Graph Embedding for Recommendation against Attribute Inference Attacks

Shijie Zhang
The University of Queensland
shijie.zhang@uq.edu.au

Hongzhi Yin*
The University of Queensland
h.yin1@uq.edu.au

Tong Chen
The University of Queensland
tong.chen@uq.edu.au

Zi Huang
The University of Queensland
huang@itee@uq.edu.au

Lizhen Cui
Shandong University
clz@sdu.edu.cn

Xiangliang Zhang
King Abdullah University of Science and Technology
xiangliang.zhang@kaust.edu.sa

ABSTRACT

In recent years, recommender systems play a pivotal role in helping users identify the most suitable items that satisfy personal preferences. As user-item interactions can be naturally modelled as graph-structured data, variants of graph convolutional networks (GCNs) have become a well-established building block in the latest recommenders. Due to the wide utilization of sensitive user profile data, existing recommendation paradigms are likely to expose users to the threat of privacy breach, and GCN-based recommenders are no exception. Apart from the leakage of raw user data, the fragility of current recommenders under inference attacks offers malicious attackers a backdoor to estimate users' private attributes via their behavioral footprints and the recommendation results. However, little attention has been paid to developing recommender systems that can defend such attribute inference attacks, and existing works achieve attack resistance by either sacrificing considerable recommendation accuracy or only covering specific attack models or protected information. In our paper, we propose GERAI, a novel differentially private graph convolutional network to address such limitations. Specifically, in GERAI, we bind the information perturbation mechanism in differential privacy with the recommendation capability of graph convolutional networks. Furthermore, based on local differential privacy and functional mechanism, we innovatively devise a dual-stage encryption paradigm to simultaneously enforce privacy guarantee on users' sensitive features and the model optimization process. Extensive experiments show the superiority of GERAI in terms of its resistance to attribute inference attacks and recommendation effectiveness.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**.

*Corresponding author; contributing equally with the first author.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449813

KEYWORDS

Privacy-preserving Recommender System; Attribute Inference Attacks; Deep Learning; Differential Privacy

ACM Reference Format:
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 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3442381.3449813

1 INTRODUCTION

With the explosive growth of e-commerce, consumers are shopping with online platforms more frequently [10, 20, 54]. As an effective solution to information overload, recommender systems automatically discover the most relevant items or services for each user and thus improve both the user experience and business revenue. For this reason, recommender systems have become an indispensable part in our contemporary lives.

Latent factor models like matrix factorization [35] are typical collaborative filtering-based recommendations, which infer user-item interactions via learned latent user/item representations. Because user-item interactions can be conveniently formulated as graph-structured data, graph embedding-based recommenders [42, 52, 59] are highly effective in uncovering users' subtle preferences toward items. As deep neural networks demonstrate superior capability of representation learning in various machine learning tasks, deep recommendation models, especially those derived from graph convolutional networks (GCNs) [49, 51, 55, 56] have recently become one of the most prominent techniques in this field.

To enhance the recommendation performance, especially for fresh (i.e., cold-start) customers, it is a common practice to incorporate side information (a.k.a. features or contexts) [2, 47, 58] about users. During user registration, some service providers even start persuading users to complete questionnaires about personal demographics to facilitate user profiling. However, the utilization of user data containing personal information often sparks serious privacy concerns. A 2018 survey [26] showed that more than 80% US Internet users were concerned about how their personal data is being used on Facebook; and among Facebook users sharing less content on social media, 47% reported that privacy issue was the main concern. Consequently, with the growing public awareness on privacy, a dilemma is presented to e-commerce platforms: either they proceed with such sensitive data acquisition process despite

the high risk on privacy breach, or they allow users not to disclose their sensitive attributes but provide compromised recommendation performance as a result. In that sense, a sound privacy guarantee on the user side is highly desirable, which avoids uploading the unencrypted raw user features to a recommender system. Furthermore, according to the example that Apple is now telling users their personal data is protected before being shared for analytics, it also helps increase users' willingness to share their sensitive data.

Meanwhile, a more critical privacy issue comes from the fact that users' sensitive attributes can still be disclosed purely based on how they behave. Regardless of the availability of features, recommenders learn explicit or latent profiles that reflect users' preferences based on her/his behavioral footprints (e.g., previous ratings and reviews), and produce personalized recommendations with the constructed profiles [40]. However, many early studies have shown that even a user's personal information can be accurately inferred via her/his interaction history [5, 29, 50]. Such personal information includes age, gender, political orientation, health, financial status etc. and are highly confidential. Furthermore, the inferred attributes can be utilized to link users across multiple sites and break anonymity [16, 44]. For example, [36] successfully deanonymizes Netflix users using the public IMDb user profiles. Due to the open-access nature of many platforms (e.g., Yelp and Amazon), users' behavioral trajectories can be easily captured by a malicious third-party, leading to catastrophic leakage of inferred user attributes. This is known as the attribute inference attack [17], where the malicious attackers can be cyber criminals, data brokers, advertisers, etc. By proving that even a person's racial information and sexual orientation can be precisely predicted from merely the "like" behaviors on Facebook, Kosinski et al. [29] demonstrated that users' preference signals are highly vulnerable to attribute inference attacks. This is especially alarming for many GCN-based recommenders, since user representations are usually formed by aggregating information from her/his interacted items. Moreover, the personalized recommendation results can also be utilized by attackers since they are strong reflections on users' preferences and are increasingly accessible via services like friend activity tracing (e.g., Spotify) and group recommendation [54]. Hence, this motivates us to design a secure recommender system that stays robust against attribute inference attacks.

In GCN-based recommenders, graphs are constructed by linking user and item nodes via their interactions. However, though existing GCNs are advantageous in binding a node's own features and its high-order connectivity with other nodes into an expressive representation, they exhibit very little consideration on user privacy. In fact, the field of privacy-preserving recommender systems that are resistant to attribute inference attacks is far from its maturity. [6, 14, 37, 39] have applied cryptography algorithms to the recommendation models, but the computational cost of encryption is too high to support real-world deployment. Recently, the notion of differential privacy (DP) has become a well-established approach for protecting the confidentiality of personal data. Essentially, DP works by adding noise to each data instance (i.e., perturbation), thus masking the original information in the data. In the context of both recommendation and graph embedding, there has also been attempts to adopt DP to perturb the output of matrix factorization algorithms [4, 33, 53]. Unfortunately, these approaches are designed to only prevent membership attacks which infer users' real ratings in the dataset, and are unable to provide a higher level of protection on users' sensitive information against inference attacks. A recent work [3] systematically investigates the problem of developing and evaluating recommender systems under the attribute inference attack setting. Their proposed model RAP [3] utilizes an adversarial learning paradigm where a personalized recommendation model and an attribute inference attack model are trained against each other, hence the attackers are more likely to fail when inferring user attributes from interaction records. However, it suffers from two major limitations. Firstly, as the design of RAP requires a pre-specified and fixed attribute inference model, its resistance to any arbitrary attacker is unguaranteed given the unpredictability of the inference model that an attacker may choose. Secondly, though RAP assumes the existence of users' sensitive attributes, it only treats them as ground-truth labels for training the inference model, and does not incorporate such important side information for recommendation. This design not only fails to ease users' privacy concerns on submitting their original attributes, but also greatly hinders the model's ability to securely utilize user features to achieve more accurate recommendation results.

To this end, we address a largely overlooked defect of existing GCN-based recommenders, i.e., protecting users' private attributes from attribute inference attacks. Meanwhile, unlike existing inference-resistant recommenders, we would like the model to take advantage of user information for accurate recommendation without exerting privacy breach. In this paper, we subsume the GCN-based recommender under the differential privacy (DP) constraint, and propose a novel privacy-preserving recommender GERAI, namely Graph Embedding for Recommendation against Attribute Inference Attacks. In GERAI, we build its recommendation module upon the state-of-the-art inductive GCNs [11, 21, 28] to jointly exploit the user-item interactions and the rich side information of users. To achieve optimal privacy strength, we propose a novel dual-stage perturbation paradigm with DP. Firstly, at the input stage, GERAI performs perturbation on the raw user features. On one hand, this offers users a privacy guarantee while sharing their sensitive data. On the other hand, the perturbed user features will make the generated recommendations less dependent on a user's true attributes, making it harder to infer those attributes via recommendation results. Specifically, we introduce local differential privacy (LDP) for feature perturbation, where each individual's original feature vector is transformed into a noisy version before being processed by the recommendation module. We further demonstrate that the perturbed input data satisfies the LDP constraint while retaining adequate utility for the recommender to learn the subtle user preferences. Secondly, we enforce DP on the optimization stage of GERAI so that the recommendation results are less likely to reveal a user's attributes and preferences [3, 4, 33] in the inference attack. To achieve this, we innovatively resort to the functional mechanism [57] that allows to enforce DP by perturbing the loss function in the learning process. Different from methods that applies perturbation on recommendation results [4], by perturbing the loss function, GERAI defends the inference attack without setting obstacles for learning meaningful associations between user profiles and recommended items.

Overall, we summarize our contributions in the following:

- We address the increasing privacy concerns in the recommendation context, and propose a novel solution GERAI, namely differentially private graph convolutional network to protect users' sensitive data against attribute inference attacks and provide high-quality recommendations at the same time.
- Our proposed GERAI innovatively incorporates differential privacy with a dual-stage perturbation strategy for both the input features and the optimization process. As such, GERAI assures user privacy and offers better recommendation effectiveness than existing privacy-preserving recommenders.
- We conduct extensive experiments to evaluate the performance of GERAI on real-world data. Comparisons with state-of-the-art baselines show that GERAI provides a better privacy guarantee with less compromise on the recommendation accuracy.

2 PRELIMINARIES

In this section, we first revisit the definitions of differential privacy and then formally define our problem. Note that in the description below, all vectors and matrices are respectively denoted wiht bold lowercase and bold uppercase letters, and all sets are written in calligraphic uppercase letters.

Differential Privacy. Differential privacy (DP) is a strong mathematical guarantee of privacy in the context of machine learning tasks. DP was first introduced by [13] and it aims to preclude adversarial inference on any raw input data from a model's output. Given a privacy coefficient $\epsilon > 0$, the $\epsilon-$differential privacy ($\epsilon-$DP) is defined as follows:

Definition 2.1. ($\epsilon-$Differential Privacy) For a randomized function (e.g., a perturbation algorithm or machine learning model) $f(\cdot)$ that takes a dataset as its input, it satisfies $\epsilon-$DP if:

$$Pr[f(\mathcal{D}) \in O] \leq exp(\epsilon) Pr[f(\mathcal{D}') \in O], \tag{1}$$

where $Pr[\cdot]$ represents probability, $\mathcal{D}$ and $\mathcal{D}^{'}$ are any two datasets differing on only one data instance, and O denotes all subsets of possible output values that $f(\cdot)$ produces. If O is continuous, then the probability term can be replaced by a probability density function. Eq.(1) implies that the probability of generating the model output with $\mathcal{D}$ is at most $exp(\epsilon)$ times smaller than with $\mathcal{D}'$. That is, $f(\cdot)$ should not overly depend on any individual data instance, providing each instance roughly the same privacy. As a common practice for privacy protection, each individual user's personal data can be perturbed by adding controlled noise before it is fed into $f(\cdot)$. In this case, the data owned by every user is regarded as a singleton dataset, and we require the function $f(\cdot)$ to provide differential privacy when such a singleton database is given as the input. Specifically, this is termed as $\epsilon-$local differential privacy ($\epsilon-$LDP):

Definition 2.1. ($\epsilon-$Local Differential Privacy) A randomized function $f(\cdot)$ satisfies $\epsilon-$LDP if and only if for any two users' data t and t', we have:

$$Pr[f(t) = t^*] \leq exp(\epsilon) \cdot Pr[f(t') = t^*] \tag{2}$$

where t^* denotes the output of $f(\cdot)$. The lower ϵ provides stronger privacy but may result in lower accuracy of a trained machine learning model as each user's data is heavily perturbed. Hence, ϵ is also called the privacy budget that controls the trade-off between privacy and utility in DP. With the security guarantee from DP, an external attacker model cannot infer which user's data is used to produce the output t^* (e.g., the recommendation results) with high confidence.

Privacy-Preserving Recommender System. Let $\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E})$ denote a weighted bipartite graph. $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ are the sets of users and items. A weighted edge $(u, v, r_{uv}) \in \mathcal{E}$ means that user u has rated item v, with weight r_{uv} as 1. We use $\mathcal{N}(u)$ to denote the set of items rated by u and $\mathcal{N}(v)$ to denote all users who have rated item v. Following [3], for each user u we construct a dense input vector $\mathbf{x}_u \in \mathbb{R}^{d_0}$ with each element representing either a sensitive attribute $s \in \mathcal{S}$ or a pre-defined statistical feature $s \in \mathcal{S}'$ of u. All categorical features are represented by one-hot encodings in $\mathbf{x}_u$, while all numerical features are further normalized into $[-1, 1]$. We define the target of a privacy-preserving recommender system below.

Problem 1. Given the weighted graph $\mathcal{G}$ and user feature vectors $\{\mathbf{x}_u | u \in \mathcal{U}\}$, we aim to learn a privacy-preserving recommender system that can recommend K products of interest to each user, while any malicious attacker model cannot accurately infer users' sensitive attributes (i.e., gender, occupation and age in our case) from the users' interaction data including both the users' historical ratings and current recommendation results. It is worth noting that our goal is to protect users against a malicious attacker, but not against the recommender system that is trusted.

3 GCN-BASED RECOMMENDATION MODULE

As we aim to address the privacy concerns in GCN-based recommendation models, in this work we build our base recommender upon GCNs [21, 28]. A recommender, at its core, learns vector representations (a.k.a. embeddings) of both users and items based on their historical interactions, then a user's interest on each item can be easily inferred by measuring the user-item similarity in the latent vector space. When performing recommendation on the graph-structured data, owing to the ability to preserve a graphs topological structure, GCNs can produce highly expressive user and item embeddings for recommendation. Given a weighted graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E})$, users and items are two types of nodes connected by observed links. Then, for each node, GCN computes its embedding by iteratively aggregating information from its local neighbors, where all node embeddings are optimized for predicting the affinity of each user-item pair for personalized ranking.

We first introduce our recommendation module from the user side. For each user u, the information $\mathcal{I}(u)$ passed into u comes from the user's first-order neighbors, i.e., items rated by u:

$$\begin{aligned} \mathcal{I}(u) &= \{\mathbf{m}_v | v \in \mathcal{N}(u)\} \cup \{\mathbf{m}_u\} \\ &= \{MLP(\mathbf{z}_v) | v \in \mathcal{N}(u)\} \cup \{MLP(\mathbf{z}_u)\}, \end{aligned} \tag{3}$$

where $MLP(\cdot)$ is a multi-layer perceptron, $\mathbf{m}_u/\mathbf{m}_v$ are the messages from user/item nodes, and $\mathbf{z}_u, \mathbf{z}_v \in \mathbb{R}^d$ respectively denote the learnable latent embeddings of user u and item v. Note that $\mathbf{z}_u, \mathbf{z}_v$ can be initialized as follows:

$$\mathbf{z}_u = \mathbf{E}_{\mathcal{U}} \mathbf{x}_u, \ \mathbf{z}_v = \mathbf{E}_{\mathcal{V}} \mathbf{x}_v, \tag{4}$$

where $\mathbf{x}_u \in \mathbb{R}^{d_0}$ is user u's raw feature vector and $\mathbf{E}_{\mathcal{U}} \in \mathbb{R}^{d \times d_0}$ is the user embedding matrix. $\mathbf{x}_v \in \mathbb{R}^{d_1}$ and $\mathbf{E}_{\mathcal{V}} \in \mathbb{R}^{d \times d_1}$ are respectively the item feature vector and embedding matrix. To

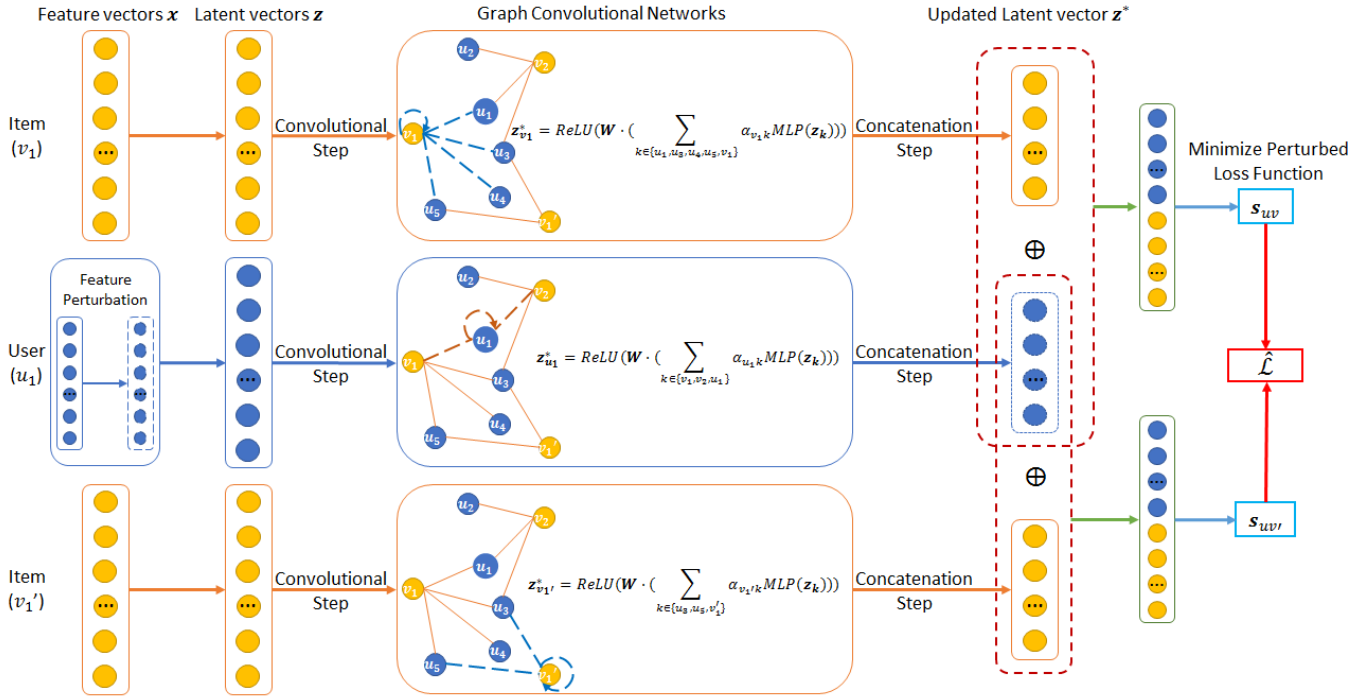


Figure 1: The overview of GERAI

ensure our model's generalizability, we formulate $\mathbf{x}_v$ as an item's one-hot encoding as we do not assume the availability of item features. Then, an aggregation operation is performed to merge all information in $\mathcal{I}(u)$, thus forming an updated user embedding $\mathbf{z}_u^*$:

$$\mathbf{z}_u^* = ReLU(\mathbf{W} \cdot Aggregate(\mathcal{I}(u)) + \mathbf{b}), \tag{5}$$

where $Aggregate(\cdot)$ is the aggregation function and $ReLU(\cdot)$ denotes the rectified linear unit for nonlinearity, and $\mathbf{W}$ and $\mathbf{b}$ are learnable weight matrix and bias vector. Motivated by the effectiveness of attention mechanism [45] in graph representation learning, we quantify the varied contributions of each element in $\mathcal{I}(u)$ to embedding $\mathbf{z}_u^*$ by assigning each neighbour node a different weight. Formally, we define $Aggregate(\mathcal{I}(u))$ as:

$$Aggregate(\mathcal{I}(u)) = \sum_{k \in \mathcal{N}(u) \cup \{u\}} \alpha_{uk} \mathbf{m}_k, \tag{6}$$

where α_{uk} denotes the attention weight implying the importance of message $\mathbf{m}_k \in \mathcal{I}(u)$ to user node u during aggregation. Specifically, to compute α_{uk}, we first calculate an attention score a_{uk} via the following attention network:

$$a_{uk} = \mathbf{w}_2^\top \cdot \sigma(\mathbf{W}_1(\mathbf{m}_k \oplus \mathbf{z}_u) + \mathbf{b}_1) + b_2, \tag{7}$$

where $\oplus$ represents the concatenation of two vectors. Afterwards, each final attention weight α_{uk} is computed by normalizing all the attentive scores using softmax:

$$\alpha_{uk} = \frac{\exp(a_{uk})}{\sum_{k' \in \mathcal{N}(u) \cup \{u\}} \exp(a_{uk'})}. \tag{8}$$

Likewise, on the item side, we repeat the message passing scheme by aggregating the information from an item's interacted users in $\mathcal{N}(v)$ to learn the item embedding $\mathbf{z}_v^*$:

$$\mathbf{z}_v^* = ReLU(\mathbf{W} \cdot Aggregate(\mathcal{I}(v)) + \mathbf{b}), \tag{9}$$

where $\mathcal{I}(v) = \{MLP(\mathbf{z}_u) | u \in \mathcal{N}(v)\} \cup \{MLP(\mathbf{z}_v)\}$. Note that the same network structure and trainable parameters are shared in the computation of both user and item embeddings.

To train our model for top-K recommendation, we leverage the pairwise Bayesian personalized ranking (BPR) loss [41] to learn model parameters. To facilitate personalized ranking, we firstly generate a ranking score s_{uv} for an arbitrary user-item tuple (u, v):

$$\begin{aligned} \mathbf{q}_{uv} &= ReLU(\mathbf{W}_3(\mathbf{z}_u^* \oplus \mathbf{z}_v^*) + \mathbf{b}_3), \\ s_{uv} &= \mathbf{h}^\top \mathbf{q}_{uv}, \end{aligned} \tag{10}$$

where $\mathbf{h} \in \mathbb{R}^d$ is the projection weight. Intuitively, BPR optimizes ranking performance by comparing two ranking scores s_{uv}, $s_{uv'}$ for user u on items v and v'. In each training case (u, v, v'), v is the positive item sampled from $\mathcal{E}$, while v' is the negative item having $r_{uv'} \notin \mathcal{E}$. Then, BPR encourages that v should have a higher ranking score than v' by enforcing:

$$\mathcal{L} = \sum_{(u,v,v') \in \mathcal{D}} -\log \sigma(s_{uv} - s_{uv'}) + \gamma ||\Theta||^2, \tag{11}$$

where $\mathcal{D}$ is the training set, $\sigma(\cdot)$ is the sigmoid function, Θ denotes parameters in the GCN-based recommendation module, and γ is the L2-regularization coefficient.

Algorithm 1: Perturbing 1-Dimensional Numerical Data with Piecewise Mechanism

Input: A single numerical feature $x \in [-1, 1]$ and coefficient $\epsilon^{\triangleright}$
Output: Perturbed feature $\widehat{x} \in [-C, C]$
Uniformly sample ξ from $[0, 1]$;
if $\xi < \frac{\exp(\frac{\epsilon^{\triangleright}}{2})}{\exp(\frac{\epsilon^{\triangleright}}{2})+1}$ **then**
  Uniformly sample $\widehat{x}$ from $[\ell(x), \pi(x)]$;
else
  Uniformly sample $\widehat{x}$ from $[-C, \ell(x)) \cup (\pi(x), C]$;
end
return $\widehat{x}$

Algorithm 2: Perturbing Multidimensional Data with Numerical and Categorical Features

Input: Feature vector $\mathbf{x}_u = \mathbf{x}_{(1)} \oplus \mathbf{x}_{(2)} \oplus \cdots \oplus \mathbf{x}_{(d')} \in \mathbb{R}^{d_0}$ and coefficient $\epsilon^{\triangleright}$
Output: Perturbed feature vector $\widehat{\mathbf{x}}_u$
$\widehat{\mathbf{x}}_u = \widehat{\mathbf{x}}_{(1)} \oplus \widehat{\mathbf{x}}_{(2)} \oplus \cdots \oplus \widehat{\mathbf{x}}_{(d')} \leftarrow \{0\}^{d_0}$;
$\mathcal{A} \leftarrow \zeta$ different values uniformly sampled from $\{1, 2, ..., d'\}$;
for *each feature index* $i \in \mathcal{A}$ **do**
  if $\boldsymbol{x}_{(i)}$ *is a numerical feature* **then**
    $\widehat{\mathbf{x}}_{(i)} \leftarrow$ Execute Algorithm 1 with $x = \mathbf{x}_i$ and $\epsilon^{\triangleright} = \frac{\epsilon^{\triangleright}}{\zeta}$;
    $\widehat{\mathbf{x}}_{(i)} \leftarrow \frac{d'}{\zeta}\widehat{\mathbf{x}}_{(i)}$;
  else
    fetch categorical feature i's one-hot encoding from $\mathbf{x}_u$, denoted by $\mathbf{c} \leftarrow \mathbf{x}_{(i)} = [0, \cdots, 0, 1, 0, \cdots, 0]$;
    for *each element* $c \in \mathbf{c}$ **do**
      Draw c' from $\{0, 1\}$ with
      $$Pr[c' = 1] = \begin{cases} 0.5, & \text{if } c' = 1 \\ \frac{1}{\exp(\frac{\epsilon^{\triangleright}}{\zeta})+1}, & \text{if } c = 0. \end{cases};$$
      $c \leftarrow c'$;
    end
    $\widehat{\mathbf{x}}_{(i)} \leftarrow \mathbf{c}$;
  end
end
return $\widehat{\mathbf{x}}_u$

4 GERAI: GRAPH EMBEDDING FOR RECOMMENDATION AGAINST ATTRIBUTE INFERENCE

In this section, we formally present the design of GERAI, a recommendation model that can defend attribute inference attacks via a novel dual-stage differential privacy constraint. Figure 1 depicts the workflow of GERAI, where two important perturbation operations take place at both the input stage for user features and the optimization stage for the loss function. The first step is to achieve $\epsilon^{\triangleright}$−local differential privacy ($\epsilon^{\triangleright}$−LDP) by directly adding noise to users' raw feature vectors $\mathbf{x}_u$ used for learning user embeddings, which can avoid exposing users' sensitive data to an unsecured cyber environment during upload, while providing the GCN-based recommender with side information for learning expressive user representations. Then, to prevent GERAI from generating recommendation results that can reveal users' sensitive attributes, we further enforce ϵ−DP in the optimization stage by perturbing its loss function $\mathcal{L}$. However, this is a non-trivial task as it requires to calculate the privacy sensitivity of $\mathcal{L}$, which involves analyzing the complex relationship between the input data and learnable parameters. Hence, we propose a novel solution by deriving a polynomial approximation $\widetilde{\mathcal{L}}$ of the original BPR loss $\mathcal{L}$, so as to support sensitivity calculation and perform perturbation on $\widetilde{\mathcal{L}}$ to facilitate differentially private training of GERAI. Notably, to distinguish the DP constraints in two stages, we denote $\epsilon^{\triangleright}$ as local privacy budget and ϵ as global privacy budget, respectively.

4.1 User Feature Perturbation at Input Stage

At the input level, the feature vector $\mathbf{x}_u$ of each user u is perturbed before being fed into the recommender module. This helps address users' privacy concerns on sharing their personal attributes and keep them confidential during the upload process. Furthermore, as we will show in Section 5.7, perturbing user features contributes to defending attribute inference attacks as the recommendation results are no longer based on the actual attributes. Then, instead of the original $\mathbf{x}_u$, the perturbed data $\widehat{\mathbf{x}}_u$ will be used for the recommendation purpose. To achieve this, we treat numerical and categorical features separately, as these two types of data will require different perturbation strategies. Firstly, for numerical data, perturbation is performed based on a randomized encryption mechanism named piecewise mechanism (PM) [46]. Algorithm 1 shows the PM-based perturbation for each scalar numerical feature $x \in \mathbf{x}_u$. In PM, the original feature $x \in [-1, 1]$ will be transformed into a perturbed value $\widehat{x} \in [-C, C]$, with C defined as follows:

$$C = \frac{\exp(\frac{\epsilon^{\triangleright}}{2}) + 1}{\exp(\frac{\epsilon^{\triangleright}}{2}) - 1}. \tag{12}$$

The probability density function of the noisy output $\widehat{x}$ is:

$$Pr(\widehat{x} = c|x) = \begin{cases} p, & \text{if } c \in [\ell(x), \pi(x)] \\ \frac{p}{\exp(\epsilon^{\triangleright})}, & \text{if } c \in [-C, \ell(x)) \cup (\pi(x), C] \end{cases}, \tag{13}$$

where:

$$\begin{aligned} p &= \frac{exp(\epsilon^{\triangleright}) - exp(\epsilon^{\triangleright}/2)}{2exp(\epsilon^{\triangleright}/2) + 2}, \\ \ell(x) &= \frac{C+1}{2} \cdot x - \frac{C-1}{2}, \\ \pi(x) &= \ell(x) + C - 1. \end{aligned} \tag{14}$$

The following lemma establishes the theoretical guarantee of Algorithm 1.

Lemma 4.1. *Algorithm 1 satisfies $\epsilon^{\triangleright}$−local differential privacy.*

Proof. By Eq.(13), let $x, x' \in [-1, 1]$ be any two input values and $\widehat{x} \in [-C, C]$ denote the output of Algorithm 1, then we have:

$$\frac{Pr(\widehat{x}|x)}{Pr(\widehat{x}|x')} \leq \frac{p}{p/\exp(\epsilon^{\triangleright})} = \exp(\epsilon^{\triangleright}). \tag{15}$$

Thus, Algorithm 1 satisfies $\epsilon^{\triangleright}$−LDP. □

However, the PM perturbation presented above is only designed for numerical data that is 1-dimensional. Hence, inspired by [46], we generalize Algorithm 1 to the multidimensional $\mathbf{x}_u$ containing both numerical and categorical attributes. Given $\mathbf{x}_u \in \mathbb{R}^{d_0}$, considering it encodes d' different features in total, we can rewrite it as $\mathbf{x}_u = \mathbf{x}_{(1)} \oplus \mathbf{x}_{(2)} \oplus \cdots \oplus \mathbf{x}_{(d')}$, where the i-th feature $\mathbf{x}_{(i)}$ $(1 \leq i \leq d')$ is either an one-dimensional numeric or an one-hot encoding vector for a categorical feature. On this basis, we propose a comprehensive approach for perturbing such multidimensional data. The detailed perturbation process is depicted in Algorithm 2. Noticeably, we only perturb $\zeta < d'$ features in $\mathbf{x}_u$. This is because that, if we straightforwardly treat each of the d' features in $\mathbf{x}_u$ as an individual element in the dataset, then according to the composition theorem [13], the local privacy budget for each feature will shrink to $\frac{\epsilon^{\triangleright}}{d'}$ in order to maintain $\epsilon^{\triangleright}$−LDP. As a consequence, this will significantly harm the utility of encrypted data. Hence, to preserve reasonable quality of each perturbed numerical or categorical feature, we propose to encrypt only a fraction of (i.e., ζ) features in $\mathbf{x}_u$, ensuring a higher local privacy budget of $\frac{\epsilon^{\triangleright}}{\zeta}$. As shown in Algorithm 2, to prevent privacy leakage, the unselected $d' - \zeta$ features will be dropped by masking them with 0. Thus, to offset the recommendation accuracy loss caused by dropping these features, we follow the empirical study in [46] to determine the appropriate value of ζ:

$$\zeta = \max\{1, \min\{d', \lfloor \frac{\epsilon^{\triangleright}}{2.5} \rfloor\}\}. \tag{16}$$

Additionally, when perturbing each categorical feature $\mathbf{x}_{(i)} \in \mathbf{x}_u$, we extend the continuous sampling strategy in Algorithm 1 to a binarized version for each element/bit within the one-hot encoding $\mathbf{x}_{(i)}$ with the updated local privacy budget $\frac{\epsilon^{\triangleright}}{\zeta}$. As the privacy guarantee of the perturbed categorical feature $\widehat{\mathbf{x}}_{(i)}$ can be verified in a similar way to numerical features [48], we have omitted this part to be succinct. In this regard, our perturbation strategy for the user-centric data in recommendation can provide $\epsilon^{\triangleright}$−LDP, as we summarize below:

Lemma 4.2. *Algorithm 2 satisfies $\epsilon^{\triangleright}$−local differential privacy.*

Proof. As Algorithm 2 is composed of ζ times of $\frac{\epsilon^{\triangleright}}{\zeta}$−LDP operations, then based on the composition theorem [13], Algorithm 2 satisfies $\epsilon^{\triangleright}$−LDP. □

4.2 Loss Perturbation at Optimization Stage

In most scenarios, the results generated by a predictive model (e.g., models for predicting personal credit or diseases) carry highly sensitive information about a user, and this is also the case for recommender systems, since the recommended items can be highly indicative on a user's personal interests and demographics. Though privacy can be achieved via direct perturbation on the generated results [9, 30], it inevitably impedes a model's capability of learning an accurate mapping from its input to output [57], making the learned recommender unable to fully capture personalized user preferences for recommendation. Hence, in the recommendation context, we innovatively propose to perturb the ranking loss $\mathcal{L}$ (i.e., Eq.(11)) instead of perturbing the recommendation results in GERAI. This incurs the analysis of the privacy sensitivity Δ of

Algorithm 3: Optimizing GERAI

Input: Maximum iteration number $\mathcal{T}$, coefficient ϵ and learning rate η
Output: Optimal Parameters Θ^* of GERAI
$\Delta \leftarrow d + \frac{d^2}{4}$;
for $0 \leq j \leq 2$ **do**
  for $\phi \in \Phi_j$ **do**
    $\lambda_\phi \leftarrow \sum_{t \in \mathcal{D}} \lambda_{\phi t} + Lap(\frac{\Delta}{\epsilon |\mathcal{D}|})$, where $t = (u, v, v')$ denotes a triplet training sample;
  end
end
$\widehat{\mathcal{L}} \leftarrow \sum_{j=0}^{2} \sum_{t \in \mathcal{D}} \lambda_{\phi t} (\mathbf{h}^\top \mathbf{q}_{uv} - \mathbf{h}^\top \mathbf{q}_{uv'})$, where $\widehat{\mathcal{L}}$ is the perturbed loss;
Initialize Θ^* randomly;
for *each* $u \in \mathcal{U}$ **do**
  $\widehat{\mathbf{x}}_u \leftarrow$ Algorithm 2;
end
for $t \in \mathcal{T}$ **do**
  Draw a minibatch $\mathcal{B}$;
  $\widehat{\mathcal{L}} \leftarrow$ Eq.(17);
  Take a gradient step to optimize Θ^* with learning rate η;
end
Return Θ^*.

$\mathcal{L}$. For any function, the privacy sensitivity is the maximum L1 distance between its output values given two neighbor datasets differing in one data instance. Intuitively, the larger that Δ is, the heavier perturbation noise is needed to maintain a certain level of privacy. However, directly computing Δ from $\mathcal{L}$ is non-trivial due to its unbounded output range and the complex association between the input data and trainable parameters.

Hence, we present a novel solution to preserving global ϵ−DP for our ranking task. Motivated by the functional mechanism (FM) [57] used for loss perturbation in regression tasks, we first derive a polynomial approximation $\widetilde{\mathcal{L}}$ for $\mathcal{L}$ to allow for convenient privacy sensitivity computation and make the private-preserving optimization process more generic. Then, GERAI perturbs $\widetilde{\mathcal{L}}$ by injecting Laplace noise to enforce ϵ−DP. It is worth noting that, to calculate the privacy sensitivity of $\widetilde{\mathcal{L}}$, we apply a normalization step[1] to every latent predictive feature $\mathbf{q}_{uv}$ produced in Eq.(10), which ensures every element in $\mathbf{q}_{uv}$ is bounded by $(0, 1)$. Using Taylor expansion, we derive $\widetilde{\mathcal{L}}$, the polynomial approximation of $\mathcal{L}$:

$$\widetilde{\mathcal{L}} = \frac{1}{|\mathcal{D}|} \sum_{\forall (u,v,v') \in \mathcal{D}} \sum_{j=0}^{\infty} \frac{f^{(k)}(0)}{k!} (\mathbf{h}^\top \mathbf{q}_{uv} - \mathbf{h}^\top \mathbf{q}_{uv'})^j \tag{17}$$

where $\frac{f^{(k)}(0)}{k!}$ is the k-th derivative of $\widetilde{\mathcal{L}}$ at 0. Recall that $\mathbf{h} = [h_1, h_2, ..., h_d]$ is a projection vector containing d values. Let $\phi(\mathbf{h}) = h_1^{c_1} h_2^{c_2} \cdots h_d^{c_d}$ for $c_1, ..., c_d \in \mathbb{N}$. Let $\Phi_j = \{h_1^{c_1} h_2^{c_2} \cdots h_d^{c_d} \mid \sum_{l=1}^{d} c_l = j\}$ given the degree j (e.g., $\Phi_0 = \{1\}$). Following [57], we truncate the Taylor series in $\widetilde{\mathcal{L}}$ to retain polynomial terms with order lower than 3. Specially, only Φ_0, Φ_1 and Φ_2 involved in $\widetilde{\mathcal{L}}$ with polynomial coefficients as $\frac{f^{(0)}(0)}{0!} = log^2$, $\frac{f^{(1)}(0)}{1!} = -\frac{1}{2}$, $\frac{f^{(2)}(0)}{2!} = \frac{1}{8}$.

[1] This assumption can be easily enforced by the clip function.

Based on $\widetilde{\mathcal{L}}$, we now explore the global privacy sensitivity of the recommendation loss, denoted as Δ. Let $\lambda_{\phi t} \in \mathbb{R}$ denote the coefficient of $\phi(\mathbf{h})$ in the polynomial. In each mini-batch training iteration, the difference of input data only influences these coefficients, so we add perturbation to $\widetilde{\mathcal{L}}$'s coefficients based on the sensitivity. In the following lemma, we derive the global sensitivity Δ of $\widetilde{\mathcal{L}}$, which serves as the important scale factor in determining the noise intensity:

Lemma 4.3. *The global sensitivity of $\widetilde{\mathcal{L}}$ is $d + \frac{d^2}{4}$.*

Proof. Given $\widetilde{\mathcal{L}}$ and two training datasets $\mathcal{D}$, $\mathcal{D}'$ that differ in only one instance, for $J \geq 1$ and $\overline{\mathbf{q}} = [\overline{q}_1, \overline{q}_2, ..., \overline{q}_d] = \mathbf{q}_{uv} - \mathbf{q}_{uv'}$, we can derive:

$$\begin{aligned}
\Delta &= \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} || \sum_{t \in \mathcal{D}} \lambda_{\phi t} - \sum_{t' \in \mathcal{D}'} \lambda_{\phi t'} ||_1 \\
&\leq 2 \cdot \max_t \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} ||\lambda_{\phi t}||_1 \\
&\leq 2 \cdot \max_t \Big(\frac{f^{(1)}(0)}{1!} \sum_{m=1}^{d} \overline{q}_m \Big) + \frac{f^{(2)}(0)}{2!} \sum_{m \geq 1, n \leq d} \overline{q}_m \overline{q}_n \\
&\leq 2 (\frac{\dim(\mathbf{q}_{uv})}{2} + \frac{\dim(\mathbf{q}_{uv})^2}{8}) \\
&= d + \frac{d^2}{4},
\end{aligned} \tag{18}$$

where $t = (u, v, v') \in \mathcal{D}$ is an arbitrary training sample and $\dim(\cdot)$ returns the dimension of a given vector. □

Specifically, we employ FM to perturb the loss $\widetilde{\mathcal{L}}$ by injecting Laplace noise[2] $Lap(\frac{\Delta}{\epsilon |\mathcal{D}|})$ into its polynomial coefficients, and the perturbed function is denoted by $\widehat{\mathcal{L}}$. The injected Laplace noise with standard deviation of $\frac{\Delta}{\epsilon |\mathcal{D}|}$ has been widely proven to effectively retain ϵ−DP after perturbation [12, 13, 57]. Note that as Δ is the global sensitivity, it is evenly distributed to all instances in the training set $\mathcal{D}$ during perturbation. We showcase the full training process of GERAI with a differentially private loss in Algorithm 3. In Algorithm 3, we first compute the sensitivity Δ of loss $\widetilde{\mathcal{L}}$. In each iteration, we add perturbation to every coefficient in the polynomial approximation of the loss function. Afterwards, we launch the training session for GERAI with perturbed user feature vectors $\{\widehat{\mathbf{x}}_u | u \in \mathcal{U}\}$, where we use the perturbed coefficients to obtain the perturbed loss $\widehat{\mathcal{L}}$ and optimize the parameters of the model by minimizing $\widehat{\mathcal{L}}$. Finally, we formally prove that Algorithm 3 satisfies ϵ−DP:

Lemma 4.4. *Algorithm 3 maintains ϵ−differential privacy.*

[2]In our paper, the mean of our Laplace distribution is 0, i.e., $Lap(\cdot) = Lap(0, \cdot)$.

Table 1: Features extracted from the dataset.

- Number of rated products
- Number and ratio of each rating level given by a user
- Ratio of positive and negative ratings: The proportions of high ratings (4 and 5) and low ratings (1 and 2) of a user.
- Entropy of ratings: It is calculated as $-\sum_{\forall r} Prop_r \log Prop_r$, where $Prop_r$ is the proportion that a user gives the rating of r.
- Median, min, max, and average of ratings
- Gender: It is either male or female.
- Occupation: A total of 21 possible occupations are extracted.
- Age: We categorize age attribute into 3 groups: over 45, under 35, and between 35 and 45.

Proof. Assume that $\mathcal{D}$ and $\mathcal{D}^{'}$ are two training datasets differing in only one instance denoted by T and T', then we have:

$$\begin{aligned}
\frac{Pr(\widehat{\mathcal{L}}|\mathcal{D})}{Pr(\widehat{\mathcal{L}}|\mathcal{D}')} &= \frac{\Pi_{j=1}^{2} \Pi_{\phi \in \Phi_j} \exp(\frac{\epsilon}{\Delta} || \sum_{t \in \mathcal{D}} \lambda_{\phi t} - \lambda_{\phi} ||_1)}{\Pi_{j=1}^{2} \Pi_{\phi \in \Phi_j} \exp(\frac{\epsilon}{\Delta} || \sum_{t' \in \mathcal{D}'} \lambda_{\phi t'} - \lambda_{\phi} ||_1)} \\
&\leq \Pi_{j=1}^{2} \Pi_{\phi \in \Phi_j} \exp(\frac{\epsilon}{\Delta} || \sum_{t \in \mathcal{D}} \lambda_{\phi t} - \sum_{t' \in \mathcal{D}'} \lambda_{\phi t'} ||_1) \\
&= \Pi_{j=1}^{2} \Pi_{\phi \in \Phi_j} \exp(\frac{\epsilon}{\Delta} || \lambda_{\phi T} - \lambda_{\phi T'} ||_1) \\
&= \exp(\frac{\epsilon}{\Delta} \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} ||\lambda_{\phi T} - \lambda_{\phi T'}||_1) \\
&\leq \exp(\frac{\epsilon}{\Delta} \cdot 2 \cdot \max_T \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} ||\lambda_{\phi T}||_1) = \exp(\epsilon).
\end{aligned} \tag{19}$$

Then according to Definition 1, Algorithm 3 satisfies ϵ−DP. □

In short, with our proposed dual-stage perturbation strategy for both the user data and the training loss, GERAI fully preserves user privacy with a demonstrable guarantee, while being able to achieve minimal compromise on the recommendation effectiveness compared with a non-private, GCN-based counterpart. Furthermore, GERAI can be trained via stochastic gradient descent (SGD) algorithms in an end-to-end fashion, showing its real-world practicality.

5 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of GERAI in terms of both privacy strength and recommendation effectiveness. Particularly, we aim to answer the following research questions (RQs):

- **RQ1:** Can GERAI effectively protect sensitive user data from attribute inference attack?
- **RQ2:** How does GERAI perform in top-K recommendation?
- **RQ3:** How does the key hyperparameters affect the privacy-preserving property and recommendation accuracy of GERAI?
- **RQ:4** What is the contribution from each part of the dual-stage perturbation paradigm in GERAI?
- **RQ5:** Can GERAI defend different types of unseen attribute inference attack models?

5.1 Dataset

Following [3], we use the publicly available ML-100K datasets [1] in our experiments. It contains 10, 000 ratings from 943 users on 1, 682 movies collected from the MovieLens website. In addition, in the collected dataset, each user is associated with three sensitive attributes, i.e., gender (Gen), age (Age) and occupation (Occ). Similar to [3], we convert the gender, age and occupation into a 2, 3 and 21-dimensional categorical feature, respectively. Table 1 provides a summary of all the features we have used.

5.2 Baseline Methods and Parameter Settings

We evaluate GERAI by comparing with the following baselines:

- **BPR**: It is a widely used non-private learning-to-rank model for recommendation [41].
- **GCN**: This is the non-private, GCN-based recommendation model proposed in [55].
- **Blurm**: This method directly uses perturbed user-item ratings to train the recommender system [50].
- **DPAE**: In DPAE, Gaussian mechanism is combined in the stochastic gradient descent process of an autoencoder-based recommender so that the training phase meets the requirements of differential privacy [32].
- **DPNE**: It aims to develop a differentially private network embedding method based on matrix factorization, and it is the state-of-the-art privacy preserving network embedding method for link prediction [53].
- **DPMF**: It uses objective perturbation with matrix factorization to ensure the final item profiles satisfy differential privacy [24].
- **RAP**: It is the state-of-the-art recommendation model that is designed against attribute inference attacks [3]. The key idea is to facilitate adversarial learning with an RNN-based private attribute inference attacker and a CF-based recommender.

In GERAI, we set γ, learning rate and batch size to 0.01, 0.005 and 64, respectively. Without special mention, we use three-layer networks for the neural components and initialized parameters to random values by using Gaussian distribution, which has 0 mean and a standard deviation of 1. The final embedding dimension is $d = 60$ and the privacy budget is $\epsilon = 0.4$ and $\epsilon^{\triangleright} = 20$, while the effect of different hyperparameter values will be further discussed in Section 5.6. For all baseline methods, we use the optimal hyperparameters provided in the original papers.

5.3 Evaluation Protocols

Attribute Inference Attack Resistance. To evaluate all models' robustness against attribute inference attacks, we first build a strong adversary classifier (i.e., attacker). Specifically, we use a two-layer deep neural network model as the attacker. Suppose there are K items $\mathcal{R}(u)$ recommended by a fully trained recommender to user $u \in \mathcal{U}$, then the input of the attacker is formulated as $\sum_{\forall v \in \mathcal{I}(u)} onehot(v) + \sum_{\forall v \in \mathcal{R}(u)} onehot(v)$ where $onehot(\cdot)$ returns the one-hot encoding of a given item. The hidden dimension is set to 100, and a linear projection is used to estimate the class of the target attribute. We randomly choose 80% of the labelled users to train the attacker, and use the remainder to test the attacker's inference accuracy. Note that the attacker model is unknown to all recommenders during the training process. To quantify a model's privacy-preserving capability, we leverage a widely-used classification metric *F1 score* [60] to evaluate the classification performance of the attacker. Correspondingly, lower F1 scores demonstrate higher resistance to this inference attack.

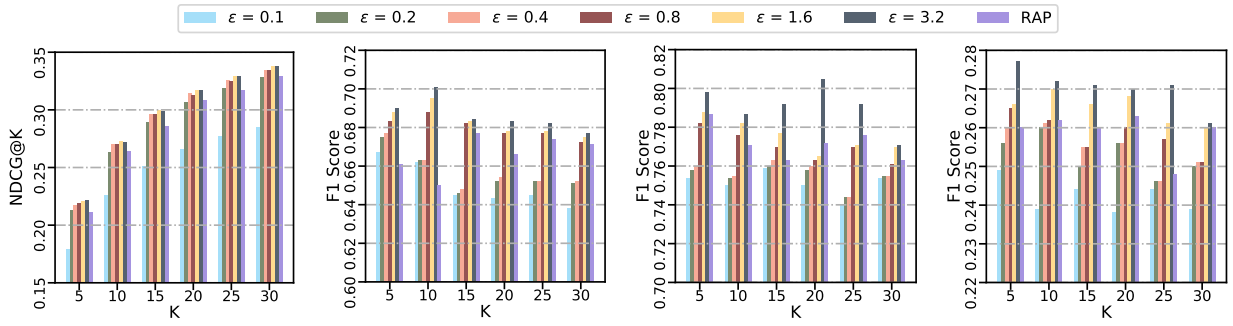
Table 2: Attribute inference attack results. Lower F1 scores represent better privacy protection from the model.

Attribute	Method	F1 Score					
		K=5	K=10	K=15	K=20	K=25	K=30
Age	BPR	0.693	0.694	0.699	0.720	0.676	0.693
	GCN	0.697	0.725	0.730	0.725	0.735	0.746
	Blurm	0.715	0.725	0.716	0.692	0.679	0.710
	DPAE	0.694	0.688	0.695	0.674	0.695	0.684
	DPNE	0.684	0.685	0.700	0.701	0.679	0.674
	DPMF	0.709	0.703	0.695	0.699	0.684	0.689
	RAP	**0.661**	**0.650**	0.677	0.666	0.674	0.671
	GERAI	0.677	0.663	**0.648**	**0.651**	**0.652**	**0.650**
Gen	BPR	0.810	0.773	0.808	0.778	0.782	0.801
	GCN	0.851	0.836	0.891	0.880	0.862	0.869
	Blurm	0.789	0.788	0.789	0.761	0.761	0.788
	DPAE	0.781	0.771	0.770	0.772	0.771	0.777
	DPNE	0.788	0.772	0.781	0.776	0.798	0.788
	DPMF	0.783	0.770	0.768	0.765	0.761	0.771
	RAP	0.787	0.771	0.763	0.772	0.776	0.763
	GERAI	**0.760**	**0.755**	**0.763**	**0.760**	**0.744**	**0.755**
Occ	BPR	0.276	0.277	0.264	0.263	0.289	0.267
	GCN	0.277	0.277	0.277	0.267	0.272	0.270
	Blurm	0.267	0.267	0.262	0.262	0.267	0.269
	DPAE	0.266	0.260	0.255	0.261	0.260	0.261
	DPNE	0.267	0.265	0.266	0.264	0.266	0.262
	DPMF	0.266	0.262	0.270	0.265	0.270	0.267
	RAP	0.260	0.262	0.260	0.263	0.248	0.260
	GERAI	**0.260**	**0.261**	**0.255**	**0.256**	**0.246**	**0.251**

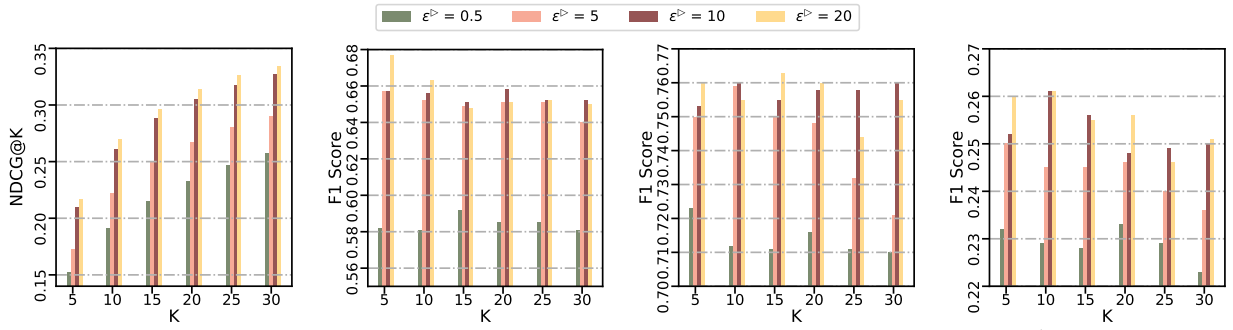
Recommendation Effectiveness. For each user, we randomly pick 80% of her/his interacted items to train all recommendation models, while the rest 20% is held out for evaluation. We employ $Hit@K$ and $NDCG@K$, which are two popular metrics to judge the quality of the top-K ranking list. Results on both attribute inference and recommendation are averaged over five runs.

5.4 Privacy Protection Effectiveness (RQ1)

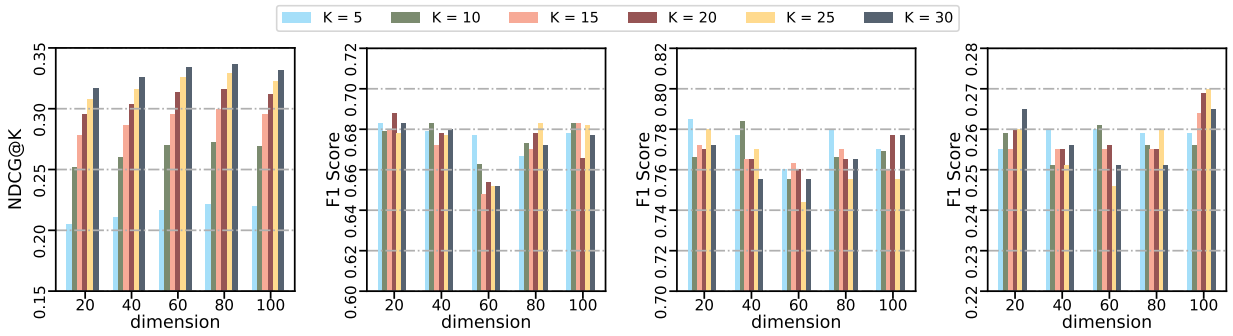
Table 2 shows the F1 scores achieved by the attribute inference attack model described in Section 5.3 on all the baselines. Lower F1 scores show higher resistance of the recommender to attribute inference attacks. Obviously, GERAI constantly outperforms all baselines with $K \in \{15, 20, 25, 30\}$, indicating that our model is able to protect users' privacy and produce recommendations with strong privacy guarantee. Though RAP achieves slightly better results on the age attribute at $K = 5$ and $K = 10$, it falls behind GERAI in all other cases. As a model specifically designed for supervised learning, RAP is naturally robust against attribute inference attack. We also observe that GERAI has significantly better performance against attribute inference attack in comparison to Blurm that obfuscates user-item rating data to the recommender system. The results confirm the effectiveness of our dual-stage perturbation



(a) Above: Recommendation, Age, Gender and Occupation Inference results w.r.t. privacy budget ϵ



(b) Above: Recommendation, Age, Gender and Occupation Inference results w.r.t. privacy budget $\epsilon^{\triangleright}$



(c) Above: Recommendation, Age, Gender and Occupation Inference results w.r.t. dimension d

Figure 2: Recommendation and privacy protection results w.r.t. privacy budget ϵ, $\epsilon^{\triangleright}$ and d.

in private attribute protection. In addition, compared with conventional recommender systems that collaboratively model user-item interactions (i.e., BPR and GCN), models that make use of differential privacy (i.e., DPAE, DPMF, DPNE and GERAI) show obvious superiority in resistance to attribute inference attack. However, compared with all DP-based recommender systems, GERAI achieves significantly lower $F1$ score for all three private attributes and thus outperform those methods in terms of obscuring users' private attribute information. The reason is that the proposed privacy mechanisms in those DP-based methods cannot have the same strength as GERAI on preventing leakage of sensitive information from recommendation results. This further validates that incorporating differential privacy may prevent directly disclosing private attributes, but these methods cannot effectively provide higher privacy levels. Furthermore, with the increasing value of K, the performance of the attacker slightly decreases. One possible reason is that, more recommended products will become a natural "noise" to help reduce the risk of privacy disclosure. Finally, we observe that GCN has the weakest privacy protection results because it directly incorporates the node features with sensitive information. Note that compared with GCN, GERAI achieves an average relative improvement of 11%, 14.4% and 6.75% respectively on age, gender and occupation, which implies that DP can ensure that the published recommendations of GERAI can avoid breaching users' privacy.

5.5 Recommendation Effectiveness (RQ2)

We summarize all models' performance on personalized recommendation with Table 3. Note that higher $Hit@K$ and $NDCG@K$ values imply higher recommendation quality. Firstly, GERAI outperforms all privacy-preserving baselines consistently in terms of both $Hit@K$ and $NDCG@K$. Particularly, the improvement of GERAI with $K = 5$ demonstrate that our model can accurately rank the ground truth movies at the top-5 positions. In addition, compared with RAP, GERAI yields recommendation results that are closer to the state-of-the-art GCN. Thanks to the dual-stage perturbation setting where two sets of privacy budgets are used, a relatively

Table 3: Recommendation effectiveness results. For both Hit@K and NDCG@K, the higher the better.

	Method	K=5	K=10	K=15	K=20	K=25	K=30
Hit@K	BPR	0.348	0.507	0.614	0.686	0.741	0.791
	GCN	0.365	0.519	0.619	0.690	0.743	0.789
	Blurm	0.184	0.263	0.319	0.364	0.405	0.443
	DPAE	0.185	0.285	0.345	0.394	0.438	0.458
	DPNE	0.301	0.430	0.525	0.595	0.640	0.684
	DPMF	0.195	0.280	0.343	0.394	0.432	0.474
	RAP	0.319	0.475	0.575	0.648	0.706	0.754
	GERAI	0.333	0.495	0.600	0.670	0.724	0.767
NDCG@K	BPR	0.228	0.280	0.310	0.330	0.341	0.363
	GCN	0.247	0.296	0.323	0.340	0.351	0.360
	Blurm	0.124	0.148	0.164	0.174	0.183	0.191
	DPAE	0.126	0.153	0.170	0.176	0.180	0.188
	DPNE	0.204	0.231	0.268	0.289	0.299	0.306
	DPMF	0.134	0.154	0.171	0.182	0.191	0.186
	RAP	0.211	0.264	0.286	0.308	0.317	0.329
	GERAI	0.217	0.270	0.296	0.314	0.326	0.334

Table 4: Ablation test results.

Variant	Recommendation Task		Attribute Inference Attack (F1 score)		
	Hit@5	NDCG@5	Age	Gen	Occ
GCN	0.365	0.247	0.697	0.851	0.277
GERAI-NL	0.340	0.221	0.688	0.791	0.270
GERAI-NF	0.337	0.219	0.679	0.788	0.266
GERAI	0.333	0.217	0.677	0.760	0.260

higher privacy for user feature perturbation does not significantly impede the recommendation accuracy, and is sufficient for high-level attribute protection. Furthermore, the gap between the ranking accuracy drops with the increasing value of K. Finally, GCN achieves the best performance among all methods except when $K = 30$, which showcases the intrinsic strength of GCN-based recommenders. Meanwhile, Blurm has the worst performance among all methods as the way it adds noise to the user-item interaction data is harmful for the recommendation quality.

5.6 Accuracy and Privacy (RQ3)

We answer RQ3 by investigating the performance fluctuations of GERAI with varied global and local privacy budgets ϵ, $\epsilon^{\triangleright}$ and embedding dimension d. We vary the value of one hyperparameter while keeping the other unchanged, and record the new recommendation and attribute inference results achieved. Figure 2 plots the results with different parameter settings.

Impact of Global Privacy Budget ϵ for Loss Perturbation. The value of the privacy budget ϵ is examined in $\{0.1, 0.2, 0.4, 0.8, 1.6, 3.2\}$. In general, our GERAI outperforms RAP in terms of recommendation accuracy, and the performance improvement tends to become less significant when ϵ becomes quite small. Since a smaller ϵ requires a larger amount of noise to be injected to the objective function, it negatively influences the recommendation results. The results further confirms the effectiveness of GCNs-based recommendation component in our model, which helps GERAI preserve recommendation quality in practice. Furthermore, though the attack results illustrate that a relatively small ϵ (large noise) can obtain better performance on privacy protection within our expectation, it also degraded recommendation results correspondingly. Compared with RAP, the results imply that, by choosing a proper value of ϵ (0.4 in our case), our GERAI can achieve a good trade-off between privacy protection and recommendation accuracy.

Impact of Local Privacy Budget $\epsilon^{\triangleright}$ for User Feature Perturbation. We study the impact of the privacy budget on input features with $\epsilon^{\triangleright} \in \{0.5, 5, 10, 20\}$. It is worth mentioning that we seek a relatively higher value of $\epsilon^{\triangleright}$ to maintain moderate utility of user features. From Figure 2, we can draw the observation that though reducing the value of privacy budget $\epsilon^{\triangleright}$ in the input features may help the model yield better performance against attribute inference attack, GERAI generally achieves a significant drop on recommendation performance with a smaller $\epsilon^{\triangleright}$. Particularly, when $\epsilon^{\triangleright} = 0.5$, the recommendation results show that GERAI cannot capture users' actual preferences. This is because the feature vector $\widehat{\mathbf{x}}_u$ determines the number of non-zero elements in base embedding of our model, which can cause significant information loss when it is small. As the recommendation is also highly accurate when $\epsilon^{\triangleright} = 10$, the attribute inference performance achieved by the attacker is occasionally comparable to setting $\epsilon^{\triangleright} = 20$. Overall, setting $\epsilon^{\triangleright}$ to 20 is sufficient for preventing privacy leakage, while helping GERAI to achieve optimal recommendation results.

Impact of Dimension d. As suggested by Eq.(18), the dimension d controls the privacy sensitivity Δ and our model's expressiveness of the network structure. We vary the dimension d in $\{20, 40, 60, 80, 100\}$ and the corresponding noise parameters in Laplace distribution are $\{0.00375, 0.01375, 0.03, 0.05, 0.08\}$. Obviously, the recommendation accuracy of GERAI benefits from a relatively larger dimension d, but the privacy protection performance is not always lower with a large d. The reason is that the value of the dimension d is directly associated with our model's expressiveness, which means that a relatively larger d can improve the recommendation results, providing better inputs to the attacker model as well. Furthermore, as shown in Figure 2, the best privacy protection performance is commonly observed with $d = 60$.

5.7 Importance of Privacy Mechanism (RQ4)

To better understand the performance gain from the major components proposed in GERAI, we perform ablation analysis on different degraded versions of GERAI. Each variant removes one privacy mechanism from the dual-stage perturbation paradigm. Table 4 summarizes the outcomes in two tasks in terms of $Hit@5$, $NDCG@5$ and *F1 score*. For benchmarking, we also demonstrate the results from the full version of GERAI and the non-private GCN.

Removing perturbation at input stage (GERAI-NL). The GERAI-NL only enforces ϵ-differential privacy by perturbing the objective function in Eq. (17). We remove the privacy mechanism in users' features by sending raw features $\mathbf{X}$ directly into the recommendation component. After that, a slight performance decrease in the recommendation accuracy appeared, while achieving better performance against attribute inference attack. The results confirm that the functional mechanism in our model can help a GCN-based recommender satisfy privacy guarantee and yield comparable recommendation accuracy. In addition, GERAI significantly outperform GERAI-NL against attribute inference attack. Apparently, the raw user features are not properly perturbed in GERAI-NL, leading to a high potential risk in privacy leakage.

Table 5: Performance of attribute-inference attack w.r.t. different types of attacker.

Attribute	Method	F1 Score			
		DT	NB	KNN	GP
Age	BPR	0.466	0.376	0.487	0.260
	GCN	0.513	0.366	0.487	0.619
	Blurm	0.471	0.402	0.492	0.265
	DPAE	0.492	0.481	0.476	**0.249**
	DPNE	0.593	0.402	0.476	0.349
	DPMF	0.561	0.402	0.486	0.275
	RAP	0.476	0.407	0.513	0.534
	GERAI	**0.434**	**0.365**	**0.466**	0.286
Gen	BPR	0.651	0.444	0.561	0.672
	GCN	0.635	0.429	0.566	0.810
	Blurm	0.630	**0.370**	0.556	0.693
	DPAE	0.635	0.381	**0.545**	0.683
	DPNE	0.640	0.381	0.556	0.667
	DPMF	0.683	0.376	0.556	0.683
	RAP	0.670	0.439	0.619	0.709
	GERAI	**0.619**	0.429	0.556	**0.656**
Occ	BPR	0.132	0.148	0.070	0.116
	GCN	0.122	0.148	0.063	0.222
	Blurm	0.127	0.185	0.074	0.105
	DPAE	0.111	0.180	**0.063**	0.212
	DPNE	0.132	0.175	0.079	0.106
	DPMF	0.175	0.180	0.074	0.104
	RAP	0.122	0.148	0.090	0.127
	GERAI	**0.111**	**0.116**	0.069	**0.090**

Removing perturbation at optimization stage (GERAI-NF). We remove the privacy mechanism in objective function by setting $\epsilon = 0$. As the users' features are perturbed against information leaks, GERAI-NF achieves a significant performance improvement in the privacy protection, compared with the pure GCN. In addition, the slight performance difference between GERAI and GERAI-NF in two tasks could be attributed to the perturbation strategy in objective function. It further verifies that the joint effect of perturbation strategies in objective function and input features are beneficial for both recommendation and privacy protection purposes.

5.8 Robustness against Different Attribute Inference Attackers (RQ5)

In real-life scenarios, the models used by attribute inference attacker are usually unknown and unpredictable, so hereby we investigate how GERAI and other baseline methods perform in the presence of different types of attack models, namely Decision Tree (DT), Naive Bayesian (NB), KNN and Gaussian Process (GP), that are widely adopted classification methods. In this study, we use the top-5 recommendation generated by corresponding recommender methods for all attackers as introduced in Section 5.3. Table 5 shows the attribute inference accuracy of each attacker. The first observation is that our proposed GERAI outperforms all the comparison methods in most scenarios. Though DPAE achieves slightly better results in several cases, its recommendation accuracy is non-comparable to GERAI. This further validates the challenge of incorporating privacy protection mechanism for personalized recommendation. Another observation is that there is a noticeable performance drop of RAP facing non-DNN attacker models. As RAP is trained to defend a specific DNN-based inference model, RAP is more effective when attacker is also DNN-based as shown in Table 2. However, RAP underperforms when facing the other five commonly used inference models, showing that GERAI can more effectively resist attribute inference attacks and protect users' privacy without any assumption on the type of attacker models.

6 RELATED WORK

Attribute Inference Attacks. The target of attribute inference attack is inferring users' private information from their publicly available information (e.g. recommendations). Three main branches of attribute inference attack approaches are often distinguished: friend-based, behavior-based and hybrid approaches. Friend-based approaches infer the target user's attribute in accordance with the target's friends' information [19, 22, 31]. He et al [22] first constructed a Bayesian network to model the causal relations among people in social networks, which is used to obtain the probability that the user has a specific attribute. Behavior-based approaches achieve this purpose via users' behavioral information such as movie-rating behavior [50] and Facebook likes [29]. The third type of works exploits both friend and behavioral information [17, 18, 23]. For example, [19] creates a social-behavior-attribute network to infer attributes. Another work [23] models structural and behavioral information from users who do not have the attribute in the training process as a pairwise Markov Random Field.

Privacy and Recommender System. With the growth of online platforms, recommender systems play a pivotal role in promoting sales and enhancing user experience. The recommendations, however, may pose a severe threat to user privacy such as political inclinations via attribute inference attack. Hence, it is of paramount importance for system designers to construct a recommender system that can generate accurate recommendations and guarantee the privacy of users. Current researches that address vulnerability to privacy attacks often rely on providing encryption schemes [6, 27] and differential privacy [25]. Encryption-based methods enhance privacy of the conventional recommender systems with advanced encryption techniques such as homomorphic encryption [8, 27]. However, these methods are considered computation expensive as a third-party crypto-service provider is required. DP-based recommender systems can provide a strong and mathematically rigorous privacy guarantee [4, 33, 34]. Works in this area aim to ensure that the recommender systems are not sensitive to any particular record and thus prevent adversaries from inferring a target user's ratings. [38] proposes a perturbation method that adds or removes items and ratings to minimize privacy risk. Similarly, RAPPOR [15] is proposed to perturb the user's data before sending them to the server by using the randomized response. More recently, graph embedding techniques have been opening up more chances to improve the efficiency and scalability of the existing recommender systems [7, 55]. As the core of GCN is a graph embedding algorithm, our work is also quite related to another area: privacy preservation on graph embedding. Hua et al. [24] and Shin et al. [43] proposed gradient perturbation algorithms for differentially private matrix factorization to protect users' ratings and profiles. Another work

enforces differential privacy to construct private covariance matrices to be further used by recommender [12]. Liu et al. [32] proposed DPAE that leverages the privacy problem in recommendation with the Autoencoders. Gaussian noise is added in the process of gradient descent. However, the existing privacy-preserving works in recommendation systems focus on protecting users against the membership attacks in which an adversary tries to infer a targeted user's actual ratings and deduce if the target is in the database, which is not fulfilled in our scenario. These limitations motivated us to propose GERAI that is able to counter private attribute inference attacks in the personalized recommendation system.

7 CONCLUSION

In this paper, we propose a GCN-based recommender system that guards users against attribute inference attacks while maintaining utility, named GERAI. GERAI firstly masks users' features including sensitive information, and then incorporates differential privacy into the GCN, which effectively bridges user preferences and features for generating secure recommendations such that a malicious attacker cannot infer their private attribute from users' interaction history and recommendations. The experimental results evidence that GERAI can yield superior performance on both recommendation and attribute protection tasks.

ACKNOWLEDGMENTS

The work has been supported by Australian Research Council (Grant No.DP190101985 and DP170103954).

REFERENCES

[1] 2019. MovieLens. http://grouplens.org/datasets/movielens
[2] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. 217–253.
[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] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying differential privacy to matrix factorization. In *RECSYS*. 107–114.
[5] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. 2011. " You might also like:" Privacy risks of collaborative filtering. In *IEEE symposium on security and privacy*. 231–246.
[6] John Canny. 2002. Collaborative filtering with privacy. In *IEEE Symposium on Security and Privacy*. 45–57.
[7] Gjorgjina Cenikj and Sonja Gievska. 2020. Boosting Recommender Systems with Advanced Embedding Models. In *WWW Companion*. 385–389.
[8] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. Secure federated matrix factorization. *IEEE Intelligent Systems* (2020).
[9] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 3 (2011).
[10] Tong Chen, Hongzhi Yin, Hongxu Chen, Rui Yan, Quoc Viet Hung Nguyen, and Xue Li. 2019. Air: Attentional intention-aware recommender systems. In *ICDE*. 304–315.
[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*. 3844–3852.
[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. 265–284.
[13] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* (2014), 211–407.
[14] Zekeriya Erkin, Michael Beye, Thijs Veugen, and Reginald L Lagendijk. 2010. Privacy enhanced recommender system. In *SITB*. 35–42.
[15] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *SIGSAC*. 1054–1067.
[16] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. 2013. Exploiting innocuous activity for correlating users across sites. In *WWW*. 447–458.
[17] Neil Zhenqiang Gong and Bin Liu. 2016. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *USENIX*. 979–995.
[18] Neil Zhenqiang Gong and Bin Liu. 2018. Attribute inference attacks in online social networks. *ACM Transactions on Privacy and Security* (2018), 1–30.
[19] Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine Shi, and Dawn Song. 2014. Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology* (2014), 1–20.
[20] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming session-based recommendation. In *SIGKDD*. 1569–1577.
[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.
[22] Jianming He, Wesley W Chu, and Zhenyu Victor Liu. 2006. Inferring privacy information from social networks. In *ISI*. 154–165.
[23] Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. 2017. AttriInfer: Inferring user attributes in online social networks using markov random fields. In *WWW*. 1561–1569.
[24] Hua Jingyu, Xia Chang, and Zhong Sheng. 2015. Differentially Private Matrix Factorization. In *IJCAI*. 57–62.
[25] Thivya Kandappu, Arik Friedman, Roksana Boreli, and Vijay Sivaraman. 2014. PrivacyCanary: Privacy-aware recommenders with adaptive input obfuscation. In *MASCOTS*. 453–462.
[26] Rimma Kats. 2018. Many Facebook Users are Sharing Less Content. In *eMarketer, https://www.emarketer.com/content/many-facebook-users-are-sharing-less-content-because-of-privacy-concerns*.
[27] Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, Junbum Shin, and Sungwook Kim. 2018. Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption. *ACM Transactions on Privacy and Security* (2018), 1–30.
[28] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
[29] Michal Kosinski, David Stillwell, and Thore Graepel. 2013. Private traits and attributes are predictable from digital records of human behavior. *PNAS* (2013), 5802–5805.
[30] Jing Lei. 2011. Differentially private m-estimators. In *NeurIPS*. 361–369.
[31] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. 2009. Inferring private information using social network data. In *WWW*. 1145–1146.
[32] Xiaoqian Liu, Qianmu Li, Zhen Ni, and Jun Hou. 2019. Differentially private recommender system with autoencoders. In *iThings*. 450–457.
[33] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. 2015. Fast differentially private matrix factorization. In *RECSYS*. 171–178.
[34] Frank McSherry and Ilya Mironov. 2009. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *SIGKDD*. 627–636.
[35] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NeurIPS*. 1257–1264.
[36] A Naranyanan and V Shmatikov. 2008. Robust de-anonymization of large datasets. In *IEEE Symposium on Security and Privacy*. 111–125.
[37] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. 2013. Privacy-preserving matrix factorization. In *SIGSAC*. 801–812.
[38] Javier Parra-Arnau, David Rebollo-Monedero, and Jordi Forné. 2014. Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems. *Entropy* (2014), 1586–1631.
[39] Huseyin Polat and Wenliang Du. 2005. Privacy-preserving collaborative filtering. *International journal of electronic commerce* (2005), 9–35.
[40] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. 2002. Getting to know you: learning new user preferences in recommender systems. In *IUI*. 127–134.
[41] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. *UAI* (2009), 452–461.
[42] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018), 357–370.
[43] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. 2018. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering* (2018), 1770–1782.
[44] Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. 2017. User identity linkage across online social networks: A review. *Acm SIGKDD Explorations Newsletter* (2017), 5–17.
[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. In *ICLR*.

[46] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and analyzing multidimensional data with local differential privacy. In *ICDE*. 638–649.
[47] Qinyong Wang, Hongzhi Yin, Tong Chen, Zi Huang, Hao Wang, Yanchang Zhao, and Nguyen Quoc Viet Hung. 2020. Next Point-of-Interest Recommendation on Resource-Constrained Mobile Devices. In *WWW*. 906–916.
[48] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In *USENIX*. 729–745.
[49] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.
[50] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. 2012. BlurMe: Inferring and obfuscating user gender based on ratings. In *RECSYS*. 195–202.
[51] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2020. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *AAAI*.
[52] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based poi embedding for location-based recommendation. In *CIKM*. 15–24.
[53] Depeng Xu, Shuhan Yuan, Xintao Wu, and HaiNhat Phan. 2018. DPNE: Differentially private network embedding. In *PAKDD*. 235–246.
[54] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social influence-based group representation learning for group recommendation. In *ICDE*. 566–577.
[55] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*. 974–983.
[56] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2020. Enhance Social Recommendation with Adversarial Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).
[57] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. 2012. Functional mechanism: regression analysis under differential privacy. *VLDB* (2012).
[58] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. 2020. GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection. In *SIGIR*. 689–698.
[59] Shijie Zhang, Hongzhi Yin, Qinyong Wang, Tong Chen, Hongxu Chen, and Quoc Viet Hung Nguyen. 2019. Inferring Substitutable Products with Deep Network Embedding.. In *IJCAI*. 4306–4312.
[60] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW*. 22–32.