Revisit Targeted Model Poisoning on Federated Recommendation: Optimize via Multi-objective Transport

Jiajie Su
Zhejiang University
Hangzhou,China
sujiajie@zju.edu.cn

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

Weiming Liu
Zhejiang University
Hangzhou,China
21831010@zju.edu.cn

Zibin Lin
Zhejiang University
Hangzhou,China
zibinlin@zju.edu.cn

Shuheng Shen
Ant Group
Hangzhou,China
shuheng.ssh@antgroup.com

Weiqiang Wang
Ant Group
Hangzhou,China
weiqiang.wwq@antgroup.com

Xiaolin Zheng
Zhejiang University
Hangzhou,China
xlzheng@zju.edu.cn

ABSTRACT

Federated Recommendation (FedRec) is popularly investigated in personalized recommenders for preserving user privacy. However, due to the distributed training paradigm, FedRec is vulnerable to model poisoning attacks. In this paper, we focus on the targeted model poisoning attack against FedRec, which aims at effectively attacking the FedRec via uploading poisoned gradients to raise the exposure ratio of a multi-target item set. Previous attack methods excel with fewer target items but suffer performance decline as the amount of target items increases, which reveals two perennially neglected issues: (i) The simple promotion of prediction scores without considering intrinsic collaborations between users and items is ineffective in multi-target cases. (ii) Target items are heterogeneous, which requires discriminative attacking users and strategies for different targets. To address the issues, we propose a novel Heterogeneous Multi-target Transfer Attack framework named HMTA which consists of two stages, i.e., (1) diverse user agent generation and (2) optimal multi-target transport attack. The former stage leverages collaboration-aware manifold learning to extract latent associations among users and items, and develops a differentiable contrastive sorting to generate user agents from both difficulty and diversity scale. The latter stage conducts poisoning in a fine-grained and distinguishing way, which first completes distribution mapping from target items to generated user agents and then achieves a hybrid multi-target attack. Extensive experiments on benchmark datasets demonstrate the effectiveness of HMTA.

*Chaochao Chen is the corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657764

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Federated Recommendation, Targeted Model Poisoning Attack

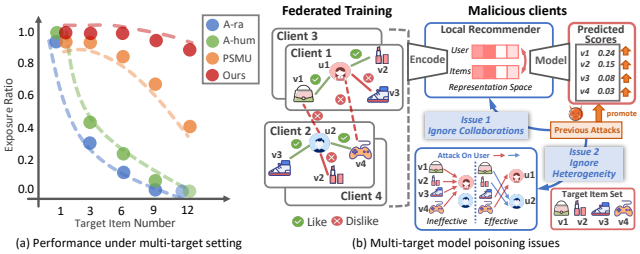
ACM Reference Format:
Jiajie Su, Chaochao Chen, Weiming Liu, Zibin Lin, Shuheng Shen, Weiqiang Wang, and Xiaolin Zheng. 2024. Revisit Targeted Model Poisoning on Federated Recommendation: Optimize via Multi-objective Transport. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657764

1 INTRODUCTION

Personalized recommendations are extensively employed on online platforms for delivering precise content to people [16, 28, 30, 39]. Conventional recommenders centrally store users' personal data for training, arousing severe privacy concerns and risks. Federated Recommendation (FedRec) [1, 23, 43], a privacy-preserving framework that trains a global recommender by collaboratively modeling decentralized data in local clients, has emerged in response to apprehensions regarding privacy and regulatory restrictions [14, 35].

However, due to the distributed nature of federated learning, FedRec lends itself to adversarial attacks in the presence of malicious clients who intend to alter training results. In this paper, we focus on the **Targeted Model Poisoning Attack** (TMPA) against FedRec, where the adversary manipulates a few clients to corrupt the global recommender into a targeted misprediction on specific subtasks, e.g., promoting the overall exposure ratio of *a target item set*. Once malicious parties control a part of clients, they modify local training and upload carefully crafted gradients, to influence the global model biased towards recommending the target item set.

Several approaches have been proposed for conducting TMPA. One line of the attacks assumes the adversary has access to *partial knowledge* of global datasets. PipAttck [53] utilizes the given



(a) Performance under multi-target setting (b) Multi-target model poisoning issues

Figure 1: Motivation of multi-target poisoning in FedRec.

item popularity information to endow target items with features of popular items, while FedRecAttack [37] relies on public interactions to infer user representations. Considering the unavailability of data prior knowledge, another line of the attacks realizes target item promotion in a *data-free* setting. A-ra and A-hum [36] employ random sampling and hard user approximation respectively for malicious user generation. PSMU [50] simulates synthetic users by randomly selecting interactions. These studies overcome the lack of prior information through *a unified paradigm* which i) *imitates user representations* on malicious clients, and ii) *promotes prediction scores* of target items on malicious clients. Unfortunately, their attacking performances sharply diminish with an increasing amount of target items, as shown in Fig. 1(a). From this phenomenon, we rethink TMPA in FedRec with consideration of two issues:

Issue 1: Simply promoting prediction scores ignores collaborations between simulated users and target items. As Fig.1(b) shows, the intrinsic reason prompting FedRec to provide different recommendation lists, lies in the collaborative nature of clients and items. Previous attacks simply alter predicted scores on limited malicious clients, which disregards revising collaborations on the representation level. Local score promotion only affects limited clients or a group of similar users, failing to have a global impact on FedRec. Besides, straightforwardly and brutally boosting scores of all targets also overlooks relations among target items, which triggers potential conflicts in optimization.

Issue 2: Attacking multiple target items with identical optimization objective ignores target heterogeneity. As Fig.1(b) shows, target items may exhibit rich diversity, implying the need to generate distinct users and find the best matching between users and targets for poisoning. For example, item $v1$ and $v2$ are liked by user $u1$ but disliked by $u2$, while $v3$ and $v4$ are exactly the opposite. To promote exposure, poisoning targets on the users who initially dislike them is more effective, e.g., poison $v1/v2$ on $u2$ while $v3/v4$ on $u1$. Previous attacks indiscriminately poison randomly generated users, neglecting target heterogeneity. The heterogeneity requires simulating more targeted users, thereby formulating discriminative attack strategies to achieve a globally optimal promotion.

To address these issues, we propose a Heterogeneous Multi-target Transfer Attack (**HMTA**), which aims at exploiting distinctive traits of targets and leveraging multifaceted user-item collaborations, to tailor various attack strategies for optimal performances. **HMTA** contains two main stages, diverse user agent generation and optimal multi-target transport attack. (1) The *diverse user agent generation* produces targeted user agents with high attack quality on malicious clients, which indirectly enhances the performance of downstream attacks. Unlike existing attacks [36, 50] that craft synthetic users randomly, **HMTA** endeavors to explore informative associations between users and target items in latent feature space to generate representative agents for poisoning. The generation stage has two steps: (i) First, we propose *collaboration-aware manifold learning* to extract topological correlations among users and items regarding their collaborations (**Issue 1**). (ii) Second, we develop a *differentiable contrastive sorting* to select an agent group with potent attack capability from both difficult scale and diversity scale (**Issue 2**). (2) With elaborately generated agents from the first stage, the *optimal multi-target transport attack* then reconstructs TMPA paradigm by conducting discriminative attacking strategies on different target items. This stage is composed of two steps, i.e., (i) we achieve *distribution mapping* between user agents and target items via optimal transport so that the underneath coupling matching is uncovered (**Issue 1**). (ii) We develop a hybrid multi-target attacking based on the optimal transfer coupling matrix, which accomplishes feature-oriented attack to align feature distributions between agents and targets, and reforms preference-oriented attack to promote the global performance (**Issue 2**).

The main contributions as summarized as: (1) We propose a novel and effective TMPA framework, i.e., **HMTA**, for promoting multi-target poisoning performance on FedRec, which includes diverse user agent generation stage and optimal multi-target transport attack stage. (2) The first stage mines collaborations among users and items to produce an agent group with high attack quality, while the second stage leverages optimal transportation to achieve both feature- and preference-oriented attacks. (3) Extensive empirical studies on benchmark datasets demonstrate that **HMTA** significantly improves the state-of-the-art models.

2 RELATED WORK

Federated Recommendation. FedRec collaboratively trains a global recommender while preserving sensitive data locally. FCF [1] is the first federated learning (FL) based recommendation that utilizes collaborative filtering. Later, FedRecs like [5, 9, 22, 23] adapt distributed matrix factorization to the FL setting for privacy protection. The performance of these FedRecs is limited due to issues such as cold start, data sparsity, non-identical and independent distributions. With the advancement of deep learning, meta-learning [6], graph neural network [29, 43, 44], neural collaborative filtering [33], reinforcement learning [19], contrastive learning [46] and self-supervised learning [20] based FedRecs emerge, greatly improving the recommendation accuracy. Recently, an increasing number of FedRecs [9, 24, 32] focus on enhancing privacy preservation.

Attacks on Federated Recommendation. FedRec is inherently vulnerable to potential attacks because of its open and decentralized nature [42, 51]. Poisoning attack is widely adopted by adversaries to alter recommendation results of FedRec, which is categorized into data poisoning and model poisoning. ***Data poisoning attack*** maliciously manipulates training data to mislead local models on clients. To generate more targeted interactions for pollution, data poisoning always requires full knowledge of the global dataset [10, 11, 18, 52], which unfortunately is not available in FedRec. The only related research is FedAttack [45], which leverages globally hardest samples to subvert model training. ***Model poisoning attack*** interferes with local training directly for uploading poisoned

gradients to the global recommender, which is a more effective attack against FedRec. Concerning attack goals, model poisoning has two types, i.e., untargeted and targeted. The untargeted model poisoning [49] attacks FedRec to diminish the general recommendation accuracy. The **targeted model poisoning**, as the topic of this paper, aims at promoting exposure rates of specific target items. Early work on targeted model poisoning heavily depends on *prior knowledge of global datasets*. For instance, PipAttack [53] utilizes information about item popularity to disguise target items as popular items. FedRecAttack [37] relies on partial public user interactions to simulate user feature vectors for poisoning. In light of the unavailability of data prior knowledge in federated protocols, recent studies engaged in conducting potent attacks in the *data-free setting*. A-ra and A-hum [36] first employ random approximation and hard user mining respectively to manipulate malicious users for poisoning. PSMU [50] generates synthetic users on malicious clients with randomly sampled interactions and considers target items' alternative products when executing score promotion. Although these methods archive attacks without prior knowledge, they ignore i) collaborations between users and targets which constitute the inherent factor of prediction scores. ii) Target items have heterogeneous attributes so that indiscriminate attack strategies result in suboptimal performance. Therefore their performances significantly decrease when tackling multi-target poisoning tasks.

3 PRELIMINARIES

3.1 Federated Recommendation Framework

Based on [36, 50, 53], we employ the most popularly used FedRec framework proposed by [1] in this paper.

Base recommender model. Following [50], we apply widely used recommenders, i.e., Neural Collaborative Filtering (NCF) [16] and LightGCN [15]. We adapt both into the federated setting, i.e., Fed-NCF and Fed-GCN, to validate the generalization of our attack.

Local training. Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of N users/clients and M items. Each client u_i holds its own local dataset $\mathcal{D}_i$, which is composed of implicit feedback tuples (u_i, v_j, y_{ij}). If u_i interacts with item v_j, then $y_{ij} = 1$ (positive item), otherwise $y_{ij} = 0$ (negative item). Due to the large amount of unobserved interactions, we randomly select the negative item set for each user with a positive-to-negative ratio of 1:q. The local recommender is trained to predict interaction probabilities $\hat{y}_{ij}$ between u_i and all non-interacted items v_j, recommending top-K items with the highest scores. We employ the cross-entropy loss function for local training:

$$\mathcal{L}^i_{rec} = -\sum\nolimits_{(u_i,v_j,y_{ij})\in\mathcal{D}_i} y_{ij}\log\hat{y}_{ij} + (1-y_{ij})\log(1-\hat{y}_{ij}). \quad (1)$$

Federated protocols. There are two types of parameters in FedRec, i.e., public and private parameters. The public parameters contain item embeddings $\boldsymbol{V}$ and other common parameters $\boldsymbol{\Theta}$ for recommendation model stored in the central server, while private parameters indicate user embeddings $\boldsymbol{U}$ stored locally in each client. To start with, the server and clients initialize all public and private parameters. At every global epoch t, the central server randomly selects a proportion of clients $\mathcal{U}^t$ and distributes a copy of current global parameters $\boldsymbol{V}^t$ and $\boldsymbol{\Theta}^t$ to them. For the selected client $u_i \in \mathcal{U}^t$, incorporating the received public parameters $\boldsymbol{V}^t$ and $\boldsymbol{\Theta}^t$ with private user embedding $\boldsymbol{u}_i$, it trains the recommender with dataset $\mathcal{D}_i$, as specified in Eq.(1). With several steps of local training, the client updates user embedding with the gradient $\nabla\boldsymbol{u}_i^t$ as:

$$\boldsymbol{u}_i^{t+1} = \boldsymbol{u}_i^t - \gamma\nabla\boldsymbol{u}_i^t, \quad (2)$$

where γ denotes the learning rate. Meanwhile, the client derives the gradients of public parameters as $\nabla\boldsymbol{V}_i^t$ and $\nabla\boldsymbol{\Theta}_i^t$, uploading them to the server. The server obtains updated public parameters by aggregating uploaded gradients:

$$\boldsymbol{V}^{t+1} = \boldsymbol{V}^t - \gamma\sum_{u_i\in\mathcal{U}^t}\nabla\boldsymbol{V}_i^t,\ \boldsymbol{\Theta}^{t+1} = \boldsymbol{\Theta}^t - \gamma\sum_{u_i\in\mathcal{U}^t}\nabla\boldsymbol{\Theta}_i^t. \quad (3)$$

3.2 Targeted Model Poisoning Attack

Following [53], an adversary starts conducting targeted model poisoning attacks on *a small proportion* of *malicious clients* at the s-th global training epoch. From this epoch, malicious clients alter the local training process for a special purpose, and upload deceptive gradients $\nabla\widetilde{\boldsymbol{V}}_i^t$ and $\nabla\widetilde{\boldsymbol{\Theta}}_i^t$, to steer the global model towards making biased recommendations on targeted items.

Attack task. Given the targeted item set as $\widetilde{\mathcal{V}}$, the adversary aims to promote the *average exposure ratio* of these items. For a recommender that recommends top-K items to users, we define the exposure ratio at rank K of the target item set as:

$$ER@K = \frac{1}{|\widetilde{\mathcal{V}}|}\sum_{v_j\in\widetilde{\mathcal{V}}}\frac{\left|\{u_i\in\mathcal{U}|v_j\in\mathcal{V}_i^+\wedge v_j\in\mathcal{V}_i^-\}\right|}{\left|u_i\in\mathcal{U}|v_j\in\mathcal{V}_i^-\right|}, \quad (4)$$

where $\mathcal{V}_i^+$ denotes the top-K items in the output recommendation list and $\mathcal{V}_i^-$ is the non-interacted item set for u_i. Inspired by [50], let T denote the total number of global epochs, we reform the attack into an optimization problem, where the adversary seeks to find the ideal poisoned gradients $\nabla\widetilde{\boldsymbol{V}}$ and $\nabla\widetilde{\boldsymbol{\Theta}}$ to maximize ER@K:

$$\underset{\{\nabla\widetilde{\boldsymbol{V}}^t,\nabla\widetilde{\boldsymbol{\Theta}}^t\}_{t=s}^{T-1}}{\operatorname{argmax}} \quad ER@K(\boldsymbol{U}^T,\boldsymbol{V}^T,\boldsymbol{\Theta}^T). \quad (5)$$

Here, $\boldsymbol{U}^T$ indicates the learned user embedding matrix of benign clients, while $\boldsymbol{V}^T$ and $\boldsymbol{\Theta}^T$ are learned public parameters after aggregating gradients from malicious and benign clients for T epochs. Since malicious clients lack access to any prior knowledge, i.e., benign user's embedding vector and training dataset, directly optimizing Eq.(5) is infeasible. Previous work [50, 53] introduces a paradigm about reforming an approximated optimization objective:

$$\underset{\{\nabla\widetilde{\boldsymbol{V}}^t,\nabla\widetilde{\boldsymbol{\Theta}}^t\}}{\operatorname{argmax}} \quad ER@K(\boldsymbol{U}^t,\boldsymbol{V}^t-\gamma\nabla\widetilde{\boldsymbol{V}}^t,\boldsymbol{\Theta}^t-\gamma\nabla\widetilde{\boldsymbol{\Theta}}^t), \quad (6)$$

which provides the alternative solution of greedily optimizing ER@K on malicious clients at each global epoch to achieve the final attack goal of promoting target items on benign clients.

4 METHODOLOGY

4.1 An Overview of HMTA

The aim of **HMTA** is conducting effective multi-target model poisoning attacks on FedRec. As Figure 2, **HMTA** consists of two stages, i.e., **Diverse User Agent Generation** and **Optimal Multi-target Transport Attack**. The generation stage explores latent associations between users and target items thus producing attack user agents with high quality. In this stage, we first develop a *collaboration-aware manifold* learning to construct an agent space. Then we apply a *differentiable contrastive sorting* that selects a user

agent group with two constraints, i.e., difficulty scale and diversity scale. Due to the lack of prior knowledge, the attack stage is built upon the agent generation. In this stage, we first realize *distribution mapping via optimal transport* to obtain the best coupling matching between agents and targets. Then we *design a hybrid multi-target attacking* which achieves feature-oriented attack with latent distribution alignment and preference-oriented attack with score gap elimination. In each epoch, local recommenders on malicious clients train with the recommendation loss and attack loss for several steps, then upload poisoned gradients to influence global training.

4.2 Diverse User Agent Generation

Due to the inaccessibility of prior knowledge, the first stage for targeted model poisoning attacks is generating user agents purposefully on malicious clients. Previous studies randomly construct synthetic users, which overlook intricate collaborations between users with target items and the multiplicity of targets. Thus, we propose this stage to simulate discriminative user agents for downstream attack, which has two parts, i.e. (1) *collaboration-aware manifold learning* to extract intrinsic associations and (2) *differentiable contrastive sorting* to enhance difficulty and diversity of agents.

4.2.1 Collaboration-aware manifold learning. Capturing data correlations in high-dimensional space is difficult, thus we utilize manifold learning here to explore topological relationships between target items and agent users in the shared feature subspace.

User agent space construction. As the attacker has no access to interaction data, we first approximate benign user embeddings. Inspired by [36], we categorize benign users into three broad classes, i.e., *easy*, *median* and *difficult users*. Easy users inherently like these target items, median users never interact with target items, while difficult users dislike target items. Intuitively, reversing the disinterest of difficult users is most beneficial for increasing EK@K. Thus, we pre-train a **parent space** consisting of difficult users, from which we select a **subspace** composed of targeted user agents to maximize attack effectiveness. Given the target item set $\widetilde{\mathcal{V}} = \{v_1, v_2, ..., v_{N_t}\}$, we select N_d subsets from its power set to constitute a target item subset $\widetilde{\mathcal{P}}$, which includes all subsets with one element, i.e., $\{\{v_1\}, \{v_2\}, ..\{v_{N_t}\}\}$, and $N_d - N_t$ randomly sampled subsets. For each set $\widetilde{p_i}$ in $\widetilde{\mathcal{P}}$, we initialize a user agent embedding $\widetilde{\boldsymbol{u}_i} \sim \mathcal{N}(0, \sigma^2)$. To endow user agents with characteristics of difficult users, we feed these user embeddings into the local recommender $\mathcal{R}$ and train them as those who take target items as negative instances:

$$\widetilde{y}_{ij} = \mathcal{R}\left(\widetilde{\boldsymbol{u}_i}, \widetilde{\boldsymbol{v}}_j, \Theta\right), \ \mathcal{L}^i_{pre} = -\sum\nolimits_{v_j \in \widetilde{p_i}} \log(1 - \widetilde{y}_{ij}).$$

For each user agent, we optimize $\widetilde{\boldsymbol{u}_i}$ with gradient descent $\widetilde{\boldsymbol{u}_i} \leftarrow \widetilde{\boldsymbol{u}_i} - \delta \nabla_{\widetilde{\boldsymbol{u}_i}} \mathcal{L}^i_{pre}$, where δ is the learning rate. Repeating this optimization for several rounds, we obtain the pre-trained user agents $\widetilde{\boldsymbol{U}}$.

Collaborative distance extraction. To evaluate the applicability of pre-trained user agents, we develop an *collaborative distance metric*, which (i) projects representations into a topology-preserving subspace through principal manifolds, and (ii) computes attentive distances between agents and targets.

(i) Subspace projection. We combine the distributed item embedding matrix $\boldsymbol{V}$ and pre-trained user agents $\widetilde{\boldsymbol{U}}$ as $\boldsymbol{X} = \left[\boldsymbol{V}, \widetilde{\boldsymbol{U}}\right] \in \mathbb{R}^{(M+N_d)\times D}$. For each x_i in $\boldsymbol{X}$, let $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \cdots, x_{ik}) \in \mathbb{R}^{k\times D}$ be a local neighborhood containing its k-nearest neighbors. As [54], we compute the affine subspace approximation for $\boldsymbol{X}_i$:

$$\arg\min_{\boldsymbol{W}} \sum\nolimits_{j=1}^{k} \|x_{ij} - \overline{x}_i - \boldsymbol{W}\boldsymbol{W}^\top(x_{ij} - \overline{x}_i)\|_F^2, \quad \overline{x}_i = \frac{1}{k}\sum\nolimits_{j=1}^{k} x_j,$$

where $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}$, and $\boldsymbol{W}$ is the projection matrix. The subspace coordinate which represents the local geometry is $\boldsymbol{\Phi}_i = (\theta_{i1}, \theta_{i2}, \cdots, \theta_{ik}) \in \mathbb{R}^{k\times D_c}$, where $\theta_{ij} = \boldsymbol{W}^\top(x_{ij} - \overline{x}_i)$. By minimizing the sum of distances between each sample and its projection, we find the local tangent space for each neighborhood which indicates common latent features of each implicit user/item cluster. Then we reconstruct global coordinates through local tangent space alignment. Let $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_{M+N_d}) \in \mathbb{R}^{(M+N_d)\times D_c}$ denotes the global coordinates of $\boldsymbol{X}$, the global representation of neighborhood $\boldsymbol{X}_i$ is $\boldsymbol{Z}_i = (\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2}, \cdots, \boldsymbol{z}_{ik})$. The global and local representations satisfy the affine transformation $\boldsymbol{z}_{ij} = \boldsymbol{L}_i\theta_{ij} + \boldsymbol{c}_i + \epsilon_{ij}$, where $\boldsymbol{L}_i$ is affine transformation matrix, $\boldsymbol{c}_i$ is center of $\boldsymbol{Z}_i$, and ϵ_{ij} is reconstruction error. The alignment error of the local space is:

$$E_i = \sum\nolimits_{j=1}^{k} \|\boldsymbol{z}_{ij} - (\boldsymbol{L}_i\theta_{ij} + \boldsymbol{c}_i)\|^2 = \|\boldsymbol{Z}_i - (\boldsymbol{L}_i\boldsymbol{\Phi}_i + \boldsymbol{c}_i\boldsymbol{1}^\top)\|^2.$$

Minimizing E_i of all neighborhoods, we generate the topology-preserving latent feature space: $\boldsymbol{Z} = \arg\min_{\boldsymbol{Z}} \sum_{i=1}^{M+N_d} \|E_i\|$.

(ii) Attentive distance computation. From latent space, we retrieve embeddings for target items and user agents as $\boldsymbol{Z}^T$ and $\boldsymbol{Z}^A$. We propose an attentive distance *difficulty scale* that describes the extent to which a user agent possesses difficult user characteristics. First we extract the center of $\boldsymbol{Z}^T$ as $\boldsymbol{z}^T_c = \frac{1}{N_t}\sum_{i=1}^{N_t} \boldsymbol{Z}^T_i$, representing general features of target items. Then for each user agent $\boldsymbol{Z}^A_u$, we apply an attention mechanism to formulate its difficulty scale $\boldsymbol{D}_u$:

$$\boldsymbol{\alpha}(i,u) = \boldsymbol{h}^\top \tanh(\boldsymbol{W}_c\left[\boldsymbol{z}^T_c, \boldsymbol{Z}^A_u\right] + \boldsymbol{W}_i\left[\boldsymbol{Z}^T_i, \boldsymbol{Z}^A_u\right] + \boldsymbol{b}),$$

$$\boldsymbol{D}_u = \sum\nolimits_{i=1}^{N_t} \frac{\exp(\boldsymbol{\alpha}(i,u))}{\sum_{j=1}^{N_t}\exp(\boldsymbol{\alpha}(j,u))} D_{euc}(\boldsymbol{Z}^T_i, \boldsymbol{Z}^A_u),$$

where $\boldsymbol{W}_c$, $\boldsymbol{W}_i$, $\boldsymbol{h}$, and $\boldsymbol{b}$ are trainable weights. $D_{euc}(\cdot,\cdot)$ denotes the Euclidean distance. The attention score $\boldsymbol{\alpha}(i,u)$ is determined jointly by its local collaborations with each target item $\boldsymbol{Z}^T_i$ and its global association with the target center $\boldsymbol{z}^T_c$.

4.2.2 Differentiable contrastive sorting. Now we seek to select an agent group with high attack quality from the agent space. The attack quality is attributed from: (i) *difficulty scale* with regard to all target items which decides the gain on exposure ratio. (ii) *Diversity scale* which facilitates the coverage of target items with heterogeneous properties. We develop a differentiable contrastive sorting which (1) completes a differentiating agent selection process with isotonic optimization and (2) employs a contrastive loss to strengthen the attack quality of selected agents from both scales.

Differentiating sorting process. A traditional sorting procedure outputs two vectors, i.e., sorted values and sorting permutation, but neither of them is differentiable, making sorting incompatible with an end-to-end deep learning network. To make agent generation trainable for optimization, we adapt a fast differentiable sorting operator into agent selection [4, 38]. Formally, we denote a permutation of user agents as $\Lambda = (\Lambda_1, \cdots, \Lambda_{N_d})$ based on their difficulty scale $\boldsymbol{D}$. The set of $N_d!$ permutations of $[N_d]$ is denoted as Σ. The **argsort** of $\boldsymbol{D}$ is defined as the indices sorting $\boldsymbol{D}$, i.e., $\Lambda(\boldsymbol{D}) := (\Lambda_1(\boldsymbol{D}), \cdots, \Lambda_{N_d}(\boldsymbol{D}))$, and the **sort** of $\boldsymbol{D}$ is denoted as

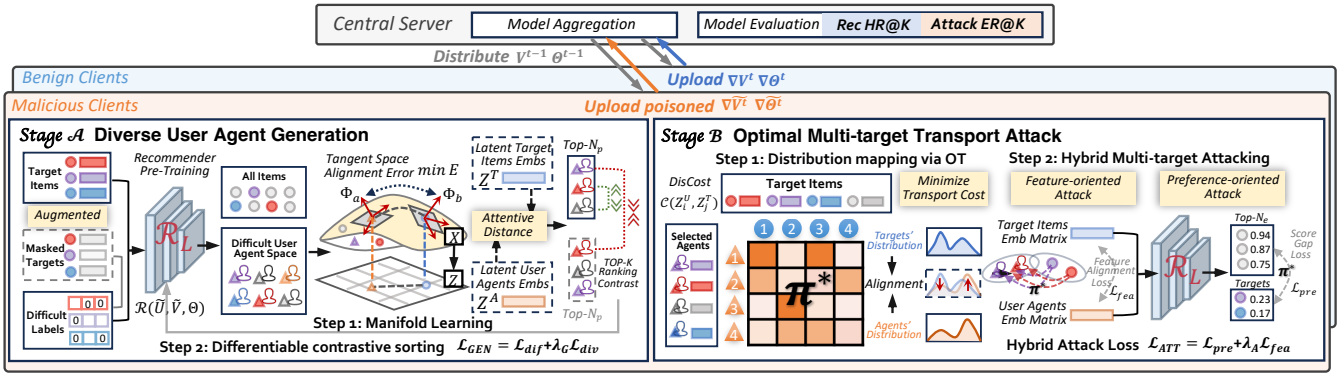


Figure 2: The framework of HMTA.

$\varphi(\boldsymbol{D}) := \boldsymbol{D}_{\Lambda(\boldsymbol{D})}$. Then we formulate the argsort operation as optimization problem over the set of permutations Σ as:

Theorem 4.1 (Discrete optimization of agent argsort). *For difficulty scale $\boldsymbol{D}$ and vector $\boldsymbol{\rho} := (N_d, N_d - 1, \cdots, 1)$, we have*

$$\Lambda(\boldsymbol{D}) = \text{argmax}_{\Lambda \in \Sigma} \langle \boldsymbol{D}_\Lambda, \boldsymbol{\rho} \rangle.$$

To cast it into a continuous optimization problem, following [4], we introduce the permutahedron, which represents the convex hull of permutations of a vector $\boldsymbol{\theta}$: $\mathcal{E}(\boldsymbol{\theta}) := \text{conv}(\{\theta_\Lambda : \Lambda \in \Sigma\})$. Thus when $\boldsymbol{\theta} = \boldsymbol{\rho}, \mathcal{E}(\boldsymbol{\theta}) = \text{conv}(\Sigma)$. With the convex polytope, we derive the discrete optimization into a linear program:

Theorem 4.2 (Linear programming of agent sorting). *For difficulty scale $\boldsymbol{D}$ and vector $\boldsymbol{\rho} := (N_d, N_d - 1, \cdots, 1)$, we have*

$$\varphi(\boldsymbol{D}) = \text{argmax}_{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{D})} \langle \boldsymbol{e}, \boldsymbol{\rho} \rangle.$$

To realize efficient computable approximations to the sorting operator, we introduce the entropic regularization [7] and turn it into a *log* Kullback-Leibler (KL) projection onto the permutahedron:

$$\begin{aligned} P_{KL}(\boldsymbol{\rho}, \boldsymbol{D}) &:= \log \underset{\boldsymbol{e} \in \mathcal{E}(\exp(\boldsymbol{D}))}{\text{argmax}} \langle \boldsymbol{\rho}, \boldsymbol{e} \rangle - \langle \boldsymbol{e}, \log \boldsymbol{e} - 1 \rangle \\ &= \log \underset{\boldsymbol{e} \in \mathcal{E}(\exp(\boldsymbol{D}))}{\text{argmin}} \mathbf{KL}(\boldsymbol{e}, \exp(\boldsymbol{\rho})), \end{aligned} \tag{7}$$

where $\mathbf{KL}(\cdot, \cdot)$ is the KL divergence between two positive measures. As [4], we adopt isotonic optimization to solve Eq.(7), which reduces the projection with chain constraints and benefits fast differentiation. Without loss of generality, for a sorted $\boldsymbol{D}'$:

$$P_{KL}(\boldsymbol{\rho}, \boldsymbol{D}') = \boldsymbol{\rho} - \boldsymbol{\chi}_{KL}(\boldsymbol{\rho}_{\Lambda(\boldsymbol{\rho})}, \boldsymbol{D}')_{\Lambda^{-1}(\boldsymbol{\rho})},$$

$$\textit{where} \quad \boldsymbol{\chi}_{KL}(\boldsymbol{s}, \boldsymbol{D}') := \underset{\chi_1 \geq \cdots \geq \chi_{N^d}}{\text{argmax}} \langle \exp(\boldsymbol{s} - \boldsymbol{\chi}, \mathbf{1}) \rangle + \langle \exp(\boldsymbol{D}'), \boldsymbol{\chi} \rangle.$$

Contrastive ranking loss. After sorting user agents by difficulty scale, we design a contrastive ranking loss to optimize the user agent generation. The ranking loss has two purposes, i.e., (i) guarantee the extensive *difficulty* of the selected user agent group and (ii) enhance the *diversity* of agents to radiate the whole target item set. First, we obtain the top-N_p difficult user agents from the sorted $\boldsymbol{D}'$ as $\widetilde{\mathcal{U}}$, retrieving their user embedding $\widetilde{\boldsymbol{U}}_A$ in the parent space. To control the integral difficulty scale, we send the selected agents and target items into local recommender to acquire the difficulty loss:

$$\mathcal{L}_{dif} = - \sum_{u_i \in \widetilde{\mathcal{U}}} \sum_{v_j \in \widetilde{\mathcal{V}}} \log(1 - \widetilde{y}_{ij}), \ \widetilde{y}_{ij} = \mathcal{R}\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{v}}_j, \boldsymbol{\Theta}\right).$$

Besides, we employ a random strategy to augment the target item embedding $\widetilde{\boldsymbol{V}}$ in the parent space similarly with Dropout [17, 41]:

$$\widetilde{\boldsymbol{V}'} = g(\widetilde{\boldsymbol{V}}) = \widetilde{\boldsymbol{V}} \cdot \boldsymbol{I}, \ \boldsymbol{I} \sim \text{Bernoulli(p)},$$

where $Bernoulli(\cdot)$ is a Bernoulli distribution and $\boldsymbol{I}$ is a matrix of Bernoulli random variables each of which has probability p of being 1. We engage the augmented target item matrix $\widetilde{\boldsymbol{V}'}$ into manifold learning as well. Then we get two selected agent group embedding matrices based on the original and augmented target items respectively, denoted as $\widetilde{\boldsymbol{U}}_A$ and $\widetilde{\boldsymbol{U}}_B$. To preserve the uniformity of agent distribution [34, 40, 47], we contrast two top-N_p groups as:

$$\mathcal{L}_{div} = -\frac{1}{N_p} \sum_{i=1}^{N_p} \log \frac{\exp(sim(\widetilde{\boldsymbol{u}}_i^a, \widetilde{\boldsymbol{u}}_i^b)/\tau)}{\sum_{j=1}^{N_p} \left[\exp(sim(\widetilde{\boldsymbol{u}}_i^a, \widetilde{\boldsymbol{u}}_j^a)/\tau) + \exp(sim(\widetilde{\boldsymbol{u}}_i^a, \widetilde{\boldsymbol{u}}_j^b)/\tau)\right]},$$

where $sim(\cdot, \cdot)$ is the cosine distance that measures the similarity between user pairs, and τ is the temperature parameter. With $\mathcal{L}_{div}$, we enhance the distinctiveness among the top-N_p agents, thereby improving the diversity of the selected agent group. Combining the difficulty loss and diversity loss, we optimize the agent generation stage with a weight hyper-parameter λ_G:

$$\mathcal{L}_{GEN} = \mathcal{L}_{dif} + \lambda_G \mathcal{L}_{div}. \tag{8}$$

4.3 Optimal Multi-target Transport Attack

Previous studies ignore item characteristics and user preferences, thus carrying out indiscriminate attacks. In contrast, we propose an optimal multi-target transport attack to formulate distinct strategies for different targets, leveraging limited agents to achieve maximum attack effectiveness. This stage has two steps, i.e., (1) distribution mapping via optimal transport and (2) hybrid multi-target attacking.

4.3.1 Distribution mapping via optimal transport. The manner and extent of agents' involvement in local training determine the direction of poisoned gradients, which leads to varying degrees of exposure promotion for different target items. Thus, the insight of designing discriminative attack strategies for each target item is to reveal hidden matching between targets and agents. To achieve this task, we employ the optimal transport technique [12, 13, 25] which tackles the general problem of moving one distribution of mass to another as efficiently as possible, by giving out optimal solution of the coupling matching among typical anchors in two distributions. Given the selected agent group $\widetilde{\mathcal{U}}$ and target item set $\widetilde{\mathcal{V}}$, their distributions as Z^U and Z^T, we define the optimal transport between them based on Monge-Kantorovich problem [2, 27].

Problem. Given distance measurement $C(Z_i^U, Z_j^T)$ on samples, the objective is to find the optimal coupling matrix $\pi^* \in \mathbb{R}^{|\widetilde{\mathcal{U}}|\times|\widetilde{\mathcal{V}}|}$ minimizing the total transport cost

$$\pi^* = \arg\min_{\pi} \int_{\widetilde{\mathbf{U}}\times\widetilde{\mathbf{V}}} C(Z_i^U, Z_j^T) d\pi(Z_i^U, Z_j^T).$$

To obtain a differentiable solution, we smooth the objective with an entropic regularization and reform the objective as:

$$\min \left[\sum_{j=1}^{|\widetilde{\mathcal{V}}|} \sum_{i=1}^{|\widetilde{\mathcal{U}}|} \pi_{ij} \|Z_i^U - Z_j^T\|_2^2 + \varepsilon \cdot \sum_{j=1}^{|\widetilde{\mathcal{V}}|} \sum_{i=1}^{|\widetilde{\mathcal{U}}|} \pi_{ij} \cdot (\log(\pi_{ij}) - 1) \right]$$
$$s.t. \quad \|\pi\|_1 = 1, \pi \geq 0, \sum\nolimits_i \pi_{ij} = a, \sum\nolimits_j \pi_{ij} = b.$$

Here, $a = 1/|\widetilde{\mathcal{V}}|$ and $b = 1/|\widetilde{\mathcal{U}}|$, and ε is a balance hyper parameter. We add the constraints on π to facilitate a balanced division of matching between agents and target items.

Optimization. To facilitate the efficient solution to the transport, we apply the Sinkhorn divergence [8, 26]. Specifically, we adopt the Lagrangian multiplier to minimize the objective function as below:

$$\max_{f,g} \min_{\pi} \mathcal{J} = \sum_{j=1}^{|\widetilde{\mathcal{V}}|} \sum_{i=1}^{|\widetilde{\mathcal{U}}|} \pi_{ij} \boldsymbol{c}_{ij} + \varepsilon \cdot \sum_{j=1}^{|\widetilde{\mathcal{V}}|} \sum_{i=1}^{|\widetilde{\mathcal{U}}|} \pi_{ij} \cdot (\log(\pi_{ij}) - 1)$$
$$\sum_{j=1}^{|\widetilde{\mathcal{V}}|} f_j [(\sum_{i=1}^{|\widetilde{\mathcal{U}}|} \pi_{ij}) - \boldsymbol{a}] - \sum_{i=1}^{|\widetilde{\mathcal{U}}|} g_i [(\sum_{j=1}^{|\widetilde{\mathcal{V}}|} \pi_{ij}) - \boldsymbol{b}],$$

where $c_{ij} = \|Z_i^U - Z_j^T\|_2^2$. Taking the differentiation on π_{ij}, we have

$$\frac{\partial \mathcal{J}}{\partial \pi_{ij}} = 0 \Rightarrow c_{ij} + \varepsilon \log(\pi_{ij}) - f_j - g_i = 0.$$

To optimize multipliers, we adopt dual ascent method, where we first fix g_i and update f_j as *step1*, then fix f_j and update g_i as *step2*:

$$step1: f_j^{(t+1)} = \varepsilon \left\{ \log(\boldsymbol{a}) - log \left[\sum_{i=1}^{|\widetilde{\mathcal{U}}|} \exp\left(\frac{g_i^{(t)} - c_{ij}}{\varepsilon} \right) \right] \right\}$$
$$step2: g_i^{(t+1)} = \varepsilon \left\{ \log(\boldsymbol{b}) - log \left[\sum_{j=1}^{|\widetilde{\mathcal{V}}|} \exp\left(\frac{f_j^{(t)} - c_{ij}}{\varepsilon} \right) \right] \right\}$$

We iteratively update the multipliers, and after several iterations, we obtain the final convergence result on π^*.

4.3.2 Hybrid multi-target attacking. To adapt heterogeneous target items' features, we develop a hybrid multi-target attack loss to resolve contradictions between local optimization directions, thus achieving global maximization of exposure ratio improvements. Concretely, hybrid multi-target attack has two aspects, i.e., (1) feature-oriented attack and (2) preference-oriented attack.

Feature-oriented attack. First, we reduce the feature bias between the agent group and target item set, so that target items are endowed with characteristics that *difficult users* favor. With the optimal coupling matrix π^*, we find the best direction of transferring agents, i.e., difficult users, from user space to item feature space. Thus we propose the feature alignment loss to minimize the distribution discrepancy between agents and target items:

$$\mathcal{L}_{fea} = \sum_{v_j \in \widetilde{\mathcal{V}}} \sum_{u_i \in \widetilde{\mathcal{U}}} \pi_{ij}^* \|Z_i^U - Z_j^T\|_2^2.$$

Empirically, the closer the distance between two representations in the latent space, the more shared features they own.

Preference-oriented attack. Second, we directly reverse the preferences of user agents on target items, which implicitly modify the recommender parameters and item representations to transform target items from *unpopular* to *popular*. For malicious clients, we reduce the prediction score gap on user agents between target items and the items that they initially like:

$$\mathcal{L}_{pre} = \sum_{v_j \in \widetilde{\mathcal{V}}} \sum_{u_i \in \widetilde{\mathcal{U}}} \sum_{v_k \in \mathcal{V}_p} \pi_{ij}^* \sigma \left(\mathcal{R}\left(\widetilde{\boldsymbol{u}_i}, \boldsymbol{v}_k, \Theta\right) - \mathcal{R}\left(\widetilde{\boldsymbol{u}_i}, \widetilde{\boldsymbol{v}_j}, \Theta\right) \right),$$

where $\mathcal{R}(\boldsymbol{u}, \boldsymbol{v}, \Theta)$ denotes the prediction scores of the local recommender, and $\mathcal{V}_p$ indicates the top-N_e items except target items which gain highest scores in recommendation.

Hybrid attacking. We combine the feature-oriented attack loss with preference-oriented attack loss to achieve hybrid attacking:

$$\mathcal{L}_{ATT} = \mathcal{L}_{pre} + \lambda_A \mathcal{L}_{fea}. \tag{9}$$

where λ_A is a hyper-parameter to balance different losses.

5 EXPERIMENTS AND ANALYSIS

We conduct experiments to explore the following research questions. **RQ1**: How does **HMTA** perform compared with existing targeted model poisoning attacks on FedRec? **RQ2**: Does **HMTA** harm the recommendation performance of FedRec significantly? **RQ3**: How does the malicious client proportion affect the attack performance? **RQ4**: Can our attack bypass mainstream defensive methods in FedRec? **RQ5**: How does **HMTA** benefit from each key stage? **RQ6**: What is the impact of hyperparameters to **HMTA**?

5.1 Experimental Setup

Datasets. We evaluate our model on three widely used real-world datasets in FedRec, i.e., **MovieLens (ML)**, **Amazon (AZ)**, and **IJCAI**. Following [36, 50], we binarize user-item ratings into implicit data, and filter the data to ensure all users have at least 5 interactions. The interaction ratio of the training and test set is 4:1. We show detailed statistics of datasets in Table 2.

Evaluation protocols. As [50], we first train the FedRec for several epochs, and then the attack is launched at a certain epoch. When malicious clients start uploading poisoned gradients, we evaluate the effectiveness of attacks after each global epoch. Since the goal of a targeted poisoning attack is to promote the exposure ratio of target items while retaining the general performance of FedRec, we evaluate with two metrics: (1) **Exposure Ratio at rank K (ER@K)** to measure the effectiveness of attack. (2) **Hit Ratio at top K (HR@K)** to quantify side effects that attacks bring to recommendation performance. We vary the amount of target items to validate the effectiveness of **HMTA** in the multi-target setting.

Comparison methods. Since we focus on the attack setting without prior knowledge, we compare **HMTA** with data poisoning and model poisoning baselines which are applicable in this setting. (1) **No Attack** shows the original performance of FedRec without attack. (2) **Random Attack (RanAtt)** [21] injects malicious users with random interactions. (3) **Explicit Boosting (ExpBoo)** is a variant of [53] removing popularity boosting that requires prior information. (4) **A-ra** [36] samples malicious users from Gaussian distribution. (5) **A-hum** is an improved version of A-ra utilizing hard user mining. (6) **PSMU** [50] generates synthetic users with random interactions and considers targets' alternative products.

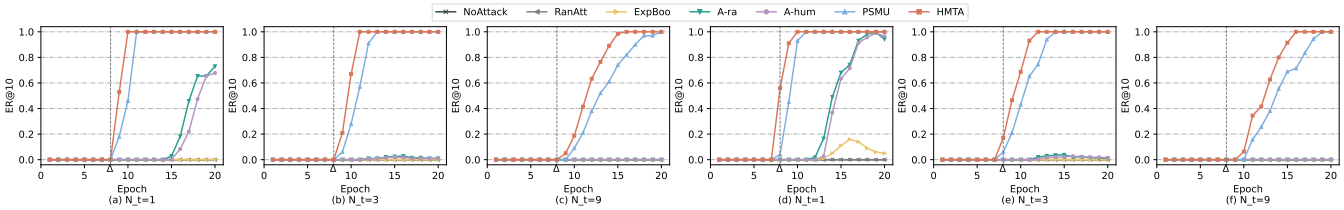


Figure 3: Attack effectiveness under different attack models on Fed-NCF ((a)-(c)) and Fed-GCN ((d)-(f)) for MovieLens.

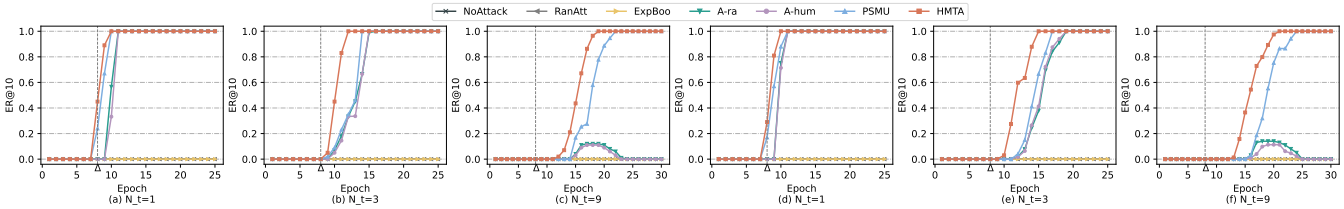


Figure 4: Attack effectiveness under different attack models on Fed-NCF ((a)-(c)) and Fed-GCN ((d)-(f)) for Amazon.

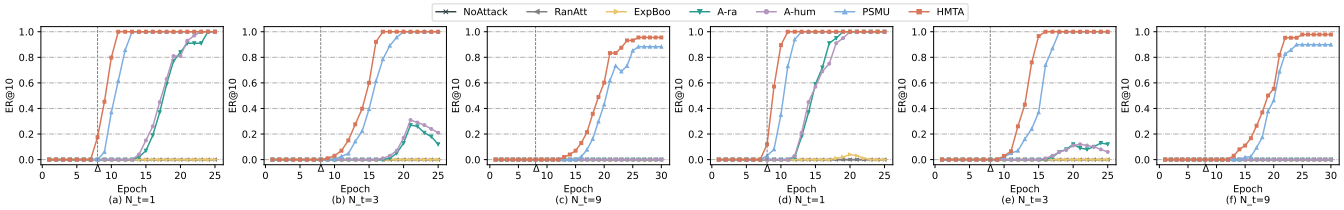


Figure 5: Attack effectiveness under different attack models on Fed-NCF ((a)-(c)) and Fed-GCN ((d)-(f)) for IJCAI.

Table 1: The attack performance of HMTA when handling various defense methods.

Defense methods	Fed-NCF						Fed-GCN					
ER@10 (AZ, IJCAI)	$\mu = 0.1\%$		$\mu = 1.0\%$		$\mu = 10\%$		$\mu = 0.1\%$		$\mu = 1.0\%$		$\mu = 10\%$	
	$N_t = 3$	$N_t = 9$	$N_t = 3$	$N_t = 9$	$N_t = 3$	$N_t = 9$	$N_t = 3$	$N_t = 9$	$N_t = 3$	$N_t = 9$	$N_t = 3$	$N_t = 9$
No Defense	(1.0,1.0)	(1.0,0.96)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,0.98)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)
Item-level Krum	(1.0,1.0)	(1.0,0.96)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,0.98)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)
Median	(1.0,1.0)	(1.0,0.96)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,0.98)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)	(1.0,1.0)
Trimmed Mean	(0.86,0.82)	(0.77,0.73)	(1.0,1.0)	(0.96,0.89)	(1.0,1.0)	(1.0,1.0)	(0.87,0.85)	(0.85,0.82)	(1.0,1.0)	(0.98,0.93)	(1.0,1.0)	(1.0,1.0)
l_2 clipping	(0.86,0.82)	(0.75,0.73)	(1.0,1.0)	(0.91,0.85)	(1.0,1.0)	(1.0,1.0)	(0.87,0.83)	(0.82,0.79)	(1.0,1.0)	(0.98,0.94)	(1.0,1.0)	(1.0,1.0)
HiCS	(0.75,0.72)	(0.68,0.62)	(0.91,0.84)	(0.82,0.76)	(1.0,1.0)	(1.0,1.0)	(0.78,0.73)	(0.72,0.67)	(0.91,0.88)	(0.82,0.79)	(1.0,1.0)	(1.0,1.0)

Implemented details. We set target item number $N_t = 1, 3, 9$, and malicious client proportion $\mu = 0.1\%$ in main experiments. All attacks are launched at 8^{th} epoch. We set the dimension of the embedding $D = 32$ and $D = 16$ for Fed-NCF and Fed-GCN, and the latent dimension in manifold learning $D_c = 8$. The learning rate δ for user agent pre-training is 0.01, and the learning rate of Adam optimizer $\gamma = 0.001$ and $\gamma = 0.01$ for Fed-NCF and Fed-GCN. We test various lengths of ranked list, for space save, we only report ER@10 and HR@20 here. Moreover, we set the number of nearest neighbors $k = 5$ in collaborative distance extraction and positive-to-negative ratio $q = 4$ in FedRec local training. Specifically, we study effects of hyper-parameters N_p, N_e, λ_G, and λ_A in RQ6.

5.2 Attack Performance (RQ1-RQ3)

Model comparison (RQ1). We evaluate the attack performance of **HMTA** and all baselines on Fed-NCF and Fed-GCN for three datasets. From the results in Fig.3-5, we find that: (1) Simple baselines that cannot execute precise attacks, i.e., RanAtt and ExpBoo, have no impact on exposure promotion when $\mu = 0.1\%$. (2) A-ra and A-hum show comparable performances and they both achieve 1.0 ER@10 on Fed-GCN for three datasets when $N_t = 1$. But their attack effectiveness significantly deteriorates when the amount of target item N_t increases. (3) **HMTA** outperforms all baselines on both FedRecs for three datasets. Both **HMTA** and the best SOTA PSMU realize 1.0 ER@10 when $N_t = 1$ and $N_t = 3$, but **HMTA** always takes fewer epochs, e.g., **4** and **3** epochs less than PSMU on Fed-NCF for ML and IJCAI, which **improves attack efficiency and greatly reduces probability of being detected**. When N_t increases to 9, **HMTA** exhibits **higher attack capability and stability encountering with the multi-target cases**, where it achieves **0.955** ER@10 on Fed-NCF for IJCAI (*PSMU 0.883*), and **0.978** on Fed-GCN (*PSMU 0.899*). The improvement ratios in this case are **8.15%** and **8.78%**. These results demonstrate that **HMTA** shows superiority on various FedRecs especially in multi-target attacks, which is attributed to the accurate generation of user agents and more discriminative poisoning tactics towards target items.

Datasets	#User	#Item	Interactions	Sparsity
MovieLens	6,040	3,706	1,000,208	95.53%
Amazon	16,566	11,797	169,781	99.91%
IJCAI	423,423	874,328	36,222,123	99.99%

Table 2: Statistics on Datasets.

Influence to recommendation performance (RQ2). To ensure the stealthiness of our attack, we further evaluate side effects that poisoning brings to recommendation performance of FedRec. Due

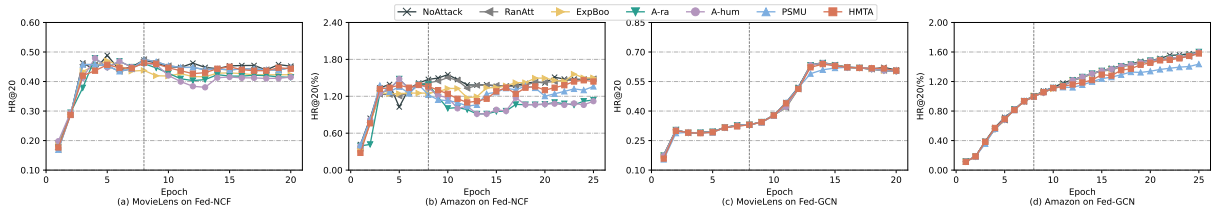
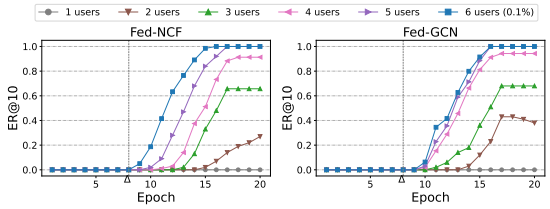
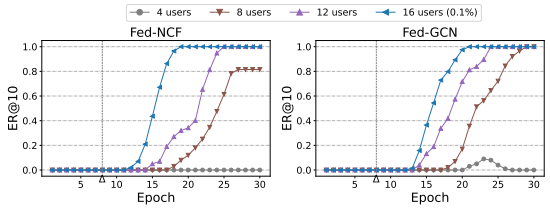


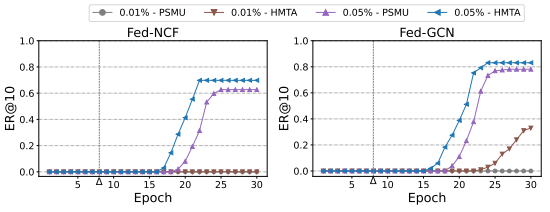
Figure 6: The recommendation performance of FedRec under different attack models.



(a) **HMTA** with different number of malicious users on Movielens.



(b) **HMTA** with different number of malicious users on Amazon.



(c) **HMTA** and **PSMU** with different number of malicious users on IJCAI.

Figure 7: Impact of the malicious client proportion.

to limited space, we present the HR@20 on both FedRecs for ML and AZ under each attack in Fig.6, from which we observe that all methods especially **HMTA** produce insignificant effects on recommendation accuracy. This phenomenon indicates the stealthy of all attacks when the malicious client proportion $\mu = 0.1\%$.

Impact of malicious client proportion (RQ3). We investigate the attack stability by varying malicious client proportions. In RQ1, we proved the effectiveness of **HMTA** when the proportion $\mu = 0.1\%$. In Fig.7, we test **HMTA** on both FedRecs with fewer malicious users when $N_t = 9$, which implies a significant amplification of attacking difficulty. From the results, we find (1) **HMTA** possesses stable attack capability even with extremely limited malicious clients. On ML, as the μ decreases, **HMTA** needs more training rounds to achieve the highest exposure ratio, but it achieves 0.657 and 0.680 on Fed-NCF and Fed-GCN respectively with only 3 users. On AZ, with only 8 malicious users and 9 target items, **HMTA** reaches 0.815 and 1.0 HR@10 on two FedRecs. (2) **HMTA** obtains higher exposure rates than the best baseline PSMU under all circumstances with lower malicious client proportion. As shown in 7(c), with $N_t = 0.5\%$ **HMTA** achieves 0.698 and 0.831 ER@10 on Fed-NCF and Fed-GCN for dataset IJCAI, while PSMU only reaches 0.627 and 0.781. This demonstrates that since **HMTA** considers latent collaborations and realizes discriminative attacking towards each target on user agents with high quality, it exhibits stronger robustness and stability than PSMU when tackling multi-target tasks.

Table 3: Ablation results on diverse user agent generation.

Variants	Fed-NCF			Fed-GCN		
(ML,IJCAI)	$N_t = 1$	$N_t = 3$	$N_t = 9$	$N_t = 1$	$N_t = 3$	$N_t = 9$
HMTA-dif	(4,6)	(6,12)	(13,0.899)	(4,6)	(7,11)	(11,0.934)
HMTA-div	(3,4)	(4,10)	(11,0.937)	(3,4)	(6,10)	(11,0.962)
HMTA-RA	(4,6)	(7,13)	(14,0.913)	(4,6)	(8,11)	(11,0.925)
HMTA-HU	(4,5)	(6,12)	(11,0.937)	(3,4)	(7,10)	(10,0.955)
HMTA	**(3,4)**	**(4,10)**	**(9,0.955)**	**(3,4)**	**(5,9)**	**(9,0.978)**

5.3 Handling Defenses (RQ4)

We study whether our attack can bypass the state-of-art defense methods at the server side in FedRec. Following [50, 53], we utilize 5 mainstream defense baselines: (1) **Item-level Krum** adapts the original Krum [3] into FedRec by selecting embedding gradient closest to the mean of other clients' gradients as the aggregated value. (2) **Median** [48] chooses the median of uploaded parameters as aggregated ones. (3) **Trimmed Mean** [48] calculates the mean of parameter gradients by excluding the largest and smallest values. (4) l_2 **clipping** [31] normalizes all gradients before aggregation. (5) **HiCS** [50] develops a hierarchical gradient clipping with sparsified updating. We report attack performance of **HMTA** under defenses in Table 1, from which we find: (1) When $\mu = 10\%$, ER@10 keeps 1.0 against all defenses on both FedRecs for both datasets with $N_t = 3$ and $N_t = 9$, which indicates **HMTA** successfully evades all defenses in this case. (2) When μ decreases to 1.0%, **HMTA** is still able to bypass most defenses with $N_t = 3$, achieving 1.0 ER@10 on both FedRecs. (3) The attack difficulty is at its maximum when $\mu = 0.1\%$ with $N_t = 9$ among these cases. There are more items requiring increased exposure and fewer malicious clients to realize poisoning, so the gradient differences between malicious and benign clients become too pronounced to defend against. Nevertheless, **HMTA** still promotes ER@10 to (0.68,0.62) on Fed-NCF, while (0.72,0.67) on Fed-NCF for AZ and IJCAI respectively. These results suggest that current defenses in FedRec cannot successfully resist HMTA, which further underscores the imperative to develop sophisticated defenses against this type of attack.

5.4 Ablation Study (RQ5)

Study of diverse user agent generation stage. In order to verify the effectiveness of the generation, we conduct ablation studies on the following variants: (a) **HMTA**-dif removes the difficult scale, i.e., $\mathcal{L}_{dif}$, from contrastive ranking in agent generation. (b) **HMTA**-div removes the diversity scale, i.e., $\mathcal{L}_{div}$, from contrastive ranking.

Table 4: Ablation results on multi-target transport attack.

Variants	Fed-NCF			Fed-GCN		
(ML,IJCAI)	$N_t = 1$	$N_t = 3$	$N_t = 9$	$N_t = 1$	$N_t = 3$	$N_t = 9$
HMTA-pre	(6,10)	(7,18)	(0.762,0.757)	(6,10)	(8,16)	(0.895,0.828)
HMTA-fea	(4,6)	(6,12)	(11,0.913)	(3,5)	(6,11)	(11,925)
HMTA-OP	(3,4)	(4,10)	(10,0.937)	(3,4)	(5,10)	(10,0.955)
HMTA-OF	(3,4)	(4,11)	(10,0.921)	(3,4)	(6,11)	(11,0.934)
HMTA	**(3,4)**	**(4,10)**	**(9,0.955)**	**(3,4)**	**(5,10)**	**(9,0.978)**

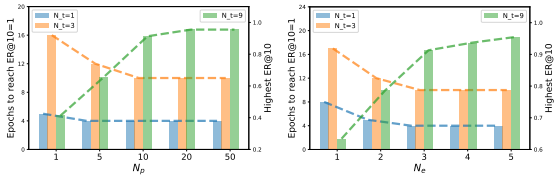


Figure 8: Effects of hyper-parameters N_p and N_e.

(c) **HMTA**-RA removes the whole generation stage, and produces malicious users with random selected interactions as PSMU [50]. (d) **HMTA**-HU also removes the whole stage, and simulates users with hard user mining as A-hum [36]. We report the comparison results in Table.3, note that if the variant can reach 1.0 ER@10 in some cases, we use the epoch number used for convergence as the metric, otherwise we report ER@10 score. From the table, we conclude: (1) Both **HMTA**-dif and **HMTA**-div perform worse than **HMTA**, which indicates the rationality of incorporating the difficulty scale and diversity scale in user agent generation. **HMTA**-div outperforms **HMTA**-dif, showing that difficulty scale has a greater impact than diversity scale in improving user agents' attack quality. Specifically, **HMTA**-div exhibits a competitive attacking capability with **HMTA** when $N_t = 1$, but performs notably poorer than **HMTA** when $N_t = 9$, which further validates that considering the diversity of user agents is essential for the multi-target attack. (2) **HMTA** outperforms both **HMTA**-RA and **HMTA**-HU, which proves that generating targeted and diverse user agents helps promote the attack effectiveness. With differentiable contrastive sorting from difficulty and diversity, our generation stage is superior to either random manner or hard user mining.

Study of optimal multi-target transport attack stage. To study how each component in the attack stage contributes to the final attack performances, we compare **HMTA** with following variants: (a) **HMTA**-pre removes the preference-oriented attack in this stage. (b) **HMTA**-fea removes the feature-oriented attack. (c) **HMTA**-OP removes the optimal coupling matrix in the feature-oriented loss $\mathcal{L}_{fea}$. (d) **HMTA**-OF removes the optimal coupling matrix in the preference-oriented loss $\mathcal{L}_{pre}$. From the results shown in Table 4, we conclude: (1) **HMTA**-pre and **HMTA**-fea perform worse than **HMTA**, suggesting that either preference-oriented or feature-oriented attack contributes to exposure promotion of target items. **HMTA**-fea achieves higher ER@10 than **HMTA**-pre on both FedRecs for dataset IJCAI when $N_t = 9$ and takes fewer epochs to reach 1.0 ER@10 in other cases, which indicates the preference-oriented loss takes on a dominant role in attacking. (2) **HMTA** outperforms both **HMTA**-OP and **HMTA**-OF, showing that mapping distribution between agents and targets with optimal transport facilitates the hybrid multi-target attack resolving local contradictions and optimizing the attack for maximum global impact.

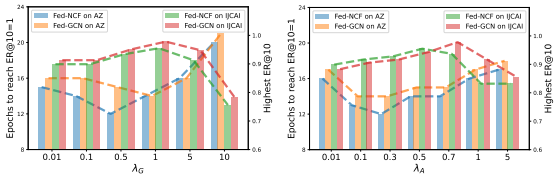


Figure 9: Effects of hyper-parameters λ_G and λ_A.

5.5 Parameter Sensitivity (RQ6)

We now study the effects of hyper-parameters on model performance, including $N_p, N_e, \lambda_G, \lambda_A$. We first study the effect of selected user agent amount N_p in λ_{fea} and selected popular item amount N_e in λ_{pre}, by varying them in $\{1, 5, 10, 20, 50\}$ and $\{1, 2, 3, 4, 5\}$ respectively. The results on Fed-NCF for IJCAI are presented in Fig.8, which shows (1) The effects of N_p and N_e are not evident when $N_t = 1$, performance reaches its peak with very small N_p and N_e. (2) When tackling multi-target cases, the exposure ratio gradually improves when N_p and N_e increase and keep a fairly stable level after reaching a certain threshold. This demonstrates that more user agents and popular items used in $\mathcal{L}_{pre}$ enhance attack intensity because they cover more target items. Considering the computational cost for generating user agents, as well as the risk of abnormal gradient production caused by excessive use of popular items, we choose $N_p = 5, 10, 10$ and $N_e = 2, 3, 3$. Then we study the effects of λ_G and λ_A by varying them in $\{0.01, 0.1, 0.5, 1, 5, 10\}$ and $\{0.01, 0.1, 0.3, 0.5, 0.7, 1, 5\}$ when $N_t = 9$, and report the results in Fig.8. The bell-shaped curves show that when λ_G and λ_A approach 0, the diversity scale in the generation stage and the feature-oriented loss in the attack stage fail to yield positive outcomes on exposure promotion. When λ_G becomes too large, the agent generation excessively focuses on self-group diversity and ignores collaborations with target items, i.e., difficulty scale, thus reduce the performance. When λ_A is set large, the feature-oriented loss takes on a dominant position which suppresses preference-oriented $\mathcal{L}_{pre}$ and thus produces negative effects. Empirically, we choose $\lambda_G = 0.5, 1, 1, 1$ and $\lambda_A = 0.3, 0.3, 0.5, 0.7$ on Fed-NCF and Fed-GCN for AZ and IJCAI.

6 CONCLUSION

We propose a model poisoning framework named **HMTA** against FedRec from a new perspective, i.e., to overcome the multi-target exposure promotion problem. We develop two stages, i.e., diverse user agent generation and optimal multi-target transport attack. The first stage extracts intrinsic collaborations between users and targets, and selects agents with high quality from the difficulty and diversity scales. The second stage proposes a hybrid multi-target attacking strategy, which leverages optimal transport to match agents and targets, and conducts feature- and preference-oriented attacks. Extensive experiments demonstrate the effectiveness of **HMTA** but also expose the inadequate resilience of current defensive methods in FedRec. We plan to investigate more potent and targeted defense strategies against the model poisoning attack as our future work, to enhance the reliability of federated recommendation.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China (No. 2022YFB4501504).

REFERENCES

[1] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888* (2019).

[2] Sigurd Angenent, Steven Haker, and Allen Tannenbaum. 2003. Minimizing flows for the Monge–Kantorovich problem. *SIAM journal on mathematical analysis* 35, 1 (2003), 61–97.

[3] Peva Blanchard, ElMahdiEl Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: byzantine tolerant gradient descent. *Neural Information Processing Systems* (Dec 2017).

[4] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. 2020. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*. PMLR, 950–959.

[5] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2021. Secure Federated Matrix Factorization. *IEEE Intelligent Systems* (Sep 2021), 11–20. https://doi.org/10.1109/mis.2020.3014880

[6] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated Meta-Learning with Fast Convergence and Efficient Communication. *arXiv: Learning,arXiv: Learning* (Feb 2018).

[7] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).

[8] Marco Cuturi. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2292–2300.

[9] Yongjie Du, Deyun Zhou, Yu Xie, Jiao Shi, and Maoguo Gong. 2021. Federated matrix factorization for privacy-preserving recommender systems. *Applied Soft Computing* 111 (2021), 107700.

[10] Jiaxin Fan, Qi Yan, Mohan Li, Guanqun Qu, and Yang Xiao. 2022. A Survey on Data Poisoning Attacks and Defenses. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*. IEEE, 48–55.

[11] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. 2018. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference*. 381–392.

[12] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. 2021. Pot: Python optimal transport. *The Journal of Machine Learning Research* 22, 1 (2021), 3571–3578.

[13] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. 2016. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems* 29 (2016).

[14] Elizabeth Liz Harding, Jarno J Vanto, Reece Clark, L Hannah Ji, and Sara C Ainsworth. 2019. Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy* 2, 3 (2019), 234–253.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 639–648. https://doi.org/10.1145/3397271.3401063

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[17] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

[18] Hai Huang, Jiaming Mu, Neil Zhenqiang Gong, Qi Li, Bin Liu, and Mingwei Xu. 2021. Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644* (2021).

[19] Wei Huang, Jia Liu, Tianrui Li, Tianqiang Huang, Shenggong Ji, and Jihong Wan. 2021. Feddsr: Daily schedule recommendation in a federated deep reinforcement learning framework. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[20] Mubashir Imran, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Alexander Zhou, and Kai Zheng. 2023. ReFRS: Resource-Efficient Federated Recommender System for Dynamic and Diversified User Preferences. *ACM Trans. Inf. Syst.* 41, 3, Article 65 (feb 2023), 30 pages. https://doi.org/10.1145/3560486

[21] Saakshi Kapoor. 2017. A REVIEW OF ATTACKS AND ITS DETECTION ATTRIBUTES ON COLLABORATIVE RECOMMENDER SYSTEMS. *International Journal of Advanced Research in Computer Science* 8, 7 (Sep 2017), 1188–1193. https://doi.org/10.26483/ijarcs.v8i7.4550

[22] Feng Liang, Weike Pan, and Zhong Ming. 2021. Fedrec++: Lossless federated recommendation with explicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4224–4231.

[23] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. 2021. FedRec: Federated Recommendation With Explicit Feedback. *IEEE Intelligent Systems* 36, 5 (2021), 21–30. https://doi.org/10.1109/MIS.2020.3017205

[24] Zhaohao Lin, Weike Pan, and Zhong Ming. 2021. FR-FMSS: Federated recommendation via fake marks and secret sharing. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 668–673.

[25] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Yanchao Tan, Fan Wang, Xiaolin Zheng, and Yew Soon Ong. 2024. Learning Accurate and Bidirectional Transformation via Dynamic Embedding Transportation for Cross-Domain Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8815–8823.

[26] Weiming Liu, Xiaolin Zheng, Chaochao Chen, Jiajie Su, Xinting Liao, Mengling Hu, and Yanchao Tan. 2023. Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation. In *Proceedings of the ACM Web Conference 2023*. 383–394.

[27] Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. 2022. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of International ACM SIGIR conference on research and development in information retrieval*. 312–321.

[28] Weiming Liu, Xiaolin Zheng, Jiajie Su, Longfei Zheng, Chaochao Chen, and Mengling Hu. 2023. Contrastive proxy kernel stein path alignment for cross-domain cold-start recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[29] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S Yu. 2022. Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–24.

[30] Zheqi Lv, Wenqiao Zhang, Zhengyu Chen, Shengyu Zhang, and Kun Kuang. 2024. Intelligent Model Update Strategy for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2024*.

[31] ElMahdiEl Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. *International Conference on Machine Learning* (Jul 2018).

[32] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. 2021. Stronger privacy for federated collaborative filtering with implicit feedback. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 342–350.

[33] Vasileios Perifanis and Pavlos S Efraimidis. 2022. Federated neural collaborative filtering. *Knowledge-Based Systems* 242 (2022), 108441.

[34] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.

[35] General Data Protection Regulation. 2018. General data protection regulation (GDPR). *Intersoft Consulting, Accessed in October* 24, 1 (2018).

[36] Dazhong Rong, Qinming He, and Jianhai Chen. 2022. Poisoning Deep Learning based Recommender Model in Federated Learning Scenarios. In *International Joint Conference on Artificial Intelligence*.

[37] Dazhong Rong, Shuai Ye, Ruoyan Zhao, Hon Ning Yuen, Jianhai Chen, and Qinming He. 2022. FedRecAttack: model poisoning attack to federated recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2643–2655.

[38] Michael Eli Sander, Joan Puigcerver, Josip Djolonga, Gabriel Peyré, and Mathieu Blondel. 2023. Fast, differentiable and sparse top-k: a convex analysis perspective. In *International Conference on Machine Learning*. PMLR, 29919–29936.

[39] Jiajie Su, Chaochao Chen, Zibin Lin, Xi Li, Weiming Liu, and Xiaolin Zheng. 2023. Personalized Behavior-Aware Transformer for Multi-Behavior Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6321–6331.

[40] Jiajie Su, Chaochao Chen, Weiming Liu, Fei Wu, Xiaolin Zheng, and Haoming Lyu. 2023. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM Web Conference 2023*. 165–176.

[41] Fangye Wang, Yingxu Wang, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2023. CL4CTR: A Contrastive Learning Framework for CTR Prediction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 805–813.

[42] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. 2021. Fast-adapting and privacy-preserving federated recommender system. *The VLDB Journal* (2021), 1–20.

[43] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925* (2021).

[44] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications* 13, 1 (2022), 3091.

[45] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. 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* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 4164–4172. https://doi.org/10.1145/3534678.3539119

[46] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. Fedcl: Federated contrastive learning for privacy-preserving recommendation. *arXiv preprint arXiv:2204.09850* (2022).

[47] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[48] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 5650–5659. https://proceedings.mlr.press/v80/yin18a.html

[49] Yang Yu, Qi Liu, Likang Wu, Runlong Yu, Sanshi Lei Yu, and Zaixi Zhang. 2023. Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4854–4863.

[50] Wei Yuan, Quoc Viet Hung Nguyen, Tieke He, Liang Chen, and Hongzhi Yin. 2023. Manipulating Federated Recommender Systems: Poisoning with Synthetic Users and Its Countermeasures. *arXiv preprint arXiv:2304.03054* (2023).

[51] Wei Yuan, Chaoqun Yang, Quoc Viet Hung Nguyen, Lizhen Cui, Tieke He, and Hongzhi Yin. 2023. Interaction-level membership inference attack against federated recommender systems. *arXiv preprint arXiv:2301.10964* (2023).

[52] Hengtong Zhang, Changxin Tian, Yaliang Li, Lu Su, Nan Yang, Wayne Xin Zhao, and Jing Gao. 2021. Data poisoning attack against recommender system using incomplete and perturbed data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2154–2164.

[53] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Quoc Viet Hung Nguyen, and Lizhen Cui. 2022. PipAttack: Poisoning Federated Recommender Systems for Manipulating Item Promotion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1415–1423. https://doi.org/10.1145/3488560.3498386

[54] Zhenyue Zhang and Hongyuan Zha. 2004. Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment. *SIAM J. Sci. Comput.* 26, 1 (jan 2004), 313–338. https://doi.org/10.1137/S1064827502419154