Robust federated contrastive recommender system against targeted model poisoning attack

SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2025, Vol. 68, Iss. 4, 140103:1–140103:16
https://doi.org/10.1007/s11432-024-4272-y

Special Topic: Cloud-Edge Collaboration for On-Device Recommendation

Robust federated contrastive recommender system against targeted model poisoning attack

Wei YUAN[1†], Chaoqun YANG[2†], Liang QU[1], Guanhua YE[3*], Quoc Viet Hung NGUYEN[2] & Hongzhi YIN[1*]

[1]*School of Electrical Engineering and Computer Science, The University of Queensland, Brisbane 4072, Australia*
[2]*School of Information and Communication Technology, Griffith University, Gold Coast 4222, Australia*
[3]*Deep Neural Computing Company Limited, Shenzhen 518000, China*

Received 10 April 2024/Revised 8 August 2024/Accepted 8 January 2025/Published online 18 March 2025

Abstract Federated recommender systems (FedRecs) have garnered increasing attention recently, thanks to their privacy-preserving benefits. However, the decentralized and open characteristics of current FedRecs present at least two dilemmas. First, the performance of FedRecs is compromised due to highly sparse on-device data for each client. Second, the system's robustness is undermined by the vulnerability to model poisoning attacks launched by malicious users. In this paper, we introduce a novel contrastive learning framework designed to fully leverage the client's sparse data through embedding augmentation, referred to as CL4FedRec. Unlike previous contrastive learning approaches in FedRecs that necessitate clients to share their private parameters, our CL4FedRec aligns with the basic FedRec learning protocol, ensuring compatibility with most existing FedRec implementations. We then evaluate the robustness of FedRecs equipped with CL4FedRec by subjecting it to several state-of-the-art model poisoning attacks. Surprisingly, our observations reveal that contrastive learning tends to exacerbate the vulnerability of FedRecs to these attacks. This is attributed to the enhanced embedding uniformity, making the polluted target item embedding easily proximate to popular items. Based on this insight, we propose an enhanced and robust version of CL4FedRec (rCL4FedRec) by introducing a regularizer to maintain the distance among item embeddings with different popularity levels. Extensive experiments conducted on four commonly used recommendation datasets demonstrate that rCL4FedRec significantly enhances both the model's performance and the robustness of FedRecs.

Keywords federated recommender system, contrastive learning, model poisoning attack and defense

Citation Yuan W, Yang C Q, Qu L, et al. Robust federated contrastive recommender system against targeted model poisoning attack. Sci China Inf Sci, 2025, 68(4): 140103, https://doi.org/10.1007/s11432-024-4272-y

1 Introduction

Recommender systems have evolved into an indispensable component of numerous online services (e.g., social media [1–3], e-commerce [4], and online news [5]) to assist users in identifying the potential interests among massive information. Traditionally, recommender systems have been built by leveraging user personal data gathered and stored in a centralized server [6]. Nevertheless, in light of the increasing privacy concerns and the advent of stringent privacy protection regulations, such as the General Data Protection Regulation (GDPR[1)]) and the California Consumer Privacy Act (CCPA[2)]), conventional centralized recommender systems encounter the imminent risks of leaking sensitive user information and running afoul of these regulatory frameworks [7].

To address privacy concerns, federated learning [8], a privacy-preserving training paradigm, has been embraced in recommender systems, giving rise to federated recommender systems (FedRecs) [9]. Within FedRecs, the recommender model is partitioned into public and private parameters. Private parameters, such as user embeddings, are locally maintained on the user side, while public parameters, like item embeddings, are exchanged between users and the central server to facilitate collaborative learning. Throughout the training process, users/clients[3)] train their local recommender models using personal data

* Corresponding author (email: g.ye@bupt.edu.cn, h.yin1@uq.edu.au)
† These authors contributed equally to this work.

1) https://gdpr-info.eu/.
2) https://oag.ca.gov/privacy/ccpa.
3) In this paper, one client represents one user; therefore, the concepts of "user" and "client" are equivalent and can be interchangeably used.

and subsequently upload the public parameter updates to a central server for aggregation. This design makes users' private data undisclosed to other participants. Owing to the privacy-preserving nature, FedRecs have garnered growing attention and demonstrated notable achievements recently [10–13].

While the decentralized and open characteristics of FedRecs offer a certain level of privacy guarantee, they simultaneously face at least two challenges. Firstly, each user's data is extremely sparse. Training on these sparse data dramatically increases the training difficulty of FedRecs and even compromises the model performance [14]. Secondly, since all participants can directly upload model updates to modify the recommender model, FedRecs become susceptible to model poisoning attacks, in which malicious users send poisoned updates to manipulate the recommender system to achieve adversarial goals (e.g., promote or demote target items) [15].

Contrastive learning stands out as one of the foremost research avenues aimed at addressing the data sparsity challenge in centralized recommender systems [16], as it excels in leveraging unlabeled data effectively. Fundamentally, a crucial aspect of contrastive recommender systems involves crafting and enhancing multiple contrastive views. Categorized by view augmentation strategies, contrastive learning within centralized recommender systems bifurcates into data augmentation-based methods and representation augmentation-based methods [17]. Regrettably, the significant disparities between centralized recommender systems and FedRecs, spanning both data and model domains, render these contrastive learning methodologies ineffective in the context of FedRecs.

Specifically, data augmentation-based contrastive learning methods predominantly entail perturbations tailored to specific data characteristics [18–21]. For example, edge and node dropout find widespread application in graph data scenarios [20], while sequence shuffling and item masking serve as common augmentation techniques for sequential data [21]. However, in the context of FedRecs, where each client typically possesses limited data, often represented as a one-hop bipartite graph or a very short single sequence, the feasibility of applying these data augmentation methods becomes highly constrained.

Representation augmentation-based contrastive learning methods directly manipulate the recommendation model to generate distinguishable views. An exemplary study by Yu et al. [22] suggests that the application of uniform noise alone can yield comparable or even superior performance to data augmentation techniques. However, our subsequent experiments uncover the inadequacy of this simplistic augmentation approach in FedRecs. This failure could stem from the severely limited data available on clients, which may not suffice for meaningful model learning via random noise perturbation. Alternatively, other methodologies construct contrastive views based on the intricate structures of the model. For instance, Ref. [23] devised contrastive views by comparing representations from different layers of the recommendation model. However, these approaches also prove unsuitable for FedRecs, where each client's local model remains rudimentary due to constraints on both on-device data and computational resources.

Given the stringent on-device constraints that hinder the integration of contrastive learning in FedRecs, recent studies [24, 25] have opted to directly amend the fundamental learning protocol. This modification allows participants to tap into other clients' private resources, such as user embeddings, thereby enabling the utilization of traditional augmentation methods. However, these endeavors raise significant privacy concerns. Other work [26] disregarded the role of user embeddings and focused solely on applying contrastive learning to item embeddings. This approach coarsely utilizes interacted items as positive samples and non-interacted items as negative samples, albeit with less effectiveness in enhancing model performance. Consequently, the development of an efficient federated contrastive recommender system that preserves privacy remains largely unexplored.

In this paper, we propose a contrastive learning framework tailored for federated recommendation, namely CL4FedRec, which incorporates contrastive learning in FedRecs without sacrificing privacy. In CL4FedRec, to facilitate user view construction, we generate a group of synthetic users on the central server, serving as negative users for all real participants. Then, inspired by [22], we incorporate lightweight uniform noises to user embeddings to create positive pairs. Based on the augmented user representations, we enhance two distinct views of item embeddings by approximately optimizing them on clients' local data. We do not directly add the uniform noises [22] to item embedding as we empirically find it cannot achieve satisfied recommendation performance. This may stem from the relatively small datasets within each FedRec client, preventing the model from learning good item representations from the augmentation based on uniform noises. It is essential to highlight that CL4FedRec functions as an auxiliary task that can be jointly optimized with the recommendation task on the client side and is consistent with the basic FedRec learning protocol. Therefore, it is compatible with most existing FedRecs.

After that, we empirically assess the robustness of FedRecs equipped with CL4FedRec using several state-of-the-art model poisoning attacks. Unexpectedly, we find that contrastive learning intensifies the susceptibility of FedRecs to these poisoning attacks. We then conduct extensive analysis, attributing this phenomenon to the uniformity of the representation distribution enforced by the contrastive learning task. Specifically, in a more uniform embedding space, adversaries can easily modify target item embeddings to mimic popular ones. In light of this, we propose a robust CL4FedRec (rCL4FedRec) by adding a regularizer in the central server. The regularizer is designed to maintain the distance among item embeddings with various popularity levels, thwarting malicious users from easily boosting the exposure rate of target items by aligning them with arbitrary popular items. To demonstrate the effectiveness of our proposed methods, we conduct extensive experiments on four recommendation datasets (MovieLens-1M [27], Amazon-Phone, Amazon-Video [28], and QB-Article [29]) with the commonly used FedRecs. The experimental results showcase that rCL4FedRec can improve the recommendation effectiveness and make FedRecs more robust to model poisoning attacks.

To sum up, the main contributions of this paper are fourfold. (1) We propose the first contrastive learning framework tailored for federated recommendation (CL4FedRec) without privacy compromise. (2) Through empirical analysis, we find that contrastive learning significantly reduces FedRecs' resistance to model poisoning attacks because of the improved uniformity of embeddings. To the best of our knowledge, this work is the first to reveal such an interesting phenomenon in federated recommendation. (3) We further propose a robust version of CL4FedRec, named rCL4FedRec, by incorporating a popularity-based contrastive regularizer. It is noteworthy that, unlike traditional defense methods, our regularizer does not compromise but enhances the recommendation effectiveness. (4) Extensive experiments show that our proposed contrastive learning frameworks for FedRecs significantly improve recommendation effectiveness and robustness to poisoning attacks.

2 Related work

2.1 Federated recommender system

The privacy-preserving ability has made FedRecs receive remarkable focus in recent years [9, 13, 30–39]. Ammad-Ud-Din et al. [10] proposed the first federated recommender system with collaborative filtering models. Building upon this foundational framework, numerous extended studies have emerged to improve the effectiveness, efficiency, and privacy protection of FedRecs [40]. For instance, some studies [12, 41] applied graph neural networks [42] to improve the FedRecs' recommendation accuracy. The efficiency-related studies mainly focus on reducing each training round's communication costs or achieving fast convergence. For example, Muhammad et al. [43] incorporated an advanced client sampling strategy to benefit model convergence in the early stage, while Zhang et al. [44] investigated hash methods [45] to light the public parameters during transmission. Privacy protection is another hot research topic in FedRecs. Chai et al. [11] introduced a user-level distributed matrix factorization FedRec framework, incorporating homomorphic encryption to enhance user privacy protection. Local differential privacy (LDP) is achieved in [46] by adding noise in public parameters to protect user behaviors, however, Refs. [47, 48] revealed that LDP still cannot well protect user information. FRU [49] explores the machine unlearning problem in FedRecs.

2.2 Attacks and defenses for federated recommender system

With notable achievements, many researchers have redirected their focus to verifying the security of FedRecs by investigating poisoning attacks [50–52]. Generally, according to the adversarial goal, poisoning attacks can be categorized into targeted attacks [15, 53–56] and untargeted attacks [26, 57]. The untargeted attacks endeavor to cause a loss of recommendation performance. FedAttack [57] designs a hard negative sampling-based strategy to undermine the utility of a recommendation model in FedRecs. Yu et al. [26] proposed a clustering adversarial objective function to make item embeddings identical, therefore, disrupting the prediction. However, in reality, untargeted attacks are easy to detect [58] and have less motivation to launch.

The targeted attacks aim to promote specific items to most users [56]. Compared with untargeted attacks, targeted attacks are more imperceptible and are more common due to financial incentives in real-world scenarios. Zhang et al. [15] presented Pipattack, the first model poisoning attack capable

of adversarially promoting items in FedRecs. However, Pipattack necessitates a substantial number of malicious users. To reduce the requirement of malicious user amounts, FedRecAttack [53] assumes that adversaries can observe a proportion of user interaction data and utilize these data to calculate high-quality poisoned gradients. Apparently, this strong assumption depresses the practical threats of FedRecAttack. A-hum [54] and PSMU [55] are two recently released model poisoning attacks. A-hum utilizes the random vector from Gaussian distribution and constructs "hard users" to compute poisoned model updates. PSMU [55] constructs malicious synthetic users with random items and further improves attack performance by leveraging alternative products. These two attacks achieve state-of-the-art attack performance with fewer malicious users and without relying on prior knowledge, making the threats of target attacks more realistic. In this paper, we will leverage these two attacks to assess the robustness of FedRecs.

2.3 Contrastive learning in recommender system

Essentially, the core concept of contrastive learning is to maximize agreement between different views. Therefore, unless multiple views naturally exist in some cases [59, 60], the primary focus of research in contrastive learning-based recommender systems is the construction of high-quality views, known as augmentation. Overall, the augmentation can be applied from two dimensions: data and representation [17]. The data-based augmentation [18–21] primarily involves perturbations based on specific data attributes. For instance, edge and node dropout are widely employed for graph data [20], while sequence shuffling, cross-sequence segmentation [61], and item masking serve as common augmentation operators for sequential data [21]. Nonetheless, in FedRecs, where each client has limited data, often in the form of a one-hop bipartite graph or a very short sequence, these data augmentation methods tend to be infeasible. Ref. [62] proposed a contrastive learning framework considering privacy, however, their work is mainly designed for semi-distributed training, which still cannot apply in FedRecs.

The representation-based augmentation aims to directly perturb the recommendation model to create distinguishable views. One notable work is [22], which suggests that by simply applying uniform noises, contrastive learning can achieve comparable or even superior performance to data augmentation. Unfortunately, our subsequent experiments reveal that this simplistic augmentation method fails to deliver good performance in FedRecs, possibly due to the exceedingly limited data in clients, which cannot support meaningful model learning from random noise perturbation. XSimGCL [23] is the advanced version of [22], but it is designed for multi-layer graph neural network (GNN)-based recommendation models, which is not appropriate for FedRecs. Wang et al. [63] was the first work exploring the security problem in contrastive recommender systems, however, they only focus on centralized recommender systems with data poisoning attacks, which is essentially different from our research topic. As a result, the contrastive learning methods in centralized recommender systems cannot be directly applied in FedRecs to achieve effective performance.

2.4 Contrastive learning in federated recommender system

While contrastive learning has demonstrated potent representation learning capabilities in centralized recommender systems [17], its application in FedRecs remains relatively unexplored. FedCL [24] is one of the first work that utilizes contrastive learning in a federated recommendation. It loses the original FedRec learning protocol by requiring clients to upload private parameters to the central server. Although they apply LDP [46] in uploaded private parameters, FedCL still raises severe privacy concerns. Other studies, such as [25, 64], followed the basic idea of FedCL and achieved contrastive learning by compromising FedRecs' privacy protection ability. Yu et al. [26] introduced a contrastive learning approach named UNION in FedRecs, however, the primary focus of UNION is to detect untargeted attacks rather than improve model performance. In this paper, we explore applying contrastive learning without reducing privacy-preserving ability in FedRecs. Furthermore, we novelly provide a security view of contrastive learning in FedRecs with target model poisoning attacks.

3 Preliminaries

In this section, we first describe the primary settings of the general federated recommender systems. Then, we briefly introduce the model poisoning threats in the federated recommendation. Table 1 lists

Table 1 List of important notations.

Notation	Description
$\mathcal{D}_i$	The local dataset for user u_i
$\mathcal{U}$	All users in the federated recommender system
$\mathcal{U}_{\text{syn}}$	Synthetic users for contrastive learning
$\mathcal{V}$	All items in the federated recommender system
r_{ij}	The preference score of user u_i for item v_j
$\hat{r}_{ij}$	The predicted score for item v_j by user u_i's local model
$\boldsymbol{u}_i$	User u_i's private parameters (i.e., user embedding)
$\boldsymbol{u}'_i, \boldsymbol{u}''_i$	User u_i's augmented user views
$\boldsymbol{s}_i$	Synthetic user s_i's user embedding
$\boldsymbol{V}_i$	User u_i's item embeddings
$\boldsymbol{V}'_i, \boldsymbol{V}''_i$	User u_i's augmented item views
$\nabla \boldsymbol{V}$	Normal item embeddings updates
$\nabla \widetilde{\boldsymbol{V}}$	Poisoned item embeddings updates
N	The number of interacted items for a synthetic user
λ_1	Factor controls the strengths of user view contrastive learning
λ_2	Factor controls the strengths of item view contrastive learning

some important notations.

3.1 General federated recommender system framework

Following most FedRec robustness studies [15,53–55], our research is based on the most general federated recommendation framework described as follows. Let $\mathcal{U}$ and $\mathcal{V}$ represent the sets of all users (clients) and items in FedRecs. Each user/client u_i possesses a local training dataset $\mathcal{D}_i$ containing a few user-item interaction records (u_i, v_j, r_{ij}). $r_{ij} = 1$ indicates that the user has interacted with item v_j, while $r_{ij} = 0$ implies v_j is a negative sample. Note that due to the data sparsity in recommender systems [65], the size of each local dataset $|\mathcal{D}_i|$ is usually small. The goal of FedRecs is to train a recommender model based on distributed datasets $\{\mathcal{D}_i\}_{u_i \in \mathcal{U}}$ that can predict users' preference scores $\hat{r}_{ij}$ for non-interacted items and make top-K recommendations based on these scores.

General FedRec learning protocol. Generally, FedRecs collaboratively learn a recommender model under the following learning protocol. A central server functions as a coordinator, determining which clients will participate in the current training round. The recommender model's parameters are categorized into two groups: public parameters and private parameters. The private parameters are user embeddings $\boldsymbol{U}$, encompassing users' sensitive attributes. Consequently, these parameters are locally managed by the respective users. Public parameters include item embeddings $\boldsymbol{V}$ and other model parameters $\boldsymbol{\Theta}$, which clients collaboratively update. In the initial stage, the central server initializes the public parameters while the clients initialize their private parameters. Subsequently, the recommender model undergoes training by repetitively executing the following steps. Firstly, the central server randomly selects a subset of users to contribute to the current round's training and disperses public parameters $\boldsymbol{V}^{t-1}$ and $\boldsymbol{\Theta}^{t-1}$ to these clients. Then, the selected clients optimize recommendation loss functions (e.g., (1)) based on the received public parameters and their private parameters:

$$\mathcal{L}^{\text{rec}} = -\sum_{(u_i, v_j, r_{ij}) \in \mathcal{D}_i} r_{ij} \log \hat{r}_{ij} + (1 - r_{ij}) \log(1 - \hat{r}_{ij}). \tag{1}$$

After the local training, clients directly update their local private parameters while sending the updates of public parameters $\nabla \boldsymbol{V}^{t-1}$ and $\nabla \boldsymbol{\Theta}^{t-1}$ back to the central server. The central server aggregates received gradients using FedAvg [66].

Base recommender. Following most research [15, 54, 55] of FedRecs' security, we employ neural collaborative filtering (NCF) [67] as the basic recommender model. It is worth noting that the above-mentioned general federated recommendation framework and our proposed methods are compatible with most deep learning-based recommenders [68, 69] since we do not make special assumptions based on recommenders.

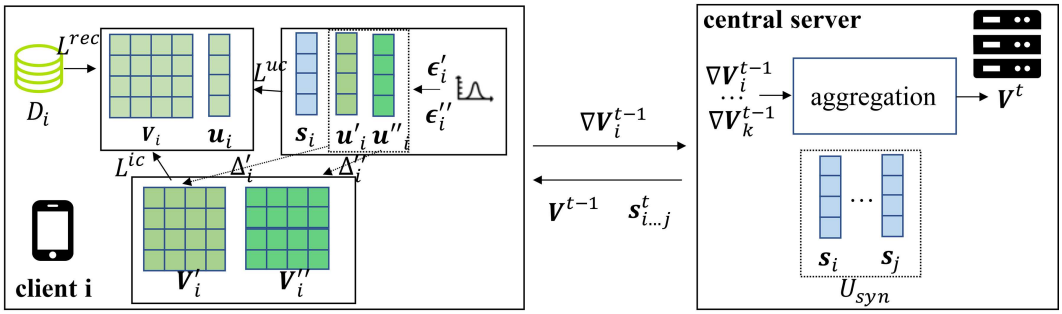

Figure 1 (Color online) Overview of CL4FedRec.

3.2 Model poisoning attacks in federated recommendation

While the federated recommendation framework outlined in Subsection 3.1 offers privacy protection for users, its open characteristics leave a backdoor for adversaries to directly alter the recommender model's public parameters via uploading poisoned gradients, namely model poisoning attacks [15]. In this paper, we mainly focus on targeted model poisoning attacks [15,53–55], which aim to promote target items $\widetilde{\mathcal{V}}$ with financial incentives. These kinds of attacks are more imperceptible and will cause unfair recommendation [53], therefore, they are more harmful for real recommender systems. Eq. (2) formally defines these model poisoning attacks,

$$\begin{aligned} &\underset{\{\nabla\widetilde{\boldsymbol{V}}^t,\nabla\widetilde{\boldsymbol{\Theta}}^t\}_{t=s}^{T-1}}{\operatorname{argmax}} \mathrm{ER@}K(\widetilde{\mathcal{V}}|\boldsymbol{U}^{\mathrm{T}},\boldsymbol{V}^{\mathrm{T}},\boldsymbol{\Theta}^{\mathrm{T}}),\\ &\mathrm{ER@}K=\frac{1}{|\widetilde{\mathcal{V}}|}\sum_{v_j\in\widetilde{\mathcal{V}}}\frac{|\{u_i\in\mathcal{U}|v_j\in\hat{\mathcal{V}}_i\wedge v_j\in\mathcal{V}_i^-\}|}{|\{u_i\in\mathcal{U}|v_j\in\mathcal{V}_i^-\}|}, \end{aligned} \tag{2}$$

where $\hat{\mathcal{V}}_i$ is the recommendation result for user u_i. $\mathcal{V}_i^-$ is the set of items that u_i has not interacted with. $\nabla\widetilde{\boldsymbol{V}}^t$ and $\nabla\widetilde{\boldsymbol{\Theta}}^t$ are poisoned gradients uploaded by compromised clients. ER@K is the exposure ratio. It reflects the proportion of users for whom the target items appear in their top-K recommendation lists. As of now, Refs. [54,55] are two state-of-the-art model poisoning attacks that achieve high ER@K scores with fewer assumptions, thereby exposing the vulnerability of FedRecs in scenarios that closely resemble real-world conditions. Therefore, we employ these two attacks to evaluate the robustness of FedRecs.

4 Methodology

This section presents our contrastive learning framework customized for federated recommender systems. Specifically, we first introduce CL4FedRec, consisting of user and item contrastive learning. Subsequently, we empirically find that although CL4FedRec improves the recommendation effectiveness, it exacerbates the vulnerability of FedRecs to model poisoning attacks. We propose a popularity-based regularizer to address this security concern, resulting in a robust version of CL4FedRec, defined as rCL4FedRec. Figure 1 and Algorithm 1 outline rCL4FedRec in a generic FedRec.

4.1 CL4FedRec

4.1.1 *User contrastive learning*

Given a user u_i, obtaining positive and negative samples for the user in FedRecs is challenging since the learning protocol strictly constrains that a client cannot access other clients' embeddings and private data. Refs. [24, 25] radically overlooked the protocol requirement to enable user embeddings shared so that the conventional contrastive learning methods can be simply applied in their studies. However, the exposure of user embeddings will cause severe privacy leakage as it violates the basic FedRec learning protocol [47, 48]. In light of this, we propose a privacy-preserving user contrastive learning method.

Negative user sample construction. To facilitate clients construct negative users, the central server in CL4FedRec manages a group of synthetic users $\mathcal{U}_{\mathrm{syn}}$. All the normal clients can access these synthetic users' private parameters and utilize them as negative users. Specifically, the central server

Algorithm 1 FedRec with rCL4FedRec.

Input: global epoch T, local epoch L, learning rate lr, client batch size B, ...;
Output: public parameters $\boldsymbol{V}, \boldsymbol{\Theta}$, private parameters $\boldsymbol{u}_i|_{i\in\mathcal{U}}$;
Initialize global parameter $\boldsymbol{V}_0, \boldsymbol{\Theta}_0$;
Construct synthetic users $\mathcal{U}_{\text{syn}}$ using (3);
for training round $t = 1, \ldots, T$ **do**
  Shuffle and divide $\mathcal{U}$ into $\{\mathcal{U}_k\}_{k=1,\ldots,|\mathcal{U}|/B}$;
  for $k = 1, \ldots, |\mathcal{U}|/B$ **do**
    for $u_i \in \mathcal{U}_k$ **in parallel do**
      Augment user views using (5);
      Augment item views using (7) and (8);
      $\nabla \boldsymbol{V}_i^{t-1}, \nabla \boldsymbol{\Theta}_i^{t-1}, \nabla \boldsymbol{u}_i^{t-1} \leftarrow$ train L epochs on $\mathcal{D}_i$ using (10);
      $\boldsymbol{u}_i^t \leftarrow$ update private parameter using $\nabla \boldsymbol{u}_i^{t-1}$;
      Upload $\nabla \boldsymbol{V}_i^{t-1}, \nabla \boldsymbol{\Theta}_i^{t-1}$;
    end for
    $\boldsymbol{V}^{t-1}, \boldsymbol{\Theta}^{t-1} \leftarrow$ aggregate public parameter updates;
  end for
  $\boldsymbol{V}^t \leftarrow$ calibrate item embedding using (13);
  $\{\boldsymbol{s}_j^{t+1}\}_{s_j\in\mathcal{U}_{\text{syn}}} \leftarrow$update synthetic users using (4).
end for

selects a set of items for each synthetic user s_i as its interacted items. To align the data distribution of synthetic users with that of normal users (i.e., power-law distribution of interaction data), the items are randomly selected, taking into account item popularity:

$$\mathcal{D}_{\text{syn},i} \leftarrow \text{random}(\mathcal{V}|\text{popularity}(\mathcal{V}), N), \tag{3}$$

where N is the number of interacted items for a synthetic user. popularity($\mathcal{V}$) is the popularity information of each item, which is one of the common commercial statistics available in many real-world applications (e.g., video view counts on YouTube and music listen counts on Spotify).

Then, the central server calculates synthetic user embedding by optimizing the recommendation loss function based on these synthetic data and the up-to-date public parameters:

$$\boldsymbol{s}_i^t \leftarrow \text{argmin}\, \mathcal{L}^{\text{rec}}(\boldsymbol{s}_i^{t-1}|\boldsymbol{V}^{t-1}, \boldsymbol{\Theta}^{t-1}, \mathcal{D}_{\text{syn},i}). \tag{4}$$

It is worth mentioning that Eq. (4) only updates synthetic user embedding and will not influence the public parameters, as these synthetic users are developed to simulate negative user samples. After that, these synthetic user embeddings $\{\boldsymbol{s}_i^t\}_{s_i\in\mathcal{U}_{\text{syn}}}$ are dispersed to normal clients to function as negative user embeddings.

User view augmentation for positive sample construction. Inspired by [22], the normal client augments its user embedding $\boldsymbol{u}_i$ by adding noise vectors:

$$\boldsymbol{u}_i' = \boldsymbol{u}_i + \boldsymbol{\epsilon}_i', \quad \boldsymbol{u}_i'' = \boldsymbol{u}_i + \boldsymbol{\epsilon}_i''. \tag{5}$$

We omit the time index to make the formula clear. $\boldsymbol{\epsilon}$ is based on the noise vector sampled from the uniform distribution $\bar{\boldsymbol{\epsilon}} \sim \text{uniform}(\boldsymbol{0}, \boldsymbol{I})$ with the following constraints to avoid too much deviation of augmented representation: $\boldsymbol{\epsilon} = \bar{\boldsymbol{\epsilon}} \odot \text{sign}(\boldsymbol{u}_i)$ and $\|\boldsymbol{\epsilon}\|_2 = \eta$.

User contrastive objective. Based on the synthetic user embeddings $\{\boldsymbol{s}_i^t\}_{s_i\in\mathcal{U}_{\text{syn}}}$ and augmented user views $\boldsymbol{u}_i'$ and $\boldsymbol{u}_i''$, the client u_i optimizes the following loss function:

$$\mathcal{L}^{\text{uc}} = -\log \frac{\exp(\text{sim}(\boldsymbol{u}_i', \boldsymbol{u}_i'')/\tau)}{\exp(\text{sim}(\boldsymbol{u}_i', \boldsymbol{u}_i'')/\tau) + \sum_{s_j\in\mathcal{U}_{\text{syn}}} \exp(\text{sim}(\boldsymbol{u}_i, \boldsymbol{s}_j)/\tau)}, \tag{6}$$

where $\text{sim}(x, y)$ is the similarity score of x and y calculated using cosine similarity and τ is the temperature hyper-parameter. In (6), only $\boldsymbol{u}_i$ is trainable while all synthetic user embeddings are frozen.

4.1.2 *Item contrastive learning*

Compared with users, the negative item samples are relatively straightforward as each client often has more than one training item in $\mathcal{D}_i$. In this paper, for one training item, we simply treat the other items in the training set as negative items as these items refer to different entities.

Item view augmentation. A direct approach to augment item representation involves the application of (5) to item embeddings. However, our empirical findings indicate that the performance gains from this

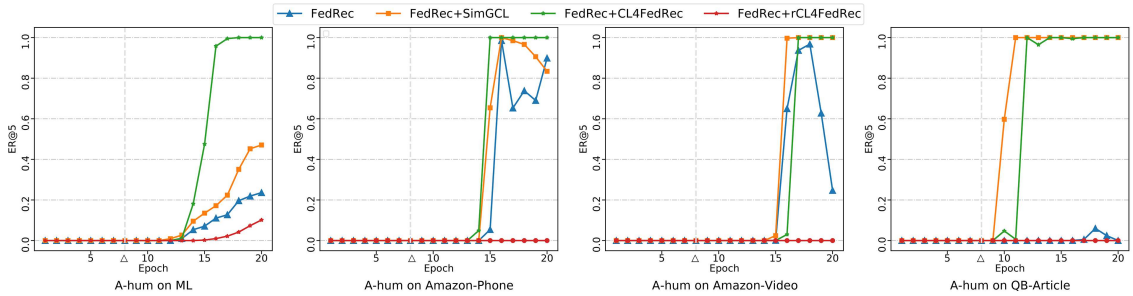

Figure 2 (Color online) ER@5 scores of A-hum attack FedRec with and without contrastive learning methods on different recommendation datasets.

method are limited. This limitation may arise due to the scarcity of training data on each client side, preventing the model from effectively learning meaningful knowledge amidst the presence of relatively inconsequential uniform noise. In response to this challenge, we introduce a more potent technique by augmenting item views with more purposeful "noise". Specifically, based on the augmented user representations $\boldsymbol{u}_i'$ and $\boldsymbol{u}_i''$, CL4FedRec can obtain the corresponding item embedding updates as follows:

$$\boldsymbol{\Delta}' \leftarrow \operatorname{argmin} \mathcal{L}^{\mathrm{rec}}(\boldsymbol{V}|\boldsymbol{u}_i', \boldsymbol{\Theta}, \mathcal{D}_i), \quad \boldsymbol{\Delta}'' \leftarrow \operatorname{argmin} \mathcal{L}^{\mathrm{rec}}(\boldsymbol{V}|\boldsymbol{u}_i'', \boldsymbol{\Theta}, \mathcal{D}_i). \tag{7}$$

Note that since Eq. (7) is only employed for item view augmentation, there is no need to meticulously compute updates over multiple epochs. In contrast, we opt to perform a single forward and backward pass to approximate the updates. This strategy is adopted to mitigate the burden of high computational costs. Subsequently, the item view is augmented as follows:

$$\boldsymbol{V}_i' = \boldsymbol{V}_i + \boldsymbol{\Delta}_i', \quad \boldsymbol{V}_i'' = \boldsymbol{V}_i + \boldsymbol{\Delta}_i''. \tag{8}$$

Item contrastive objective. Based on the augmented item views, CL4FedRec optimizes the following contrastive loss function:

$$\mathcal{L}^{\mathrm{ic}} = \sum_{v_k \in \mathcal{D}_i} -\log \frac{\exp(\mathrm{sim}(\boldsymbol{v}_k', \boldsymbol{v}_k'')/\tau)}{\sum_{v_j \in \mathcal{D}_i} \exp(\mathrm{sim}(\boldsymbol{v}_k', \boldsymbol{v}_j'')/\tau)}. \tag{9}$$

4.1.3 *Joint learning with recommendation task*

To improve the recommendation performance, we jointly train the contrastive learning tasks (Eqs. (6) and (9)) with the recommendation task (Eq. (1)) on each client device:

$$\mathcal{L} = \mathcal{L}^{\mathrm{rec}} + \lambda_1 \mathcal{L}^{\mathrm{uc}} + \lambda_2 \mathcal{L}^{\mathrm{ic}}, \tag{10}$$

where λ_1 and λ_2 are hyper-parameters to control the strengths of user contrastive learning and item contrastive learning, respectively.

4.2 Robust CL4FedRec against model poisoning attacks

The empirical results discussed in Subsection 5.5 demonstrate a noteworthy improvement of FedRecs in the recommendation performance brought by CL4FedRec. As FedRecs are susceptible to model poisoning attacks, we take a further step to assess the robustness of CL4FedRec.

4.2.1 *Robustness problem*

We employ one of the state-of-the-art model poisoning attacks, A-hum [54], to attack FedRecs enhanced by various contrastive learning methods to highlight their vulnerability. As illustrated in Figure 2, compared with the original FedRecs, A-hum achieves higher and rapidly increasing ER@5 scores in FedRecs with contrastive learning methods across all four datasets. This observation suggests that simply incorporating contrastive learning methods renders FedRecs more susceptible to model poisoning attacks. In Subsection 5.6, we will employ more state-of-the-art model poisoning attacks to assess the robustness of FedRecs with various contrastive learning methods.

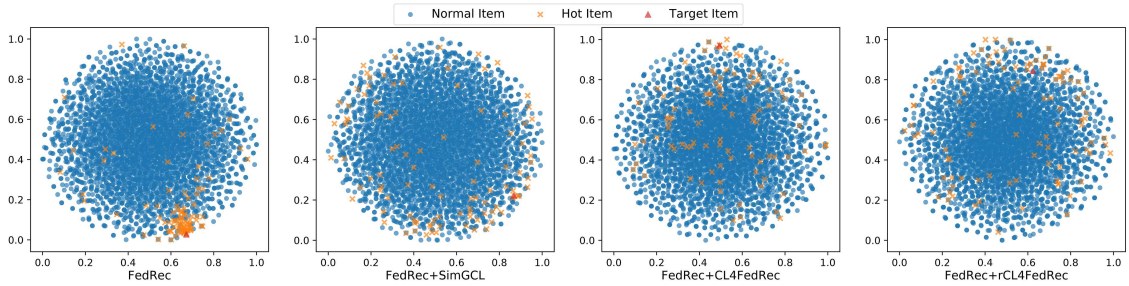

Figure 3 (Color online) Item representation distribution on MovieLens-1M for FedRecs with and without different contrastive learning methods. We define the top 100 items that have the most number of interactions as "hot item".

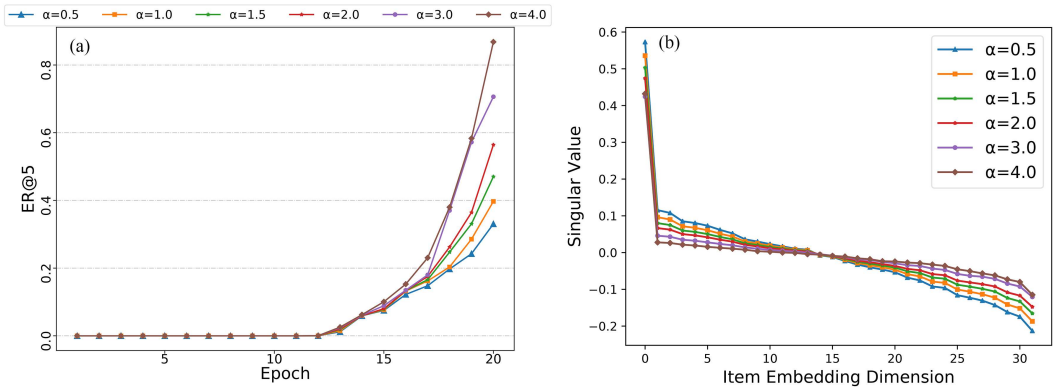

Figure 4 (Color online) Proof-of-concept. More uniform distribution (i.e., larger α) weakens the robustness of FedRecs to model poisoning attacks. The results are from MovieLens-1M but the similar conclusion can be obtained from the other three datasets. (a) The FedRec with different strengths of $\mathcal{L}^{\text{uni}}$; (b) the sorted singular value of item embedding table for FedRec with different strengths of $\mathcal{L}^{\text{uni}}$.

4.2.2 *Emperical analysis*

We posit that this phenomenon arises from the uniformity and dispersion of embedding distribution induced by contrastive learning [22, 70]. Figure 3 plots different FedRecs' item embeddings trained on MovieLens-1M, and similar distribution can also be observed on the other three datasets. In the original FedRec, most hot items' embeddings are densely clustered. Consequently, if an adversary aims to elevate a target item's popularity, it must "precisely" shift the embedding of the target item closer to the cluster of popular items. However, when integrating a contrastive learning task, popular item embeddings are globally dispersed among normal item embeddings, as depicted in the three right figures in Figure 3. Since the embeddings of popular items are scattered in the embedding space, the adversary has more opportunities to manipulate the embedding of its target item to disguise it as a popular item.

To further validate our claim that the distribution uniformity makes FedRecs vulnerable, we provide a proof-of-concept in Figure 4. Specifically, inspired by [37, 71, 72], we employ the following formula to directly adjust the "uniformity" of item embeddings:

$$\mathcal{L}^{\text{uni}} = \frac{1}{d} \left\| \text{corr} \left(\frac{\boldsymbol{V} - \bar{\boldsymbol{V}}}{\sqrt{\text{var}(\boldsymbol{V})}} \right) \right\|_F, \tag{11}$$

where d is the dimension of item embeddings, $\text{corr}(\cdot)$ is the function that calculates the correlation matrix, $\bar{\boldsymbol{V}}$ is the mean of each dimension, $\text{var}(\cdot)$ computes the variance of a matrix, and $\|\cdot\|_F$ represents the Frobenius norm. In the proof-of-concept, the client optimizes $\mathcal{L}^{\text{uni}}$ with the recommendation loss function as follows:

$$\mathcal{L}^{\text{poc}} = \mathcal{L}^{\text{rec}} + \alpha \mathcal{L}^{\text{uni}}, \tag{12}$$

where α controls the strength of the uniformity penalty. As demonstrated in Figure 4(b), with larger α, the variance of each dimension's singular value is diminished, i.e., the curve becomes less steep, indicating a more uniform embedding table. Then, we employ A-hum to assess the system's vulnerability with varying α. As depicted in Figure 4(a), it becomes evident that as α increases, A-hum achieves

higher ER@5 scores. This observation confirms our assertion that the uniformity of distribution weakens the resilience of FedRecs against model poisoning attacks.

4.2.3 *Solution: popularity-based contrastive regularizer*

Based on the finding that the uniformity and dispersion of item embeddings enable adversaries to easily push the target item's representation close to popular items, we propose a popularity-based regularizer to calibrate item embeddings. In detail, the central server divides the items into hot items and unpopular items according to the popularity information. Then, at the end of each global epoch, the central server updates the item embeddings by optimizing the following contrastive loss function:

$$\mathcal{L}^{\text{reg}} = \sum_{v_i \in \mathcal{V}^{\text{hot}}} -\log \frac{\exp(1/\tau)}{\sum_{v_j \in \mathcal{V}^{\text{sp}}} \exp(\text{sim}(\boldsymbol{v}_i, \boldsymbol{v}_j)/\tau)}, \tag{13}$$

where $\mathcal{V}^{\text{hot}}$ is the set of hot items and $\mathcal{V}^{\text{sp}}$ is the subset of items sampled from the unpopular item set. Eq. (13) pulls the unpopular items away from hot items. Therefore, malicious users cannot easily push their target items to mimic popular items.

5 Experiments

In this section, we conduct extensive experiments to answer the following research questions (RQs).

- **RQ1.** How effective are our proposed federated contrastive learning methods (CL4FedRec and rCL4FedRec) to improve the recommendation performance?
- **RQ2.** How robust are our proposed federated contrastive learning methods (CL4FedRec and rCL4FedRec) to state-of-the-art model poisoning attacks? As a new defense method, how effective is our proposed popularity-based contrastive regularizer in improving the robustness of FedRecs compared with existing defense baselines?
- **RQ3.** What is the contribution of different components in rCL4FedRec to the recommendation performance?
- **RQ4.** What is the impact of the most hyper-parameter (i.e., synthetic user size) to rCL4FedRec?

5.1 Datasets

We conduct experiments on four popular recommendation datasets: MovieLens-1M [27], Amazon-Phone, Amazon-Video [28], and QB-Article [29], covering various platforms and recommendation domains. MovieLens-1M includes 1000208 interaction records between 6040 users and 3706 movies. Amazon-Phone has 13174 users, 5970 cell phones, and 103593 feedbacks. There are 63836 interactions involving 8072 users and 11830 items in Amazon-Video. QB-Article contains 10981 users, 6493 articles, and 335663 reading records. Following the common settings in implicit feedback recommendation [54, 55, 67], all users' feedback ratings are transformed to $r_{ij} = 1$, and negative instances are sampled with $1:4$ ratio. Besides, we leave 20% data for testing, and 10% data are sampled from training data for validation.

5.2 Evaluation protocol

In this paper, we employ two widely used metrics [22], Recall at rank 20 (Recall@20) and Normalized Discounted Cumulative Gain at rank 20 (NDCG@20), to measure the recommendation effectiveness. Recall reflects the average probability of ground-truth items successfully appearing in users' recommendation lists, while NDCG considers the position of ground-truth items. We rank all items when calculating the two metrics.

To evaluate the robustness of FedRecs, we primarily employ two state-of-the-art and open-source model poisoning attacks, selected for their ability to achieve remarkable performance without requiring extensive prior knowledge.

- A-hum [54]: This method utilizes "hard users" who consider the target items as negative samples to generate poisoned gradients to optimize (2).
- PSMU [55]: This method leverages randomly constructed users to generate poisoned gradients and further improve target items' competition by adding their alternative items to optimize (2).

Table 2 Recommendation effectiveness comparison of various contrastive learning methods on four recommendation datasets. The best results are in bold.

Method	MovieLens-1M		Amazon-Phone		Amazon-Video		QB-Article	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
Original	0.04444	0.06372	0.05909	0.02571	0.04626	0.01757	0.06244	0.03345
SimGCL	0.04619	0.06252	0.05962	0.02601	0.04897	0.01994	0.06838	0.03512
UNION	0.03334	0.05606	0.05860	0.02637	0.04954	0.02059	0.06105	0.03582
CL4FedRec	0.04656	0.06842	**0.05974**	0.02654	0.05350	0.02011	**0.07413**	0.03915
rCL4FedRec	**0.04836**	**0.07103**	0.05932	**0.02782**	**0.05667**	**0.02117**	0.06962	**0.04647**

In line with the original papers describing these attacks [54, 55], we use the exposure ratio at rank 5 (ER@5) to quantify the effectiveness of the attacks. Higher and rapidly increasing ER@5 scores indicate more potent attacks and, consequently, reduce the robustness of the FedRecs.

5.3 Baselines

Contrastive learning baselines. As discussed in Subsection 2.3, most contrastive learning methods in centralized recommender models cannot work in FedRecs due to the extreme sparsity of local data on a client. Besides, as we focus on applying contrastive learning without compromising user privacy, those federated contrastive learning studies [24, 25] that need to share user embeddings are not considered in our baselines for fair comparison.

- **Original.** This method shows the original state (i.e., recommendation effectiveness and robustness) of FedRecs without using any contrastive learning.
- **SimGCL [22].** This is the state-of-the-art contrastive learning method for centralized recommender systems, and it can directly augment the user/item embeddings by adding random noise. Note that although SimGCL has an advanced version, XSimGCL [23], it is designed for graph recommender systems and is difficult to apply in FedRecs without sacrificing privacy. Therefore, we utilize SimGCL as the baseline.
- **UNION [26].** This is the only contrastive learning method in FedRecs that does not sacrifice user privacy. It naively treats clients' all interacted items as positive samples and non-interacted items as negative samples to build item views, and it only considers the item contrastive learning.

Defense baselines. We compare our proposed defense method, the popularity-based regularizer, with the following defense baselines by integrating them into CL4FedRec.

- **Krum [73].** We mainly apply Krum to the public parameters (e.g., item embeddings) on the server; i.e., we select the gradient of an item that is closest to the mean of all other clients' uploaded gradients of this item as the aggregated gradient.
- **Median [74].** This method chooses the median value of sorted gradients as the aggregated gradients.
- **Trimmed Mean [74].** This method removes a parameter's largest and smallest gradients and aggregates the remaining gradients.
- **HiCS [55].** This is a gradient-clipping-based defense tailored for FedRecs. It adopts gradient clipping with sparsification updates to limit the contributions of malicious users.

5.4 Implementation details

The user and item embedding sizes in FedRecs are set to 32. Three feedforward layers with dimensions $64, 32, 16$ are used to process the concatenated user and item embeddings. Adam [75] with 0.001 learning rate is adopted as the optimizer. For CL4FedRec, the synthetic users' interacted item size N is set to 30. $|\mathcal{U}_{\text{syn}}|$ is 60 for MovieLens-1M and QB-Article while 10 for Amazon-Phone and Amazon-Video, respectively. η is 0.1, τ equals 0.2, and $\lambda_1 = \lambda_2 = 0.5$. All the FedRecs are converged within 20 global epochs. All the settings of model poisoning attacks follow the original paper [54, 55] with 0.1% malicious users. In addition, we also perform the sensitivity analysis of key hyper-parameters in Subsection 5.8.

5.5 Recommendation effectiveness (RQ1)

Table 2 presents a comparison of the recommendation performance between our proposed methods and contrastive baselines. Specifically, when incorporating UNION, the performance of FedRec either remains unchanged or even declines on most datasets. This is attributed to the overly simplistic item views

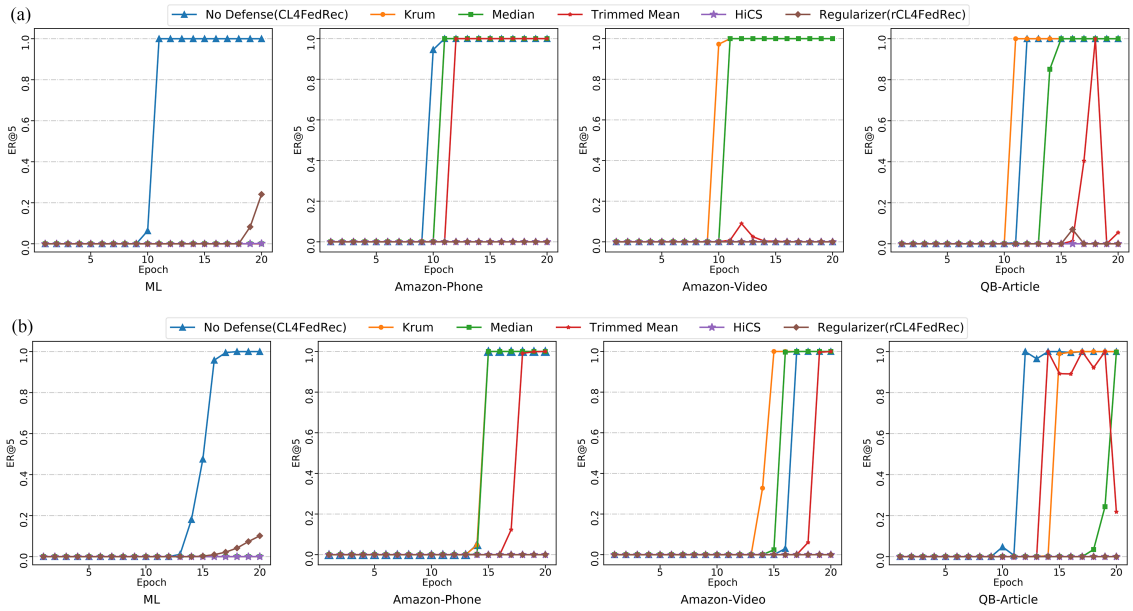

Figure 5 (Color online) Comparison of our regularizer with other defense baselines against state-of-the-art attacks. (a) Regularizer and defense baselines against PSMU; (b) regularizer and defense baselines against A-hum.

Table 3 Impact of different defenses on recommendation performance. The best results are in bold.

Method	MovieLens-1M		Amazon-Phone		Amazon-Video		QB-Article	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
CL4FedRec	0.04656	0.06842	**0.05974**	0.02654	0.05350	0.02011	**0.07413**	0.03915
+Krum	0.02794	0.05004	0.03821	0.01600	0.03298	0.01330	0.04901	0.02468
+Median	0.02772	0.05062	0.03710	0.01574	0.03095	0.01150	0.0476	0.02542
+Trimmed Mean	0.04308	0.06743	0.05695	0.02460	0.05222	0.01799	0.06527	0.03903
+HiCS	0.04437	0.06984	0.05908	0.02775	0.05317	0.01990	0.06834	0.03847
+Regularizer	**0.04836**	**0.07103**	0.05932	**0.02782**	**0.05667**	**0.02117**	0.06962	**0.04647**

in UNION and the lack of consideration for user contrastive learning. Although SimGCL yields some improvements for FedRec, the gains are limited as the item view augmentation in SimGCL introduces only meaningless uniform noises. Notably, the performance of CL4FedRec surpasses all baselines across all datasets, as evident from both Recall and NDCG scores, underscoring the superiority of our contrastive learning framework in FedRecs. Furthermore, the integration of the popularity-based contrastive regularizer in CL4FedRec (i.e., rCL4FedRec) results in further performance improvement. In summary, our proposed methods, including CL4FedRec and rCL4FedRec, effectively enhance the original FedRec's performance.

5.6 Recommendation robustness (RQ2)

One of the contributions in this paper is the popularity-based contrastive regularizer, designed to enhance the robustness of CL4FedRec. In this subsection, we compare our regularizer with some defense baselines. In general, an effective defense plugin should meet the following two requirements: (1) it can reduce the performance of attacks; (2) it does not compromise model performance.

In Figure 5, we execute two state-of-the-art attacks (A-hum [54] and PSMU [55]) for CL4FedRec, and then, we utilize several commonly used defenses to against these attacks. For these two attacks, only our regularizer and the state-of-the-art defense baseline HiCS can successfully keep the target items' ER@5 scores at zero in most cases. Other baselines cannot consistently protect CL4FedRec on various datasets.

In Table 3, we further investigate the effects of various defenses on recommendation performance. According to the results, Krum, Median, and Trimmed Mean severely compromise the recommender model, reducing these methods' utility. HiCS slightly diminishes model performance in some cases (e.g., Amazon-Video and QB-Article). Our regularizer is the only defense that significantly improves the model's performance.

Table 4 Impact of regularizer, user contrastive learning, and item contrastive learning on recommendation performance on MovieLens-1M. The best results are in bold.

Method	Recall@20	NDCG@20
rCL4FedRec	**0.04836**	**0.07103**
-regularizer	0.04656	0.06842
-user contrastive learning	0.04793	0.06923
-item contrastive learning	0.04637	0.06656

Table 5 Impact of different user and item view constructions on recommendation performance on MovieLens-1M. The best results are in bold.

Method	Recall@20	NDCG@20
rCL4FedRec	**0.04836**	**0.07103**
Random synthetic users	0.04629	0.06902
SimGCL-based item views	0.04679	0.06365

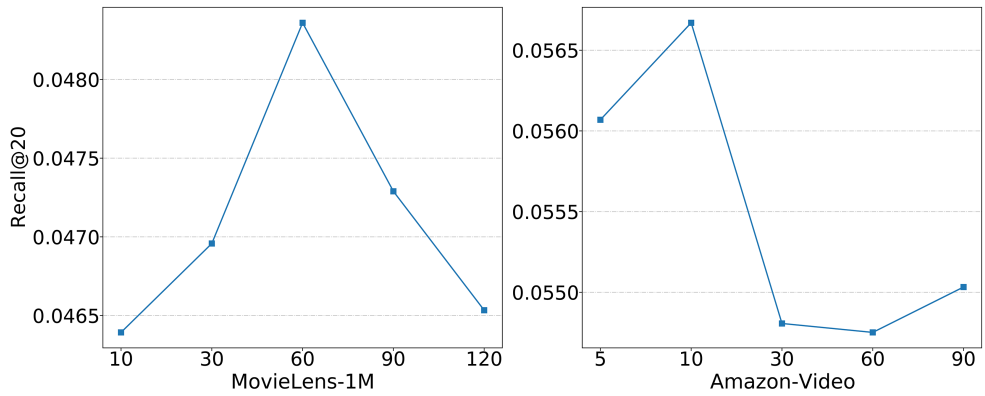

Figure 6 (Color online) Performance trend w.r.t. the change of synthetic user numbers $|\mathcal{U}_{\text{syn}}|$ on MovieLens-1M and Amazon-Video.

5.7 Ablation study (RQ3)

In this subsection, we assess the contributions of different components to recommendation performance in rCL4FedRec. rCL4FedRec comprises three main components: user contrastive learning, item contrastive learning, and the popularity-based contrastive regularizer. In Table 4, we individually remove these three components to demonstrate their impacts. As we can see, removing any one of the components results in a performance drop, indicating that all components contribute to the enhanced performance. Specifically, when removing the regularizer or item contrastive learning, the model's performance decreases from 0.048 to around 0.046 Recall@20 scores, while eliminating user contrastive learning causes about a 0.005 Recall@20 score drop. Due to space limitations, we only present the results on MovieLens-1M, but a similar conclusion can also be observed on the other three datasets.

Besides, rCL4FedRec employs a popularity-based synthetic negative user construction method (i.e., (3) and (4)) and an approximately optimization-based item view augmentation (i.e., (7) and (8)) for user and item contrastive learning. Therefore, we investigate the impacts of these two methods in Table 5. Specifically, we replace the popularity-based synthetic users with randomly constructed users and utilize SimGCL to replace our item view augmentation, respectively. According to Table 5, using these naive view construction methods cannot achieve comparable performance to rCL4FedRec, indicating the effectiveness of our user and item view construction methods.

5.8 Hyper-parameter analysis (RQ4)

The size of synthetic users (i.e., $|\mathcal{U}_{\text{syn}}|$) constructed for contrastive learning is a crucial hyper-parameter significantly influencing model performance. Figure 6 shows the performance trends concerning $|\mathcal{U}_{\text{syn}}|$. According to the results, on MovieLens-1M, the model performance exhibited a positive correlation with the number of synthetic users until it reached 60. Beyond this threshold, the performance started to decrease. This is attributed to the fact that with fewer synthetic users, normal users struggle to learn

from the negative users. While the synthetic users are too many, the contrastive learning is overwhelmed, impeding users learn recommendation knowledge. On Amazon-Video, the optimal recommendation performance is achieved when $|\mathcal{U}_{\text{syn}}|$ is 10. Note that due to space limitations, we only present the results on MovieLens-1M and Amazon-Video. The trend on QB-Article mirrors that of MovieLens-1M, and the trend on Amazon-Phone is similar to that on Amazon-Video.

6 Conclusion and future work

In this paper, we introduce a contrastive learning framework tailored for federated recommender systems, namely CL4FedRec, which designs user contrastive views by constructing synthetic users and constructs item contrastive views via approximate optimization. Subsequently, we empirically observe that incorporating contrastive learning reduces the robustness of FedRecs under model poisoning attacks. We attribute this phenomenon to the uniformity of the item embedding distribution. To address this, we propose a popularity-based contrastive regularizer for CL4FedRec, forming the robust version (rCL4FedRec). Extensive experiments conducted on four datasets demonstrate the effectiveness and robustness of our proposed methods.

Apart from model poisoning attacks, there are many various attacks and threats for FedRecs, such as data poisoning attacks that launch attacks via injecting poisoned data, membership inference attacks, and attribute inference attacks that aim to steal users' private information. In future work, we can conduct a more comprehensive analysis of these threats to validate the trustworthiness of FedRecs in the context of contrastive learning.

Acknowledgements This work was supported by Australian Research Council under the Streams of Future Fellowship (Grant No. FT210100624), Discovery Project (Grant No. DP240101108), and Linkage Project (Grant No. LP230200892).

References

1 Yin H, Cui B, Chen L, et al. Dynamic user modeling in social media systems. ACM Trans Inf Syst, 2015, 33: 1–44
2 Wang H, Fu Y, Wang Q, et al. A location-sentiment-aware recommender system for both home-town and out-of-town users. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017. 1135–1143
3 Zheng R, Qu L, Cui B, et al. AutoML for deep recommender systems: a survey. ACM Trans Inf Syst, 2023, 41: 1–38
4 Wei K, Huang J, Fu S. A survey of e-commerce recommender systems. In: Proceedings of International Conference on Service Systems and Service Management, 2007. 1–5
5 Wu F, Qiao Y, Chen J H, et al. MIND: a large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. 3597–3606
6 Zhang S, Yao L, Sun A, et al. Deep learning based recommender system. ACM Comput Surv, 2020, 52: 1–38
7 Lam S K, Frankowski D, Riedl J. Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. In: Proceedings of International Conference on Emerging Trends in Information and Communication Security, 2006. 14–29
8 Kairouz P, McMahan H B, Avent B, et al. Advances and open problems in federated learning. FNT Machine Learn, 2021, 14: 1–210
9 Yang L, Tan B, Zheng V W, et al. Federated recommendation systems. In: Federated Learning. Cham: Springer, 2020. 225–239
10 Ammad-Ud-Din M, Ivannikova E, Khan S A, et al. Federated collaborative filtering for privacy-preserving personalized recommendation system. 2019. ArXiv:1901.09888
11 Chai D, Wang L, Chen K, et al. Secure federated matrix factorization. IEEE Intell Syst, 2020, 36: 11–20
12 Wu C, Wu F, Lyu L, et al. A federated graph neural network framework for privacy-preserving personalization. Nat Commun, 2022, 13: 3091
13 Sun Z, Xu Y, Liu Y, et al. A survey on federated recommendation systems. 2022. ArXiv:2301.00767
14 Wang Q, Yin H, Chen T, et al. Fast-adapting and privacy-preserving federated recommender system. VLDB J, 2022, 31: 877–896
15 Zhang S, Yin H, Chen T, et al. PipAttack: poisoning federated recommender systems for manipulating item promotion. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining, 2022. 1415–1423

16 Yu J, Yin H, Xia X, et al. Self-supervised learning for recommender systems: a survey. IEEE Trans Knowl Data Eng, 2024, 36: 335–355

17 Jing M, Zhu Y, Zang T, et al. Contrastive self-supervised learning in recommender systems: a survey. 2023. ArXiv:2303.09902

18 Xia X, Yin H, Yu J, et al. Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 4503–4511

19 Yu J, Yin H, Li J, et al. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: Proceedings of the Web Conference, 2021. 413–424

20 Wu J, Wang X, Feng F, et al. Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021. 726–735

21 Xie X, Sun F, Liu Z, et al. Contrastive learning for sequential recommendation. In: Proceedings of IEEE 38th International Conference on Data Engineering, 2022. 1259–1273

22 Yu J, Yin H, Xia X, et al. Are graph augmentations necessary? Simple graph contrastive learning for recommendation. In: Proceedings of the 45th international ACM SIGIR Conference on Research and Development in Information Retrieval, 2022. 1294–1303

23 Yu J, Xia X, Chen T, et al. XSimGCL: towards extremely simple graph contrastive learning for recommendation. IEEE Trans Knowl Data Eng, 2023, 36: 913–926

24 Wu C, Wu F, Qi T, et al. FedCL: federated contrastive learning for privacy-preserving recommendation. 2022. ArXiv:2204.09850

25 Luo S, Xiao Y, Zhang X, et al. PerFedRec++: enhancing personalized federated recommendation with self-supervised pre-training. 2023. ArXiv:2305.06622

26 Yu Y, Liu Q, Wu L, et al. Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 4854–4863

27 Harper F M, Konstan J A. The MovieLens datasets: history and context. ACM Trans Interact Intell Syst, 2016, 5: 1–19

28 McAuley J, Targett C, Shi Q, et al. Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015. 43–52

29 Yuan G, Yuan F, Li Y, et al. Tenrec: a large-scale multipurpose benchmark dataset for recommender systems. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022. 35: 11480–11493

30 Yin H, Qu L, Chen T, et al. On-device recommender systems: a comprehensive survey. 2024. ArXiv:2401.11441

31 Liu R, Cao Y, Wang Y, et al. PrivateRec: differentially private model training and online serving for federated news recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023. 4539–4548

32 Yu S L, Liu Q, Wang F, et al. Federated news recommendation with fine-grained interpolation and dynamic clustering. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023. 3073–3082

33 Long J, Chen T, Nguyen Q V H, et al. Decentralized collaborative learning framework for next POI recommendation. ACM Trans Inf Syst, 2023, 41: 1–25

34 Long J, Chen T, Ye G, et al. Physical trajectory inference attack and defense in decentralized poi recommendation. 2024. ArXiv:2401.14583

35 Liu Z, Yang L, Fan Z, et al. Federated social recommendation with graph neural network. ACM Trans Intell Syst Technol, 2022, 13: 1–24

36 Imran M, Yin H, Chen T, et al. ReFRS: resource-efficient federated recommender system for dynamic and diversified user preferences. ACM Trans Inf Syst, 2023, 41: 1–30

37 Yuan W, Qu L, Cui L, et al. HeteFedRec: federated recommender systems with model heterogeneity. 2023. ArXiv:2307.12810

38 Yuan W, Yang C, Qu L, et al. Hide your model: a parameter transmission-free federated recommender system. 2023. ArXiv:2311.14968

39 Long J, Ye G, Chen T, et al. Diffusion-based cloud-edge-device collaborative learning for next POI recommendations. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024. 2026–2036

40 Qu L, Yuan W, Zheng R, et al. Towards personalized privacy: user-governed data contribution for federated recommendation. 2024. ArXiv:2401.17630

41 Qu L, Tang N, Zheng R, et al. Semi-decentralized federated ego graph learning for recommendation. In: Proceedings of the ACM Web Conference, 2023. 339–348

42 Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst, 2020, 32: 4–24

43 Muhammad K, Wang Q, O'Reilly-Morgan D, et al. FedFast: going beyond average for faster training of federated recommender systems. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020. 1234–1242

44 Zhang H, Luo F, Wu J, et al. LightFR: lightweight federated recommendation with privacy-preserving matrix factorization. ACM Trans Inf Syst, 2023, 41: 1–28

45 Wang J, Zhang T, Song J, et al. A survey on learning to Hash. IEEE Trans Pattern Anal Mach Intell, 2017, 40: 769–790

46 Minto L, Haller M, Livshits B, et al. Stronger privacy for federated collaborative filtering with implicit feedback. In: Proceedings of the 15th ACM Conference on Recommender Systems, 2021. 342–350

47 Zhang S, Yuan W, Yin H. Comprehensive privacy analysis on federated recommender system against attribute inference attacks. IEEE Trans Knowl Data Eng, 2024, 36: 987–999

48 Yuan W, Yang C, Nguyen Q V H, et al. Interaction-level membership inference attack against federated recommender systems. In: Proceedings of the ACM Web Conference, 2023. 1053–1062

49 Yuan W, Yin H, Wu F, et al. Federated unlearning for on-device recommendation. In: Proceedings of the 16th ACM International Conference on Web Search and Data Mining, 2023. 393–401

50 Nguyen T T, Nguyen Q V H, Nguyen T T, et al. Manipulating recommender systems: a survey of poisoning attacks and countermeasures. ACM Comput Surv, 2025, 57: 1–39

51 Wang Z, Yu J, Gao M, et al. Poisoning attacks and defenses in recommender systems: a survey. 2024. ArXiv:2406.01022

52 Wang Z, Yu J, Gao M, et al. Unveiling vulnerabilities of contrastive recommender systems to poisoning attacks. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024. 3311–3322

53 Rong D, Ye S, Zhao R, et al. FedRecAttack: model poisoning attack to federated recommendation. In: Proceedings of IEEE 38th International Conference on Data Engineering, 2022. 2643–2655

54 Rong D, He Q, Chen J. Poisoning deep learning based recommender model in federated learning scenarios. 2022. ArXiv:2204.13594

55 Yuan W, Nguyen Q V H, He T, et al. Manipulating federated recommender systems: poisoning with synthetic users and its countermeasures. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023. 1690–1699

56 Yuan W, Yuan S, Yang C, et al. Manipulating visually aware federated recommender systems and its countermeasures. ACM Trans Inf Syst, 2024, 42: 1–26

57 Wu C, Wu F, Qi T, et al. FedAttack: effective and covert poisoning attack on federated recommendation via hard sampling. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022. 4164–4172

58 Chen C, Zhang J, Tung A K, et al. Robust federated recommendation system. 2020. ArXiv:2006.08259

59 Zhou K, Wang H, Zhao W X, et al. S3-Rec: self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020. 1893–1902

60 Wei Y, Wang X, Li Q, et al. Contrastive learning for cold-start recommendation. In: Proceedings of the 29th ACM International Conference on Multimedia, 2021. 5382–5390

61 Qin X, Yuan H, Zhao P, et al. Intent contrastive learning with cross subsequences for sequential recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024. 548–556

62 Zhou Y, Liu J, Wang J H, et al. USST: a two-phase privacy-preserving framework for personalized recommendation with semi-distributed training. Inf Sci, 2022, 606: 688–701

63 Wang Z, Yu J, Gao M, et al. Poisoning attacks against contrastive recommender systems. 2023. ArXiv:2311.18244

64 Luo L, Liu B. Dual-contrastive for federated social recommendation. In: Proceedings of International Joint Conference on Neural Networks, 2022. 1–8

65 Yin H, Wang Q, Zheng K, et al. Overcoming data sparsity in group recommendation. IEEE Trans Knowl Data Eng, 2020, 34: 3447–3460

66 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Artificial Intelligence and Statistics, 2017. 1273–1282

67 He X, Liao L, Zhang H, et al. Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, 2017. 173–182

68 Bai T, Wen J R, Zhang J, et al. A neural collaborative filtering model with interaction-based neighborhood. In: Proceedings of the ACM on Conference on Information and Knowledge Management, 2017. 1979–1982

69 He X, Du X, Wang X, et al. Outer product-based neural collaborative filtering. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018. 2227–2233

70 Wang T, Isola P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: Proceedings of International Conference on Machine Learning, 2020. 9929–9939

71 Hua T, Wang W, Xue Z, et al. On feature decorrelation in self-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 9598–9608

72 Shi Y, Liang J, Zhang W, et al. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. In: Proceedings of the 11th International Conference on Learning Representations, 2022

73 Blanchard P, El Mhamdi E M, Guerraoui R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 30

74 Yin D, Chen Y, Kannan R, et al. Byzantine-robust distributed learning: towards optimal statistical rates. In: Proceedings of International Conference on Machine Learning, 2018. 5650–5659

75 Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980