DUAL PERSONALIZATION ON FEDERATED RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

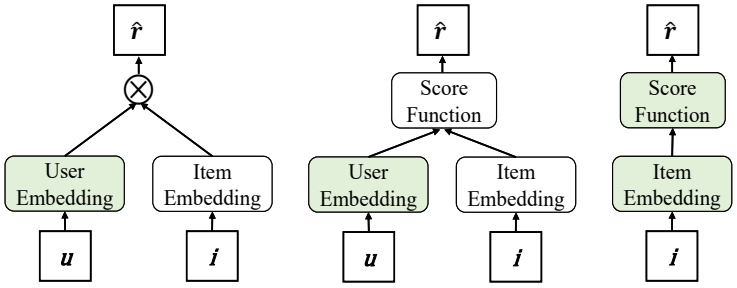
Federated recommendation is a new Internet service architecture that aims to provide privacy-preserving recommendation services in federated settings. Existing solutions are used to combine distributed recommendation algorithms and privacy-preserving mechanisms. Thus it inherently takes the form of heavyweight 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 RecSys in federated settings.

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 Ammad-Ud-Din et al. (2019); Chai et al. (2020); Muhammad et al. (2020); Perifanis & Efraimidis (2022); Singhal et al. (2021) 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 twofold 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



(a) FedMF/FedRecon (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.*

function is to mimic the user's personal decision logic that is natural to be diverse across clients. Furthermore, given an itemset, different people may have a different view to measure 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 users' preferences, and it could be implemented using a multi-layer neural networks. Second, we remove the user embedding from the framework because the current neural-based personalized score function has enough representation capability to preserve the information of user embeddings. Third, we implement a light fine-tuning to learn personalized item embeddings [1] 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 its own data on the device, and then it won't be sent to the server for model aggregation that usually generates a general view for all devices. Moreover, the personalized item embedding will be implemented through light fine-tuning in a federated learning framework, thus it can leverage both the general view from server and the personalized view from its 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 which can better fit federated learning.
- We design a novel dual personalization to capture user preferences using a personalized score function and to 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 optimization framework with the meta-learning objective.
- Our method can significantly outperform existing federated recommendation baselines.

2 Related Work

2.1 Personalized federated learning

Federated Learning (FL) 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); Bonawitz et al. (2019); Yang et al. (2019). 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

[1] When the items are one-hot encoding vectors, we can simply equivalent use item embedding network and item embedding representations.

with non-IID data. **Personalized Federated Learning (PFL)** is to learn a personalized model for each device to tackle the non-IID challenge. A simple PFL method could train a global model with FedAvg, then conduct a few steps of finetuning on each client Cheng et al. (2021). In this framework, knowledge sharing is model aggregation, and model personalization is local finetuning. Per-FedAvg Fallah et al. (2020) adds finetuning as a regularization term to the objective learning function of the global model. Ditto Li et al. (2021) proposes a bi-level optimization framework for PFL while constraining the distance between the local and global models. Investigations by Shamsian et al. (2021); Chen et al. (2018) aim to train a global hyper-network or meta-learner instead of a global model before sending it to clients for local optimization. SCAFFOLD Karimireddy et al. (2020) proposes to learn personalized control variate that corrects the local model accordingly. 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. Layer-wise personalization Arivazhagan et al. (2019); Liang et al. (2020) is also a simple and effective technique in PFL.

2.2 Federated Recommendation Systems

Federated recommendation attracts much attention recently due to the rising concern of privacy. 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 on user privacy protection, Chai *et al.* present FedMF Chai et al. (2020), which adapts distributed matrix factorization to FL setting and introduces the homomorphic encryption technique on gradients before uploading to the server. MetaMF Lin et al. (2020b) is a federated meta-learning framework where a meta-network is adopted to generate the rating prediction model and private item embedding. 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 large news model on server and a light user model on client, and reduces the computation and communication cost for users. FedNCF Perifanis & Efraimidis (2022) adapts Neural Collaborative Filtering (NCF) He et al. (2017) to federated setting which introduces neural network to learn user-item interaction function. Wu et al. (2021) presents FedGNN where each user maintains a GNN model to incorporate high-order user-item information. Lin et al. (2020a); Du et al. (2021); Yang 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. (2022b); Zhang et al. (2022), social recommendation Liu et al. (2022b), Click-Through Rate (CTR) prediction Wu et al. (2022a), fair recommendation Liu et al. (2022a) and payload optimization Khan et al. (2021).

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 accumulated loss of all clients. That is,

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

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 federated learning 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. The learning objective is usually formulated 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 on Neural Networks This work focuses on a fundamental scenario where recommendation only relies on the user-item interaction matrix. The recommendation task is then fulfilled by the Neural Collaborative Filtering (NCF) model He et al. (2017), which consists of three parts: a score function S, a user embedding module E and an item embedding module $\mathcal{E}$. We denote these modules' parameters as $\theta := (\theta^s, \theta^u, \theta^m)$ and formulate the learning objective in Eq. 4

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

where e^u and e^m are one-hot encoding representing users and items. r is a user's rate to the given item and $\hat{r}$ is a prediction from the score function $S(E(e^u), \mathcal{E}(e^m))$. L is the loss evaluating prediction performance. It 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 NCF He et al. (2017), *i.e.,* the conventional MF is a model where the score function S is simplified as the multiplication 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 NCF models deployed on end devices.

4.1 OBJECTIVE FUNCTION

Federated Learning Objectives We regard each user as a client under FL settings. The on-device recommendation task is then depicted as a PFL problem. Particularly, a personalized item embedding module E_i is assigned to be a global module which learns common item information. Further, as introduced in Sec. 1, personalization is essential to capture a client's local decision logic. 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.

Accordingly, the **local loss** function on the i-th client is formulated as

$$L_i(\theta_i; r, \hat{r}) := L_i(\theta_i; r, S_i(E_i(e^u))) \tag{5}$$

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. Besides, incorporating the PFL framework in Eq. 2, the collaborative learning is achieved by optimizing the following **global objective** function

$$\min_{\theta^m, \{\theta_i\}_{i=1}^N} \sum_{i=1}^{N} \alpha_i L_i(\theta_i; r, \hat{r}) \tag{6}$$

where θ^m is the shared parameter for the global item embedding module E, and there is

$$\theta_i := (\theta^m - \nabla_{\theta^m} L_i, \theta_i^s) \tag{7}$$

as the personalized item embedding module E_i is obtained by tuning E on the client's local data.

Loss for Recommendation We discuss the typical recommendation task with implicit feedback, that is, $r_{ij} = 1$ if the i-th user interacted with the 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*,

$$L_i(\theta_i; r_{ij}, \hat{r}_{ij}) = -\sum_{(i,j)\in D_i} \log \hat{r}_{ij} - \sum_{(i,j')\in D_i^-} \log(1 - \hat{r}_{ij'}) \tag{8}$$

where D_i^- is the negative instances set of user and $\hat{r}_{ij}$ denotes the corresponding prediction of the score function. To construct D_i^- efficiently, we uniformly sample negative instances from unobserved interactions and set the sampling ratio according to the number of observed interactions. Notably, other loss functions can also be used, and here we choose the binary cross-entropy loss to simplify the description.

4.2 DUAL PERSONALIZATION

We have implemented 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 could 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 network. The user's personalized score function network θ^s won't be sent to the server and thus won't be aggregated.

Fine-tuning the item embedding module to generate personalized representations for items on each device. According to Eq. 7, 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 network, and then replaces the global item embedding with personalized item embedding obtained by fine-tuning θ^m. Details of the learning process is illustrated in Algorithm. 1 and extensive experiments show it will achieve superior performance to the vanilla meta-learning based methods.

4.3 ALGORITHM

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

Workflow. The overall workflow of the algorithm 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 network from each client to perform global aggregation. We build on the simplified version of FedAvg, a direct average of locally uploaded item embedding network. The overall procedure is summarized in Algorithm 1.

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 minization rules (GDPR) that could greatly mitigate the risk of privacy leakage Kairouz et al. (2019). To provide the service with privacy guranttee, 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 vaniall FL to preserve data locality. For example, to tackle the privacy leakage risk caused by sending the item embedding network to the server, we could simply apply differential privacy to inject noise into the gradients so that the server cannot simply infer the updated items by watching the changes of gradients.

Algorithm 1 Dual Personalization for Federated Recommendation

ServerExecute:

Initialize item embedding $\theta^m \leftarrow \theta_0^m$ and score function $\theta^s \leftarrow \theta_0^s$
for each round $t = 1, 2, ...$ **do** ▷ Global communication rounds
  $S_t \leftarrow$ (select a set of size n randomly from all N clients)
  for each client index $i \in S_t$ **in parallel do**
    $\theta_i^m \leftarrow$ ClientUpdate(i, θ^m) ▷ Distribute global item embedding to client for update
  $\theta^m \leftarrow \frac{1}{n}\sum_{i=1}^{n} \theta_i^m$ ▷ Global aggregation over n local item embedding network

ClientUpdate:

Initialize θ_i^m with θ^m
Initialize θ_i^s with the latest updates
Retrieve user positive feedback D_i according to user index i
Sample negative feedbcak D_i^- from unobserved instances
$\mathcal{B} \leftarrow$ (split $D_i \cup D_i^-$ into batches of size B)
for e from 1 to E **do** ▷ Local training epochs
  for batch $b \in \mathcal{B}$ **do**
    Compute $L_i(\theta_i; r, \hat{r})$ with Eq. 8 ▷ Model loss of batch data b
    $\theta_i^s \leftarrow \theta_i^s - \eta \nabla_{\theta^s} L_i$ ▷ Score funtion update
    Compute $L_i(\theta_i; r, \hat{r})$ with Eq. 8 ▷ Model loss with the updated θ_i^s
    $\theta_i^m \leftarrow \theta_i^m - \eta' \nabla_{\theta^m} L_i$ ▷ post-tuning for personalized item embedding network
Return θ_i^m to server

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 to an equivalent form of other frameworks. For example, if we treat the score function as a single-layer neural network, then we can be equivalent to FedMF and FedReconn in Figure 1 (a). Moreover, if we change the personalized score function from full personalization to partial layer personalization, our framework could be equivalent to FedNCF in Figure 1 (a) which has a shared score function across clients. Furthermore, our proposed framework's architecture could be naturally aligned to the classic neural network architecture, thus it has a bigger potential to achieve a 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, Lastfm-2K and Amazon-Video. 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 **Appendix 1.1**.

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. Details can be referred to **Appendix 1.2**.

6.2 BASELINES AND IMPLEMENTATION DETAILS

Baselines Our method is compared with baselines in both centralized and federated settings.

- **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 user and item, 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-layers MLP for comparison, adopted in the original paper.
- **FedMF** Chai et al. (2020): It is a federated version of MF and a typical FedRec method. It updates user embedding locally and uploads item gradients to the server for global update.
- **FedNCF** Perifanis & Efraimidis (2022): It is a federated version of FedNCF. Specifically, each user updates user embedding locally and uploads item embedding network 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.

Implementation details We randomly sample 4 negative samples for each positive sample, following the setup in He et al. (2017). For a fair comparison, we keep the same latent user (item) embedding size for all methods, *i.e.,* 32. We set other model details of baselines according to their original papers. We use a fixed batch size of 256 and search the learning rate in $[0.0001, 0.001, 0.01, 0.1]$. For our method, we implement a one-layer MLP as the score function for simplification, which can be regarded as an enhanced FedMF with the dual personalization mechanism. We implement the methods based on the Pytorch framework and run all the experiments for 5 repetitions and report the average results.

6.3 Comparison analysis

We conduct experiments on four datasets for performance comparison.

	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
Rec	**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	**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
	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
	PFedRec (Ours)	$\mathbf{71.62 \pm 0.83}$	$\mathbf{43.44 \pm 0.89}$	$\mathbf{73.26 \pm 0.20}$	$\mathbf{44.36 \pm 0.16}$	$\mathbf{82.38 \pm 0.92}$	$\mathbf{73.19 \pm 0.38}$	$\mathbf{60.08 \pm 0.08}$	$\mathbf{39.12 \pm 0.09}$

Table 1: *Performance of HR@10 and NDCG@10 on four datasets.* ***Rec*** *and* ***FedRec*** *represent centralized and federated methods, respectively. The results are the mean and standard deviation of five repeated trials.*

Results & discussion As shown in Table 1, *(1)* PFedRec obtains better performance than centralized methods in some cases. In centralized scenario, only user embedding is regarded as 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. These two contain more samples for training, which supports the personalization depiction. *(3)* PFedRec consistently achieves the best performance on all settings. In FedRec, the common item embedding helps transfer the shared information among users, which facilitates collaborative training of individual user models. However, different users present rather distinct preferences for items. Our dual personalization mechanism offers fine-grained personalization which fits the local data. It filters out the interference of redundant information and obtains better performance.

We also analyse spacial and computational efficiency of our proposed model and compare with baselines, including parameter volume, training epoch and running time. Specifically, our proposed method achieves best performance with relatively low level of running time consumption. It shows that our dual personalization mechanism will not increase spatial and computational compexity. Detailed analysis could be found in **Appendix 1.3**. Besides, we present the convergence curves of PFedRec and all baselines on all datasets in **Appendix 1.4**. It shows PFedRec is the consistently the fastest one to converge, which emphasizes our method's advanced efficiency.

6.4 INTEGRATING BASELINES WITH OUR DUAL PERSONALIZATION MECHANISM

This paper proposes a lightweight dual personalization mechanism to enhance personalization handling, which can be easily transferred to nearly-all federated learning methods. We apply it to FedRec baselines to exhibit its efficacy.

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
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%
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%

Table 2: *Performance improvement for integrating our dual personalization mechanism to baseline algorithms.* ***Improvement*** *denotes the performance gain beyond the baselines due to incorporating our dual personalization mechanism (**DualPer**). The results are the mean and standard deviation of five repeated trials, and the significant improvemments (over 5%) are highlighted.*

Results & discussion According to Table 2, all the baselines are significantly improved by integrating our dual personalization mechanism since our mechanism enhances their modeling of user personalization. The highest HR and NDCG increases exist at FedNCF on MovieLens-100K, *i.e.,* 13.53% and 18.29%. The enhancement of FedNCF attains the most remarkable boost, which emphasizes the necessity of learning personalized item embedding for each user and the capacity of our dual personalization mechanism. Comparing with Lastfm-2K and Amazon-Video, the improvement of this mechanism is more evident on the two MovieLens datasets, almost around 10%, where each user has more samples locally. In summary, our proposed dual personalization mechanism can help the local model to learn more information when training personalized item embedding, which benefits the recommendation system prominently.

Impact Factor Analysis To further analysis the efficacy of PFedRec, we design comprehensive impact factor influence analysis, including latent embedding size, negative sample size and clients volume participationg in each round. In a nutshell, the best embedding size for all datasets is 32. Generally, the performance grows gradually as the negative samples increases. PFedRec is able to obtain the best performance regardless of different volume of client samples, while more clients facilitate the convergence. For more details please refer to **Appendix 1.5**.

6.5 A CLOSE LOOK OF PERSONALIZATION IN PFEDREC

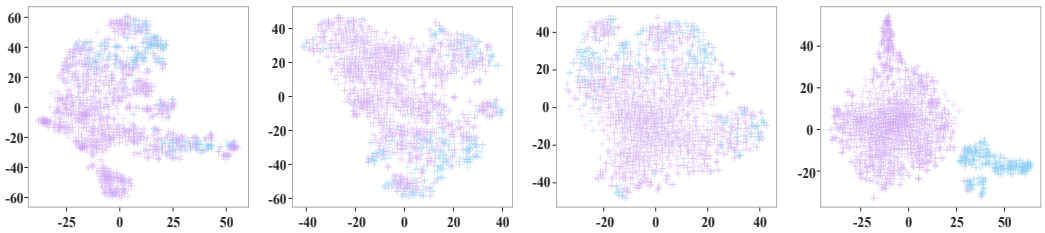
In our method, we learn the personalized item embedding network for each user based on the global item embedding to learn fine-grained personalization. To further verify and analyze the role of personalized item embedding, we conduct empirical experiments to answer the following two questions:

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

To answer *Q1*, We first discuss its straightforward insight, then we present visualization to demonstrate our claim. The recommendation system is supposed 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 leads in 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. We will demonstrate how precise the personalized embedding describes user preference of our method compared with baselines.

We compare the item embedding learned by baselines 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 & 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 together in baselines. However, they can be obviously divided into two clusters by PFedRec. We can easily conclude that our model learns which items the user prefers.



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

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

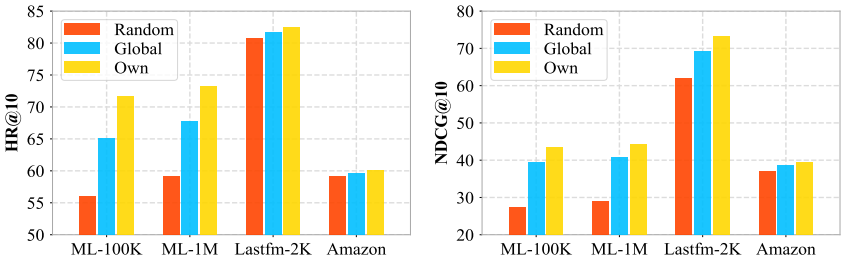
To answer *Q2*, we compare three usages of item embedding as follows:

– ***Another random user***: Each client is assigned with item embedding from another random user, *i.e.,* every client runs with its score function and item embedding from another 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 is the setting of our PFedRec, *i.e.,* every client runs with its own score function and item embedding.

Specifically, we first train PFedRec, then assign the learned item embeddings as the above three ways for inference. Experimental results are 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 another user contains a relatively low level of 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 the client preference, and different clients achieve rather distinct item embeddings, which depict the user-specific information.



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

Figure 3: *Client inference using different item embeddings.*

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 a 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 tackle many new challenges. Moreover, the proposed dual personalization is a simple-but-effective mechanism to be easily integrated with existing federated recommendation systems.

REFERENCES

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.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1: 374–388, 2019.

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

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 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.

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.

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*, pp. 173–182, 2017.

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.

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.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Farwa K Khan, Adrian Flanagan, Kuan Eeik Tan, Zareen Alamgir, and Muhammad Ammad-Ud-Din. A payload optimization method for federated recommender systems. In *Fifteenth ACM Conference on Recommender Systems*, pp. 432–442, 2021.

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

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.

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

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

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

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*, pp. 981–990, 2020b.

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

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*, pp. 168–178, 2022a.

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, 2022b.

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

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*, pp. 1273–1282. PMLR, 2017.

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*, pp. 342–350, 2021.

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*, pp. 1234–1242, 2020.

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

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

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pp. 9489–9502. PMLR, 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.

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.

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*, pp. 165–174, 2019.

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

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)*, 2022a.

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*, pp. 4164–4172, 2022b.

Liu Yang, Ben Tan, Bo Liu, Vincent W Zheng, Kai Chen, and Qiang Yang. Practical and secure federated recommendation with personalized masks. *arXiv preprint arXiv:2109.02464*, 2021.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

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.

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*, pp. 1415–1423, 2022.