L}_{ELBO}$ (line 4–7). In this way, the disentangled difference vector $\mathbf{f}^{dis}$ is computed. Next, to further reduce the influence of the estimation bias, an alternating training strategy is adopted. By determining data sample weights $\mathbf{w}$ using the current $\mathbf{p}$, we are able to minimize the reweighted loss $\mathcal{L}_{reweight}$, and obtain the reweighted difference vector $\mathbf{f}^{rew}$ (line 10–15). With input of $\mathbf{f}^{rew}$ and y_{target}, the current p can be refined using the estimation constraint $\mathcal{L}_{estimation}$ (line 16–19). During the alternating training process, $\mathcal{L}_{reweight}$ and $\mathcal{L}_{estimation}$ are optimized iteratively (line 9–20).

Algorithm 1 Training algorithm of DL-MIA.

Require: The trained shadow recommender $\mathcal{M}_{shadow}$; the difference vector $\mathbf{f}^{diff}$ from the generator; randomly initialized truth-level scores $\mathbf{p}$; the number of inner-loop epochs $epoch_{in}$ and outer-loop epochs $epoch_{out}$ for the alternating training; the number of epochs for pretraining $epoch_{pre}$; parameters Θ for the disentangled encoder and attack model.

Ensure: The disentangled and reweighted difference vector $\mathbf{f}^{rew}$ and trained attack model $\mathcal{A}$;

1: Establish the shadow recommender $\mathcal{M}_{shadow}$;
2: Calculate the difference vector $\mathbf{f}^{diff}$ (Eq. 2);
3: Randomly initialize model parameters Θ;
4: **while** $i \leq epoch_{pre}$ **do**
5: Calculate the disentangled difference vector $\mathbf{f}^{dis}$;
6: Input $\mathbf{f}^{dis}$ into the attack model $\mathcal{A}$ for predicting y_{shadow};
7: Update Θ by jointly optimizing $\mathcal{L}_{BCE}$ and $\mathcal{L}_{ELBO}$ (Eq. 3 and 5);
8: **end while**
9: **while** $i \leq epoch_{out}$ **do**
10: Compute the data sample weights $\mathbf{w}$ using the current $\mathbf{p}$;
11: **while** $j \leq epoch_{in}$ **do**
12: Calculate the reweighted feature vector $\mathbf{f}^{rew}$;
13: Input $\mathbf{f}^{rew}$ into $\mathcal{A}$ for predicting y_{target} and y_{shadow};
14: Update Θ by minizing $\mathcal{L}_{reweight}$ (Eq. 12);
15: **end while**
16: Input $\mathbf{f}^{rew}$ and y_{target} into the weight estimator;
17: **while** $k \leq epoch_{in}$ **do**
18: Refine the current truth-level scores $\mathbf{p}$ by optimizing $\mathcal{L}_{estimation}$ (Eq. 9 and 11);
19: **end while**
20: **end while**

A.2 Notation

Table 4 shows the notation we use for different experimental settings.

A.3 Implementation details

Table 5 demonstrate the parameter settings of different recommenders in our experiments. Note that, we do not give the parameter setting of ItemBase [43] since it is based on the statistical method.

A.4 Reproducibility

To facilitate the reproducibility of the results reported in this work, the code and data used in this work is available at https://github.com/WZH-NLP/DL-MIA-KDD-2022.