Dual Personalization on Federated Recommendation

Chunxu Zhang[1,2] , **Guodong Long**[3] , **Tianyi Zhou**[4] , **Peng Yan**[3] , **Zijian Zhang**[1,2] , **Chengqi Zhang**[3] and **Bo Yang**[1,2*]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China
[2]College of Computer Science and Technology, Jilin University, China
[3]Australian Artificial Intelligence Institute, FEIT, University of Technology Sydney
[4]Computer Science and UMIACS, University of Maryland
{cxzhang19, zhangzj2114}@mails.jlu.edu.cn, {guodong.long, Chengqi.Zhang}@uts.edu.au,
zhou@umiacs.umd.edu, yanpeng9008@hotmail.com, ybo@jlu.edu.cn

Abstract

Federated recommendation is a new Internet service architecture that aims to provide privacy-preserving recommendation services in federated settings. Existing solutions are used to combine distributed recommendation algorithms and privacy-preserving mechanisms. Thus it inherently takes the form of heavyweight models at the server and hinders the deployment of on-device intelligent models to end-users. This paper proposes a novel Personalized Federated Recommendation (PFedRec) framework to learn many user-specific lightweight models to be deployed on smart devices rather than a heavyweight model on a server. Moreover, we propose a new dual personalization mechanism to effectively learn fine-grained personalization on both users and items. The overall learning process is formulated into a unified federated optimization framework. Specifically, unlike previous methods that share exactly the same item embeddings across users in a federated system, dual personalization allows mild finetuning of item embeddings for each user to generate user-specific views for item representations which can be integrated into existing federated recommendation methods to gain improvements immediately. Experiments on multiple benchmark datasets have demonstrated the effectiveness of PFedRec and the dual personalization mechanism. Moreover, we provide visualizations and in-depth analysis of the personalization techniques in item embedding, which shed novel insights on the design of recommender systems in federated settings. The code is available.

1 Introduction

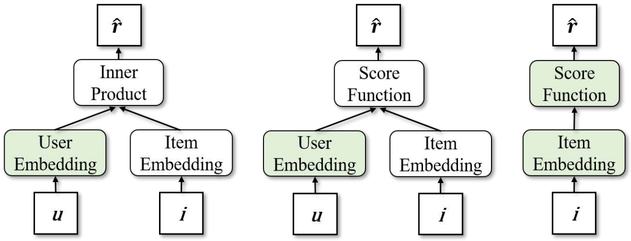
Federated recommendation is a new service architecture for Internet applications, and it aims to provide personalized recommendation service while preserving user privacy in the federated settings. Existing federated recommendation systems [Muhammad *et al.*, 2020; Yi *et al.*, 2021; Perifanis and Efraimidis, 2022; Wu *et al.*, 2022b] are usually to be an adaptation of distributed recommendation algorithms by embodying the data locality in federated setting and adding privacy-preserving algorithms with guaranteed protection. However, these implementations of federated recommendations still inherit the traditional service architecture, which is to deploy large-scale models at servers. Thus it is impractical and inconsistent with the newly raised on-device service architecture, which is to deploy a lightweight model on the device to provide service independently without frequently communicating with the server. Given the challenge of implementing data locality on devices in federated settings, the personalization mechanism needs to be reconsidered to better capture fine-grained personalization for end-users.

Personalization is the core component of implementing federated recommendation systems. Inherited from conventional recommendation algorithms, existing federated recommendation frameworks are usually composed of three modules: user embedding to preserve the user's profile, item embedding to maintain proximity relationships among items, and the score function to predict the user's preference or rating for a given item. They usually preserve user-specific personalization in the user embedding module while sharing consensus on item embeddings and score functions.

This paper proposes a new **dual personalization** mechanism designed to capture fine-grained two-fold personal preferences for users in the federated recommendation system. Inspired by human beings' decision logic, we believe all modules in the recommendation framework should be used to preserve part of personalization rather than use user embedding only. For example, the score function is to mimic the user's personal decision logic that is natural to be diverse across clients. Furthermore, given an item set, different people may have different views on measuring their proximity relationships. Therefore, personalized item embedding could be essential to capture people's personal preferences further.

To implement the aforementioned ideas in federated settings, we propose a new federated recommendation framework to implement fine-grained personalization on multiple modules which are illustrated in Figure 1 (c). First, we use a personalized score function to capture user's preferences, and it could be implemented with a multi-layer neural networks. Second, we remove the user embedding from the federated

[*]Corresponding author.



(a) FedMF (b) FedNCF (c) PFedRec

Figure 1: Different frameworks for the personalized federated recommendation. The green block represents a **personalized module**, which indicates the part of model is to preserve user preference. Our proposed model will preserve dual personalization on two modules.

recommendation framework because the current neural-based personalized score function has enough representation capability to preserve the information of user embeddings. Third, we implement light finetuning to learn personalized item embeddings in federated settings. This proposed decentralized intelligence architecture is a natural simulation of human beings' decision-making that each person has a relatively independent mind to make decisions.

The learning procedure is also carefully tailored in a federated setting. A personalized score function will be learned using the user's own data on the device, and then it won't be sent to the server for global aggregation that usually generates a general view for all devices. Moreover, the personalized item embedding will be implemented through light finetuning in a federated learning framework, thus it can leverage both the general view from server and the personalized view from user's own data.

In summary, we propose a novel federated recommendation framework that integrates both the personalized score function and personalized item embedding via light finetuning from the shared item embedding. **Our key contributions** are summarized as follows.

- We propose a novel federated recommendation framework which is more naturally consistent with layer-wise neural architecture and can better fit federated learning.
- We design a novel dual personalization mechanism to capture user preferences using a personalized score function and fine-grained personalization on item embeddings. It can be integrated with other baselines to improve their performances.
- We formulate the proposed federated recommendation learning problem into a unified federated optimization framework with a bi-level objective.
- Our method can significantly outperform existing federated recommendation baselines.

2 Related Work

2.1 Personalized Federated Learning

Federated learning is a new machine learning paradigm that a server orchestrates a large number of clients to train a model without accessing their data [Kairouz *et al.*, 2021; Li *et al.*, 2020; Long *et al.*, 2021; Long *et al.*, 2020; Tan *et al.*, 2022c; Tan *et al.*, 2022b; Chen *et al.*, 2023]. The vanilla federated learning method, FedAvg [McMahan *et al.*, 2017], is to learn a robust model at the server while embodying data locality for each device with non-IID data. Personalized federated learning is to learn a personalized model for each device to tackle the non-IID challenge. Per-FedAvg [Fallah *et al.*, 2020] exploits Model-Agnostic Meta-Learning (MAML) [Finn *et al.*, 2017] to find a shared initial model for all clients and each client can learn a personalized model with its own data. Ditto [Li *et al.*, 2021] proposes a bi-level optimization framework for PFL which introduces a regularization term by constraining the distance between the local and the global model. [Shamsian *et al.*, 2021] propose to replace the global model with a global hyper-network orchestrate clients' local training. FedRecon [Singhal *et al.*, 2021] is a meta-learning-based method that preserves a local model for each client and trains a global model collaboratively with FedAvg. There are also some attempts about graph-based [Tan *et al.*, 2022a; Chen *et al.*, 2022] and cluster-based [Ma *et al.*, 2022; Long *et al.*, 2023] methods. In this paper, we focus on developing personalization in the federated recommendation scenario where device data distributions are heavily non-IID, and it has not been well explored.

2.2 Federated Recommendation Systems

Federated recommendation has attracted much attention recently due to the rising privacy concern. Some recent works focus on only using the interaction matrix which is the most fundamental recommendation scenario. FCF [Ammad-Ud-Din *et al.*, 2019] is the first FL-based collaborative filtering method, which employs the stochastic gradient approach to update the local model, and FedAvg is adopted to update the global model. Improving user privacy protection, FedMF [Chai *et al.*, 2020] adapts distributed matrix factorization to the FL setting and introduces the homomorphic encryption technique on gradients before uploading to the server. MetaMF [Lin *et al.*, 2020b] is a distributed matrix factorization framework where a meta-network is adopted to generate the rating prediction model and private item embedding. [Wu *et al.*, 2022b] presents FedPerGNN where each user maintains a GNN model to incorporate high-order user-item information. However, the server in both MetaMF and FedPerGNN preserves all the recommendation model parameters which can be used to infer the user's interaction information, resulting in the risk of user privacy leakage. FedNCF [Perifanis and Efraimidis, 2022] adapts Neural Collaborative Filtering (NCF) [He *et al.*, 2017] to the federated setting which introduces neural network to learn user-item interaction function to enhance model learning ability.

Besides, there are federated recommendation methods using rich information that consider multiple data sources in modeling. FedFast [Muhammad *et al.*, 2020] extends FedAvg [McMahan *et al.*, 2017] with an active aggregation method to facilitate the convergence. Efficient-FedRec [Yi *et al.*, 2021] decomposes the model into a news model on the server and a user model on the client, and reduces the computation and communication cost for users. Both works rely on more data sources, such as user features or news attributes rather than an interaction matrix. [Lin *et al.*, 2020a; Du *et al.*, 2021; Minto *et al.*, 2021; Lin *et al.*, 2021] are endeavors that focus on enhancing privacy of FedRec. There are also attempts for other applications in FedRec, such as federated attack [Wu *et al.*, 2022c; Zhang *et al.*, 2022], social recommendation [Liu *et al.*, 2022b], Click-Through Rate (CTR) prediction [Wu *et al.*, 2022a] and fair recommendation [Liu *et al.*, 2022a]. Existing federated recommendation methods usually combine the distributed recommendation algorithms and privacy-preserving techniques directly. They inherits the heavyweight models at the server and pay little attention on lightweight models deployed at client. In this paper, we present a novel federated recommendation framework aiming to learn many user-specific lightweight models rather than a heavyweight model on the server.

3 Problem Formulation

Federated learning is to learn a global model parameterized by θ to serve all clients whose data are private. The optimal solution should minimize the below accumulated loss of all clients,

$$\min_{\theta} \sum_{i=1}^{N} \alpha_i L_i(\theta) \tag{1}$$

where $L_i(\theta)$ is the supervised loss on the i-th client with dataset D_i, and all clients share the global parameter θ. The α_i is a weight for the loss of the i-th client. For example, the conventional FL algorithm, FedAvg [McMahan *et al.*, 2017], defines α_i as the fraction of the size of the client's training data, *i.e.,* $\alpha_i := |D_i| / \sum_{j=1}^{N} |D_j|$. Once the global model is trained, it can be used for prediction tasks on all clients.

Personalized federated learning simultaneously leverages common knowledge among clients and learns a personalized model for each client, with learning objective as,

$$\min_{\theta, \{\theta_i\}_{i=1}^{N}} \sum_{i=1}^{N} \alpha_i L_i(\theta, \theta_i) \tag{2}$$

where each client has a unique personalized parameter θ_i, and θ is the global parameter as mentioned in Eq. (1). For example, [Fallah *et al.*, 2020] leverage θ as initialization of θ_i, *i.e.,* $\theta_i := \theta - \nabla l_i(\theta)$, where $l_i(\theta)$ is the loss of a vanilla model on the i-th client. The $L_i(\theta, \theta_i)$ is then formulated as

$$L_i(\theta, \theta_i) := l_i(\theta - \nabla l_i(\theta)) \tag{3}$$

Recommendation with neural networks. This work focuses on the fundamental scenario where recommendation only relies on the user-item interaction matrix without extra user/item attributes. The recommendation framework can be divided into three components: a user embedding module $\mathcal{E}$, an item embedding module E and a score function S. We denote these modules' parameters with $\theta := (\theta^u, \theta^m, \theta^s)$ and formulate the learning objective as,

$$\min_{\theta} L(\theta; r, \hat{r}) := \min_{\theta} L(\theta; r, S(\mathcal{E}(e^u), E(e^m))) \tag{4}$$

where e^u and e^m are one-hot encodings representing users and items. r is a user's rating to the given item and $\hat{r}$ is a prediction from the score function $S(\mathcal{E}(e^u), E(e^m))$. L is the loss evaluation metric, which could be a ***point-wise loss*** as used in [Wang *et al.*, 2016; He *et al.*, 2017], or a ***pair-wise loss*** as in [Rendle *et al.*, 2012; Wang *et al.*, 2019]. It is worth noting that conventional Matrix Factorization (MF) methods could be viewed as a special case of the framework in Eq. (4), *i.e.,* the conventional MF is a model where the score function S is simplified as the inner product operator without learnable parameters, and the embedding of user/item is obtained by the decomposition of the user-item interaction matrix.

4 Methodology

In this section, we propose a novel **P**ersonalized **Fed**erated **Rec**ommendation (PFedRec) framework, which aims to simultaneously learn many user-specific recommendation models deployed on end devices.

4.1 Objective Function

Federated learning objective. We regard each user as a client under FL settings. The on-device recommendation task is then depicted as a PFL problem. Particularly, the item embedding module E_i is assigned to be a global component which learns common item information and the score function S_i is maintained locally to learn personalized decision logic. To further capture the difference between users and achieve a preference-preserving item embedding, we devise a bi-level optimization objective,

$$\begin{aligned} \min_{\theta^m, \{\theta_i\}_{i=1}^{N}} \quad & \sum_{i=1}^{N} \alpha_i L_i(\theta_i; r, \hat{r}) \\ s.t. \quad & \theta_i := (\theta^m - \nabla_{\theta^m} L_i, \theta_i^s) \end{aligned} \tag{5}$$

where $\theta_i := (\theta_i^m, \theta_i^s)$ is the personalized parameter for E_i and S_i, and L_i will be evaluated on the i-th client local data D_i. Under this framework, PFedRec first tunes E into a personalized item embedding module E_i, and then learns a lightweight local score function S_i to make personalized predictions. Different from the conventional recommendation algorithms, the user embedding module $\mathcal{E}$ is depreciated since the personalization procedure on a client will automatically capture the client's preference. There is no use to learn extra embeddings to describe clients.

Loss for recommendation. Equipped with the item embedding module and score function, we formulate the prediction of j-th item by i-th user's recommendation model as,

$$\hat{r}_{ij} = S_i(E_i(e^j)) \tag{6}$$

Particularly, we discuss the typical recommendation task with implicit feedback, that is, $r_{ij} = 1$ if i-th user interacted with j-th item; otherwise $r_{ij} = 0$. With the binary-value nature of implicit feedback, we define the loss function of i-th user as the *binary cross-entropy loss*,

$$L_i(\theta_i; r, \hat{r}) = -\sum_{(i,j)\in D_i} \log \hat{r}_{ij} - \sum_{(i,j')\in D_i^-} \log(1 - \hat{r}_{ij'}) \tag{7}$$

where D_i^- is the negative instances set of user i. Notably, other loss functions can also be used, and here we choose the binary cross-entropy loss to simplify the description. Particularly, to construct D_i^- efficiently, we first count all the uninteracted items set as,

$$\mathcal{I}_i^- = \mathcal{I} \backslash \mathcal{I}_i \tag{8}$$

where $\mathcal{I}$ denotes the full item list and $\mathcal{I}_i$ is the interacted item set of i-th user. Then, we uniformly sample negative instances from $\mathcal{I}_i^-$ by setting the sampling ratio according to the number of observed interactions and obtain D_i^-.

4.2 Dual Personalization

We present a dual personalization mechanism to enable the proposed framework can preserve fine-grained personalization for both user and item.

Using partial-based federated model aggregation to learn personalized user score function on each device. Our proposed model is composed of a neural-based score function parameterized by θ^s and an item embedding module parameterized by θ^m. The coordinator/server of federated system will iteratively aggregate model parameters or gradients collected from each participant/device. Due to the concern of personalization and privacy, we implement a partial model aggregation strategy by keeping the score function as a private module on devices while sharing the item embedding to the server. Therefore, the server only aggregates the gradients or parameters θ^m from the item embedding module. The user's personalized score function module θ^s won't be sent to the server and thus won't be aggregated. Generally, the simple and swift multi-layer neural network is capable of tackling most scenarios, which is convenient for client deployment.

Finetuning the item embedding module to generate personalized representations for items on each device. According to Eq. (5), the learning objective of θ^m could be viewed as searching for a "good initialization" that could be fast adaptive to the learning task on different devices. It shares similar ideas with meta-learning-based methods [Fallah *et al.*, 2020] which have a local loss in Eq. (3). However, our proposed method takes a different optimization strategy we call *post-tuning*. Specifically, rather than directly tuning a global model on clients' local data, it first learns the local score function with the global item embedding, and then replaces the global item embedding with personalized item embedding obtained by finetuning θ^m.

4.3 Algorithm

Optimization. To solve the optimization problem as described in Sec. 4.1 - objective function, we conduct an alternative optimization algorithm to train the model. As illustrated in Algorithm 1, when the client receives the item embedding from server, it first replaces its item embedding with the global one, and then updates the score function while keeping the item embedding module fixed. Then the client updates the item embedding based on the updated personalized score function. Finally, the updated item embedding would be uploaded to the server for global aggregation.

Workflow. The overall algorithm workflow could be summarized into several steps as follows. The server is responsible for updating shared parameters and organizing all clients to complete collaborative training. At the beginning of federated optimization, the server initializes the model parameters, which would be used as initial parameters for all client models. In each round, the server selects a random set of clients and distributes the global item embedding θ^m to them. When local training is over, the server collects the updated item embedding from each client to perform global aggregation. We build on the simplified version of FedAvg, a direct average of locally uploaded item embeddings. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Optimization Workflow of PFedRec

```
ServerExecute:
Initialize item embedding $\theta^m$ and score function $\theta^s$
for $t = 1, 2, ...$ do
    $S_t \leftarrow$ (select a client set of size $n$ randomly from all $N$ clients)
    for client $i \in S_t$ in parallel do
        $\theta_i^m \leftarrow$ ClientUpdate($i, \theta^m$)
    $\theta^m \leftarrow \frac{1}{n} \sum_{i=1}^{n} \theta_i^m$

ClientUpdate($i, \theta^m$):
Initialize $\theta_i^m$ with $\theta^m$
Initialize $\theta_i^s$ with the latest update
Count all uninteracted items set $\mathcal{I}_i^-$ with Eq. (8)
Sample negative instances set $D_i^-$ from $\mathcal{I}_i^-$
$\mathcal{B} \leftarrow$ (split $D_i \cup D_i^-$ into batches of size $B$)
for $e$ from $1$ to $E$ do
    for batch $b \in \mathcal{B}$ do
        Compute $L_i(\theta_i; r, \hat{r})$ with Eq. (7)
        $\theta_i^s \leftarrow \theta_i^s - \eta \nabla_{\theta_i^s} L_i$
        Compute $L_i(\theta_i; r, \hat{r})$ with Eq. (7)
        $\theta_i^m \leftarrow \theta_i^m - \eta' \nabla_{\theta_i^m} L_i$
Return $\theta_i^m$ to server
```

Efficient on-device update. Focusing on the fundamental recommendation scenario, *i.e.,* with only user-item interaction matrix, the item embedding module E_i dominates the parameter volume in the recommendation model due to large item set size, which brings challenges to end devices with limited computing resources. Generally, the items set that each user interacts with is much smaller than the complete item collection. Based on this observation, we propose that each device only needs to maintain the interacted positive items and sampled negative samples instead of the complete item embedding module, resulting in an efficient on-device update. For clarity, we continue to use θ^m in the Algorithm formulation. In practice, each device only needs to maintain a subset of the complete item embeddings.

5 Discussions

5.1 Privacy on Federated Recommendation

Privacy-preserving is an essential motivation to advance existing cloud-centric recommendation to client-centric recommendation service architecture. In general, the federated learning's decentralized framework can embody data locality and information minimization rules (GDPR) that

Dataset	Interactions	Users	Items	Sparsity
MovieLens-100K	100,000	943	1,682	93.70%
MovieLens-1M	1,000,209	6,040	3,706	95.53%
Lastfm-2K	185,650	1,600	12,454	99.07%
Amazon-Video	63,836	8,072	11,830	99.93%

Table 1: Dataset statistics.

could greatly mitigate the risk of privacy leakage [Kairouz *et al.*, 2019]. To provide service with privacy guarantee, the FL framework should be integrated with other privacy-preserving methods, such as Differential Privacy and secure communication. Our proposed framework derives the same decentralized framework from vanilla FL to preserve data locality. For example, to tackle the privacy leakage risk caused by sending item embedding to the server, we could simply apply differential privacy to inject noise into the embeddings so that the server cannot simply infer the updated items by watching the changes of embeddings. More analysis and experimental verification can be found in Sec. 6.6.

5.2 A General Framework for Federated Recommendation

The proposed framework in Figure 1 (c) could be a general form of federated recommendation because our framework could be easily transformed into an equivalent form of other frameworks. For example, if we assign the score function as a one-layer linear neural network, PFedRec is equal to FedMF in Figure 1 (a). Moreover, if we change the personalized score function from full personalization to partial layer personalization, our method could be equivalent to FedNCF in Figure 1 (b) which has a shared score function across clients. Furthermore, our proposed framework's architecture could be naturally aligned with the classic neural network architecture, thus it has a bigger potential to achieve better learning efficiency and is more flexible to extend.

6 Experiments

6.1 Experimental Setup

We evaluate the proposed PFedRec on four real-world datasets: MovieLens-100K, MovieLens-1M [Harper and Konstan, 2015], Lastfm-2K [Cantador *et al.*, 2011] and Amazon-Video [Ni *et al.*, 2019]. They are all widely used datasets in assessing recommendation models. Specifically, two MovieLens datasets were collected through the MovieLens website, containing movie ratings and each user has at least 20 ratings. Lastfm-2K is a music recommendation dataset, and each user maintains a list of her favorite artists and corresponding tags. Amazon-Video was collected from the Amazon site, containing product reviews and metadata information. We excluded users with less than 5 interactions in Lastfm-2K and Amazon-Video. The characteristics of datasets are shown in Table 1. For dataset split, We follow the prevalent leave-one-out evaluation [He *et al.*, 2017]. We evaluate the model performance (%) with Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) metrics.

6.2 Baselines and Implementation Details

Baselines. Our method is compared with baselines in both centralized and federated settings. Focusing on the performance improvement of the infrastructure of recommendation models that all others derive from, we select the general and fundamental baselines that conduct recommendations only based on the interaction matrix.

- **Matrix Factorization (MF)** [Koren *et al.*, 2009]: This method is a typical recommendation algorithm. Particularly, it decomposes the rating matrix into two embeddings located in the same latent space to characterize users and items, respectively.
- **Neural Collaborative Filtering (NCF)** [He *et al.*, 2017]: This method models user-item interaction function with an MLP, and is one of the most representative neural recommendation models. Specifically, we apply the interaction function with a three-layer MLP for comparison, which is adopted in the original paper.
- **FedMF** [Chai *et al.*, 2020]: It is a federated version of MF which is a typical FedRec method. It updates user embedding locally and aggregates item gradients globally.
- **FedNCF** [Perifanis and Efraimidis, 2022]: It is a federated version of NCF. Specifically, each user updates user embedding locally and uploads item embedding and score function to the server for global update.
- **Federated Reconstruction (FedRecon)** [Singhal *et al.*, 2021]: It is a state-of-the-art PFL framework, and we test it under the matrix factorization scenario. Between every two rounds, this method does not inherit user embedding from the previous round but trains it from scratch.
- **Meta Matrix Factorization (MetaMF)** [Lin *et al.*, 2020b]: It is a distributed matrix factorization framework where a meta-network is adopted to generate the rating prediction module and private item embedding.
- **Federated Graph Neural Network (FedPerGNN) [Wu *et al.*, 2022b]**: It deploys a GNN in each client and the user can incorporate high-order user-item information by a graph expansion protocol.

Implementation details. We sample 4 negative instances for each positive instance following [He *et al.*, 2017]. For a fair comparison, we set the user (item) embedding size as 32 and the batch size is fixed as 256 for all methods, and set other model details of baselines according to their original papers. The total number of communication rounds is set to 100, and this value enables all methods to be trained to converge through experiments. One exception is FedPerGNN, we follow the experimental setting in the original paper with the official code, whose rounds number is set to 3. For our method, we assign the score function with a one-layer MLP for simplification, which can be regarded as an enhanced FedMF with our dual personalization mechanism. We implement the methods based on the Pytorch framework[1] and run all the experiments for 5 repetitions and report the average results.

[1] Code: https://github.com/Zhangcx19/IJCAI-23-PFedRec

	Method	MovieLens-100K		MovieLens-1M		Lastfm-2K		Amazon-Video	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
CenRec	**NCF**	64.14 ± 0.98	37.91 ± 0.37	64.17 ± 0.99	37.85 ± 0.68	82.44 ± 0.42	67.43 ± 0.89	60.16 ± 0.43	38.97 ± 0.14
	MF	64.43 ± 1.02	38.95 ± 0.56	68.45 ± 0.34	41.37 ± 0.18	82.71 ± 0.54	71.04 ± 0.62	46.69 ± 0.65	29.83 ± 0.45
FedRec	**FedMF**	65.15 ± 1.16	39.38 ± 1.08	67.72 ± 0.14	40.90 ± 0.14	81.64 ± 0.48	69.36 ± 0.42	59.67 ± 0.19	38.55 ± 0.21
	FedNCF	60.62 ± 0.59	33.25 ± 1.35	60.54 ± 0.46	34.17 ± 0.40	81.55 ± 0.38	61.03 ± 0.63	57.77 ± 0.07	36.86 ± 0.06
	FedRecon	64.45 ± 0.81	37.78 ± 0.38	63.28 ± 0.15	36.59 ± 0.33	82.06 ± 0.38	67.58 ± 0.35	59.80 ± 0.14	38.87 ± 0.13
	MetaMF	66.38 ± 0.24	40.59 ± 0.31	45.61 ± 0.18	25.24 ± 0.35	80.88 ± 0.45	64.24 ± 0.45	57.51 ± 0.53	37.25 ± 0.28
	FedPerGNN	10.50 ± 0.12	4.92 ± 0.21	9.69 ± 0.23	4.37 ± 0.31	10.19 ± 0.41	4.83 ± 0.25	10.72 ± 0.33	4.90 ± 0.32
	PFedRec (Ours)	**71.62 ± 0.83**	**43.44 ± 0.89**	**73.26 ± 0.20**	**44.36 ± 0.16**	**82.38 ± 0.92**	**73.19 ± 0.38**	**60.08 ± 0.08**	**39.12 ± 0.09**

Table 2: Performance of HR@10 and NDCG@10 on four datasets. **CenRec** and **FedRec** represent centralized and federated methods, respectively. The results are the mean and standard deviation of five repeated trials.

Method	MovieLens-100K		MovieLens-1M		Lastfm-2K		Amazon-Video	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
FedMF	65.15 ± 1.16	39.38 ± 1.08	67.72 ± 0.14	40.90 ± 0.14	81.64 ± 0.48	69.36 ± 0.42	59.67 ± 0.19	38.55 ± 0.21
w/ DualPer	71.62 ± 0.83	43.44 ± 0.89	73.26 ± 0.20	44.36 ± 0.16	82.38 ± 0.92	73.19 ± 0.38	60.08 ± 0.08	39.12 ± 0.09
Improvement	↑ **9.93%**	↑ **10.31%**	↑ **8.18%**	↑ **8.46%**	↑ 0.91%	↑ **5.52%**	↑ 0.69%	↑ 1.48%
FedNCF	60.62 ± 0.59	33.25 ± 1.35	60.54 ± 0.46	34.17 ± 0.40	81.55 ± 0.38	61.03 ± 0.63	57.77 ± 0.07	36.86 ± 0.06
w/ DualPer	68.82 ± 1.35	39.33 ± 0.85	68.17 ± 0.55	39.56 ± 0.29	82.31 ± 0.56	71.64 ± 0.43	59.57 ± 0.57	38.73 ± 0.62
Improvement	↑ **13.53%**	↑ **18.29%**	↑ **12.60%**	↑ **15.77%**	↑ 0.93%	↑ **17.38%**	↑ 3.12%	↑ **5.07%**
FedRecon	64.45 ± 0.81	37.78 ± 0.38	63.28 ± 0.15	36.59 ± 0.33	82.06 ± 0.38	67.58 ± 0.35	59.80 ± 0.14	38.87 ± 0.13
w/ DualPer	70.20 ± 0.90	41.83 ± 0.71	68.89 ± 0.26	40.04 ± 0.16	83.51 ± 0.23	74.83 ± 0.44	60.23 ± 0.16	39.20 ± 0.12
Improvement	↑ **8.92%**	↑ **10.72%**	↑ **8.87%**	↑ **9.43%**	↑ 1.77%	↑ **10.73%**	↑ 0.72%	↑ 0.85%

Table 3: Performance improvement for integrating our dual personalization mechanism (**DualPer**) to three federated baseline algorithms. The results are the mean and standard deviation of five repeated trials, and the significant improvements (over 5%) are highlighted.

6.3 Comparison Analysis

We conduct experiments on four datasets for performance comparison and the resuls are shown in Table 2.

Results & discussion. From the results, we have several observations: (1) **PFedRec obtains better performance than centralized methods in some cases.** In the centralized scenario, only user embedding is regarded as the personalized component to learn user characteristics, and other components are totally shared among users. In comparison, our dual personalization mechanism considers two forms of personalization, which can further exploit user preferences. (2) **PFedRec realizes outstanding advances on the two MovieLens datasets.** In these two datasets, each user has more interaction samples which can be used to train device recommendation models, hence promoting user personalization learning and fitting our method better. (3) **PFedRec consistently achieves the best performance against all federated methods.** In FedRec, the common item embeddings help transfer the shared information among users, which facilitates collaborative training of individual user models. However, different users present rather distinct preferences for items and existing federated methods deploy the global item embeddings indiscriminately for all clients ignoring user-specific preferences. In comparison, our dual personalization mechanism learns fine-grained personalization which fits user preferences.

6.4 Enhance Federated Recommendation Methods with Our Dual Personalization Mechanism

This paper proposes a lightweight dual personalization mechanism to enhance personalization handling, which can be easily integrated into federated learning methods. Particularly, we take FedMF, FedNCF and FedRecon as examples to verify the efficacy of the dual personalization mechanism.

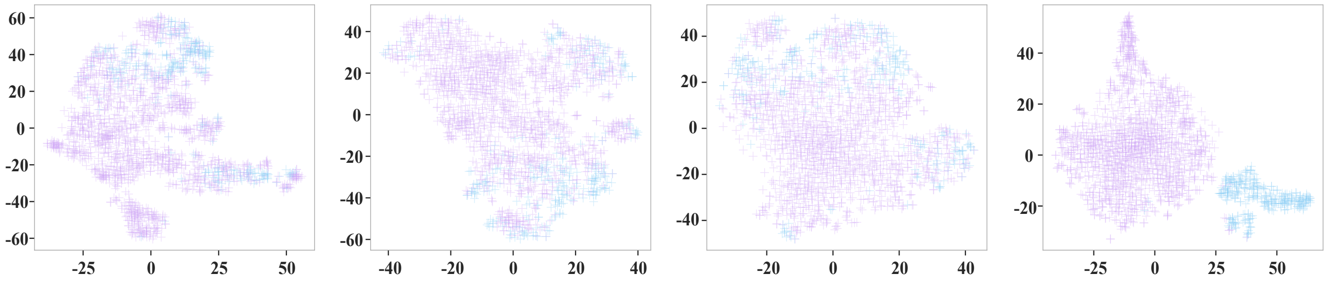
Results & discussion. According to Table 3, all three federated recommendation methods are significantly improved by integrating our dual personalization mechanism. Among them, FedNCF attains the most remarkable boost. The highest HR@10 and NDCG@10 increase exist on MovieLens-100K, *i.e.,* 13.53% and 18.29%. Compared with Lastfm-2K and Amazon-Video, the improvement of the dual personalization mechanism is more evident on the two MovieLens datasets, almost around 10%, where each user has more samples locally and facilitates user preference capture. In summary, our proposed dual personalization mechanism can help the local model to learn user-specific item embedding, which benefits the recommendation system prominently.

6.5 A Close Look of Personalization in PFedRec

To further verify and analyze the role of personalized item embedding in our method, we conduct empirical experiments to answer the following questions:

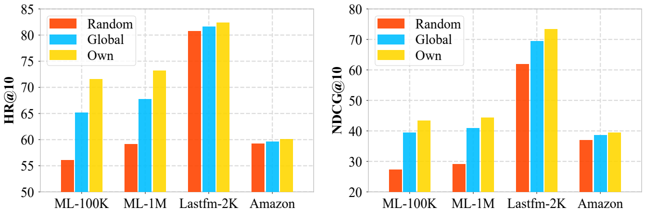
- ***Q1***: *Why personalized item embeddings benefit recommendation more than the global one?*
- ***Q2***: *How specific are the personalized item embeddings among users?*

To answer *Q1*, we first discuss its straightforward insight, then we present visualization to demonstrate our claim. The recommendation system aims to provide user-specific recommendations by exploiting historical interactions. In the FedRec setting, item embedding is consistently considered to maintain the common characteristics among users, and its role in depicting user-specific preferences has been neglected. On the other hand, describing users with common item embedding introduces noisy information, which may incur unsuitable recommendations. Through personalizing item embedding, we enhance personalization modeling in federated learning methods, which depicts the user-specific preference.



(a) FedNCF (b) FedRecon (c) FedMF (d) PFedRec

Figure 2: TSNE visualization of item embeddings learned by baselines and our method.



(a)…Performance…of…HR (b)…Performance…of…NDCG

Figure 3: Client inference using different item embeddings.

Dataset	Noise strength	λ=0	λ=0.1	λ=0.2	λ=0.3	λ=0.4	λ=0.5
ML-100K	HR@10	**71.62**	71.45	71.26	71.13	70.84	70.88
	NDCG@10	**43.44**	43.36	43.30	43.22	43.14	43.21
ML-1M	HR@10	**73.26**	73.13	73.19	73.05	73.18	73.08
	NDCG@10	**44.36**	44.16	44.25	44.26	44.23	44.18
Lastfm-2K	HR@10	**82.38**	82.04	81.91	81.85	81.98	81.88
	NDCG@10	**73.19**	72.41	72.23	72.43	72.39	72.36
Amazon	HR@10	**60.08**	59.31	59.29	59.21	59.15	59.06
	NDCG@10	**39.12**	37.97	37.92	37.83	37.81	37.34

Table 4: Performance of integrating LDP into our method with various Laplacian noise strength λ.

We compare the item embedding learned by baselines (top-3 FedRec methods for performance due to limited space) with our method. Particularly, we select a user randomly from the MovieLens-100K dataset and visualize the embeddings by mapping them into a 2-D space through t-SNE [Maaten and Hinton, 2008]. In this paper, we mainly focus on the implicit feedback recommendation, so each item is either a positive or negative sample of the user. As shown in Figure 2, the item embeddings of positive (blue) and negative (purple) samples are mixed in baselines, where all users share the global item embeddings. However, they can be divided into two clusters by PFedRec. We can easily conclude that our model learns which items the user prefers.

To answer *Q2*, we compare three usages of item embedding during inference as follows:

– ***Random***: Each client is assigned with item embedding from a random user, *i.e.,* every client runs with its score function and item embedding from a random user.

– ***Global***: We assign each client with globally shared item embedding, *i.e.,* every client runs with its score function and global item embedding.

– ***Own***: It follows our setting that every client runs with its score function and personalized item embedding.

Specifically, we first train PFedRec, then assign the learned item embeddings as the above three ways for inference. As shown in Figure 3, clients with their item embedding achieve the best performance, and clients with item embedding from others degrade significantly. Item embedding from a random user contains little helpful information for inference, even less than the common characteristics in global item embedding. The personalized item embedding learned by PFedRec has been adapted to client preference, and different clients achieve rather distinct item embeddings, depicting the user-specific preference.

6.6 Protection with Local Differential Privacy

To enhance the preservation of user privacy, we integrate the Local Differential Privacy (LDP) technique [Choi *et al.*, 2018] into our framework. Particularly, we add the zero-mean Laplacian noise to the client's item embedding before uploading to the server, *i.e.,* $\theta^m = \theta^m + Laplace(0, \lambda)$, and λ is the noise strength. We set $\lambda = [0, 0.1, 0.2, 0.3, 0.4, 0.5]$ to test our method's performance.

As shown in Table 4, performance declines slightly as the noise strength λ grows, while the performance drop is still acceptable. For example, when we set $\lambda = 0.4$, the performance is also better than baselines in most cases. Hence, a moderate noise strength is desirable to achieve a good balance between recommendation accuracy and privacy protection.

7 Conclusion

This paper proposes a novel personalized federated recommendation framework to learn many on-device models simultaneously. We are the first to design the dual personalization mechanism that can learn fine-grained personalization on both users and items. This work could be fundamental work to pave the way for implementing a new service architecture with better privacy preservation, fine-grained personalization, and on-device intelligence. Given the complex nature of modern recommendation applications, such as cold-start problems, dynamics, using auxiliary information, and processing multi-modality contents, our proposed framework is simple and flexible enough to be extended to handle many new challenges. Moreover, the proposed dual personalization is a simple-yet-effective mechanism to be easily integrated with existing federated recommendation systems.

Acknowledgements

Chunxu Zhang and Bo Yang are supported by the National Key R&D Program of China under Grant Nos. 2021ZD0112501 and 2021ZD0112502; the National Natural Science Foundation of China under Grant Nos. U22A2098, 62172185, 62206105 and 62202200; Jilin Province Capital Construction Fund Industry Technology Research and Development Project No. 2022C047-1; Changchun Key Scientific and Technological Research and Development Project under Grant No. 21ZGN30.

References

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

[Cantador *et al.*, 2011] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.

[Chai *et al.*, 2020] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *IEEE Intelligent Systems*, 2020.

[Chen *et al.*, 2022] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with a graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2575–2582, 2022.

[Chen *et al.*, 2023] Shengchao Chen, Guodong Long, Tao Shen, and Jing Jiang. Prompt federated learning for weather forecasting: Toward foundation models on meteorological data. *arXiv preprint arXiv:2301.09152*, 2023.

[Choi *et al.*, 2018] Woo-Seok Choi, Matthew Tomei, Jose Rodrigo Sanchez Vicarte, Pavan Kumar Hanumolu, and Rakesh Kumar. Guaranteeing local differential privacy on ultra-low-power systems. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pages 561–574. IEEE, 2018.

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

[Fallah *et al.*, 2020] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[Harper and Konstan, 2015] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

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

[Kairouz *et al.*, 2019] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[Li *et al.*, 2021] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[Lin *et al.*, 2020a] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. Fedrec: Federated recommendation with explicit feedback. *IEEE Intelligent Systems*, 2020.

[Lin *et al.*, 2020b] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. Meta matrix factorization for federated rating predictions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 981–990, 2020.

[Lin *et al.*, 2021] Zhaohao Lin, Weike Pan, and Zhong Ming. Fr-fmss: federated recommendation via fake marks and secret sharing. In *Fifteenth ACM Conference on Recommender Systems*, pages 668–673, 2021.

[Liu *et al.*, 2022a] Shuchang Liu, Yingqiang Ge, Shuyuan Xu, Yongfeng Zhang, and Amelie Marian. Fairness-aware federated matrix factorization. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 168–178, 2022.

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

[Long *et al.*, 2020] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated learning*, pages 240–254. Springer, 2020.

[Long *et al.*, 2021] Guodong Long, Tao Shen, Yue Tan, Leah Gerrard, Allison Clarke, and Jing Jiang. Federated learning for privacy-preserving open innovation future on digital health. In *Humanity Driven AI: Productivity, Well-being, Sustainability and Partnership*, pages 113–133. Springer, 2021.

[Long *et al.*, 2023] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023.

[Ma *et al.*, 2022] Jie Ma, Guodong Long, Tianyi Zhou, Jing Jiang, and Chengqi Zhang. On the convergence of clustered federated learning. *arXiv preprint arXiv:2202.06187*, 2022.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Minto *et al.*, 2021] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. Stronger privacy for federated collaborative filtering with implicit feedback. In *Fifteenth ACM Conference on Recommender Systems*, pages 342–350, 2021.

[Muhammad *et al.*, 2020] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 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*, pages 1234–1242, 2020.

[Ni *et al.*, 2019] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

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

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[Shamsian *et al.*, 2021] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021.

[Singhal *et al.*, 2021] Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, John Rush, and Sushant Prakash. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[Tan *et al.*, 2022a] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. *arXiv preprint arXiv:2211.13009*, 2022.

[Tan *et al.*, 2022b] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[Tan *et al.*, 2022c] Yue Tan, Guodong Long, Jie Ma, LU LIU, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. In *Advances in Neural Information Processing Systems*, 2022.

[Wang *et al.*, 2016] Meng Wang, Weijie Fu, Shijie Hao, Dacheng Tao, and Xindong Wu. Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1864–1877, 2016.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[Wu *et al.*, 2022a] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Fedctr: Federated native ad ctr prediction with cross platform user behavior data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2022.

[Wu *et al.*, 2022b] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1):1–10, 2022.

[Wu *et al.*, 2022c] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 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*, page 4164–4172, 2022.

[Yi *et al.*, 2021] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. Efficient-fedrec: Efficient federated learning framework for privacy-preserving news recommendation. *arXiv preprint arXiv:2109.05446*, 2021.

[Zhang *et al.*, 2022] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Quoc Viet Hung Nguyen, and Lizhen Cui. Pipattack: Poisoning federated recommender systems for manipulating item promotion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1415–1423, 2022.