Federated Adaptation for Foundation Model-based Recommendations

Chunxu Zhang\textsuperscript{1,2,*}, Guodong Long\textsuperscript{3}, Hongkuan Guo\textsuperscript{4}, Xiao Fang\textsuperscript{4}, Yang Song\textsuperscript{1}, Zhaojie Liu\textsuperscript{4}, Guorui Zhou\textsuperscript{4}, Zijian Zhang\textsuperscript{1,2}, Yang Liu\textsuperscript{5†}, Bo Yang\textsuperscript{2,2†}

\textsuperscript{1}Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China University of Science and Technology, China
\textsuperscript{2}College of Computer Science and Technology, Jilin University, China
\textsuperscript{3}Australian Artificial Intelligence Institute, FEIT, University of Technology Sydney
\textsuperscript{4}Kuaishou Technology
\textsuperscript{5}Institute for AI Industry Research, Tsinghua University

\{cxzhang19, zhangzj2114\}@mails.jlu.edu.cn, guodong.long@uts.edu.au, \{guohongkuan, zhouguorui\}@kuaishou.com, \{ustcfx0727, liuzj03\}@gmail.com, ys@sonyis.me, liuy03@air.tsinghua.edu.cn, ybo@jlu.edu.cn

Abstract

With the recent success of large language models, particularly foundation models with generalization abilities, applying foundation models for recommendations becomes a new paradigm to improve existing recommendation systems. It becomes a new open challenge to enable the foundation model to capture user preference changes in a timely manner with reasonable communication and computation costs while preserving privacy. This paper proposes a novel federated adaptation mechanism to enhance the foundation model-based recommendation system in a privacy-preserving manner. Specifically, each client will learn a lightweight personalized adapter using its private data. The adapter then collaborates with pre-trained foundation models to provide recommendation service efficiently with fine-grained manners. Importantly, users’ private behavioral data remains secure as it is not shared with the server. This data localization-based privacy preservation is embodied via the federated learning framework. The model can ensure that shared knowledge is incorporated into all adapters while simultaneously preserving each user’s personal preferences. Experimental results on four benchmark datasets demonstrate our method’s superior performance. The code is available.

1 Introduction

Recently, the Foundation Models (FMs) [Radford et al., 2019; Bommasani et al., 2021; Achiam et al., 2023] emerge rapidly and have made breakthroughs in various AI applications, ranging from language [Alayrac et al., 2022], vision [Saharia et al., 2022], reasoning [Kojima et al., 2022] and recommendation [Geng et al., 2022]. FMs are typically trained on extensive data sources, allowing them to capture and utilize inherent common knowledge. This capability empowers FMs to achieve outstanding performance in various downstream tasks. Applying foundation models to recommendation systems is considered a highly promising direction, which has significantly propelled the state-of-the-art in recommendation system studies [Harte et al., 2023; Liu et al., 2023a; Lin et al., 2023].

Two new open challenges have been encountered when we introduce the foundation models into a practical recommendation system. First, given the fast changes in user preference, how to timely update the foundation models-based recommendation system with reasonable cost on communication and computation. An on-device parameter-efficient fine-tuning mechanism is desired to be incorporated with the foundation models. The second challenge is how to tackle the privacy-sensitive data of users that is needed to train or fine-tune the foundation model in a recommendation system.

Federated learning trains the global model by iterative model parameters transmission between server and clients without accessing private client data [McMahan et al., 2017; Miao et al., 2023; Zhong et al., 2023; Liu et al., 2024]. Due to its excellent privacy-preserving properties, federated learning has become a popular privacy protection enhanced scheme for recommendation, named federated recommendation systems [Chai et al., 2020; Yang et al., 2020; Wu et al., 2022; Sun et al., 2022; Zhang et al., 2023a; Li et al., 2023]. Trained on extensive data, the foundation model’s attribute embedding and prediction function retain valuable general domain knowledge and user decision logic. Integrating federated learning into foundation model-based recommendation systems can benefit from this shared knowledge while ensuring privacy preservation. However, effectively addressing this integration remains unsettled.

Several challenges must be addressed urgently to adapt foundation model-based recommendations with federated learning frameworks. \textbf{Challenge 1: efficient user personalization modeling}. Given the substantial preference differences across isolated clients, the uppermost goal for this recommendation system is effective personalization modeling of diverse users under the privacy protection limitation.
At the same time, there are usually concerns about efficiency when using large-scale foundation models, which is more urgent in federated learning that requires end-device optimization. **Challenge 2: common knowledge and user personalization fusion.** Common knowledge maintained in the pre-trained foundation model can incorporate insights from collective user behavior. By integrating it with user personalization, the federated recommendation system can leverage the collective intelligence of the user community, leading to improved decision-making. However, the importance of common knowledge and user personalization varies across users, and ineffective knowledge fusion can lead to information confusion and misleading recommendations. Hence, balancing general knowledge and individual user preference emerges as a main challenge.

In this paper, we present a novel method, named **Federated recommendation with Personalized Adapter (FedPA)**, to explore the federated foundation model for recommendation. Our method leverages the pre-trained model as the foundation, allowing us to incorporate common knowledge and optimize the federated recommendation system from a well-established starting point. To capture user personalization efficiently, we propose a personalized adapter to deploy on the client, which can learn individual user preferences in a lightweight manner. Then, we learn these adapters with the pre-trained model in an adaptive fusion manner to balance the collaboration of common knowledge and user personalization in federated optimization.

Motivated by research about neural network optimization [Li et al., 2018; Aghajanyan et al., 2020], the model parameter solution for the target task usually resides within an intrinsic dimension. In the federated recommendation system, each user has task-specific intrinsic parameter space that can be learned from personal data. To this end, we design a low-rank adapter to learn user personalization in a lightweight manner. Particularly, we develop two levels of personalization to accommodate the recommendation scenario, including user-level and user-group-level. They cater to the unique preferences of individual users while also capturing and leveraging shared patterns and preferences within specific user groups. Furthermore, we design an adaptive gate learning mechanism that dynamically learns the weights for common knowledge and user personalization, enabling effective knowledge fusion. During federated optimization, our FedPA focuses on updating only the parameters relevant to user-specific modeling and the others are frozen and exempted from optimization, leading to a significant reduction in communication cost and achieving faster convergence.

We assess the performance of FedPA on four benchmark datasets and compare it with various advanced baselines. Experimental results consistently demonstrate that our method outperforms the baselines by a significant margin. In addition, we conduct comprehensive experiments to analyze FedPA’s ability to capture user personalization and the impact of common knowledge in the pre-trained model on federated recommendation systems. Furthermore, we validate the model’s feasibility in real-world applications. By distilling the pre-trained model into a smaller size, we address the computational and storage challenges of deploying the pre-trained model on edge devices. Additionally, we enhance privacy protection by leveraging the Local Differential Privacy technique. Experimental results demonstrate FedPA’s stable performance with distilled smaller models and privacy preservation, affirming its practical applicability. To summarize, the main contributions are listed as follows:

- For the first time, we investigate the federated adaptation paradigm for foundation model-based recommendation, named FedPA. It enables the integration of the rich knowledge encapsulated within pre-trained models while upholding privacy protection for users.
- We present a personalized low-rank adapter to learn user personalization from user-level and user-group-level in a lightweight manner. Furthermore, we design an adaptive gate learning mechanism to dynamically learn weights, allowing for the effective fusion of common knowledge with user personalization.
- Extensive experiments on four benchmark datasets demonstrate the superior performance of FedPA against advanced baselines. Additionally, FedPA also shows excellent feasibility in deploying on clients with limited computation capability and strengthening user privacy protection in federated recommendation systems.

## 2 Related Work

### 2.1 Foundation Models for Recommendation

Pre-training in Natural Language Processing (NLP) [Qiu et al., 2020] has witnessed significant progress, with language models like GPT [Brown et al., 2020] and BERT [Devlin et al., 2018] achieving state-of-the-art results. The pre-training and fine-tuning paradigm allows for the extraction of valuable knowledge and eliminates the need to train new models from scratch. Given its remarkable benefits, increasing research has been on developing foundation models for recommendation systems [Liu et al., 2023a; Wu et al., 2023]. The essential learning objective of recommendation is to estimate the user’s preferences for a certain item set. By incorporating the foundation models into the recommendation system, it can absorb valuable knowledge and improve the system’s ability for characteristics extraction and user decision pattern learning. However, existing foundation models for recommendation rely on collecting personal data from users to optimize, which poses severe risks to protecting user privacy.

### 2.2 Federated Recommendation System

Federated recommendation system is a rising service paradigm that can learn the model in a privacy-preserving manner [Lin et al., 2020; Perifanis and Efraimidis, 2022; Qu et al., 2023; Zhang et al., 2023b]. Existing studies focus on developing federated recommendation models with mainstream recommendation architectures, e.g., matrix factorization [Chai et al., 2020] and neural collaborative filtering [Perifanis and Efraimidis, 2022], popular recommendation tasks, e.g., POI prediction [Zhang et al., 2023c] and multi-domain recommendation [Liu et al., 2023c], and user privacy protection enhancement [Liu et al., 2023b; Huang et al., 2023]. In this paper, we investigate the integration of foundation model
into federated recommendation system, which can utilize the inherent common knowledge in the pre-trained model and prosper the powerful federated recommendation system.

3 Methodology

We present a novel federated adaption paradigm for foundation model-based recommendation, named FedPA. In this section, we first provide an overview of the framework architecture. Then, we delve into the components to elucidate the details and summarize the workflow into an optimization algorithm. Following that, we engage in a discussion on the feasibility of deploying our FedPA in physical applications. Finally, we develop a privacy-protection enhanced FedPA that can strengthen the system’s protection for user privacy.

3.1 Framework Overview

The pre-trained recommendation model is learned from a large amount of publicly available data, embodying rich knowledge. It can effectively characterize user and item attributes and possesses strong predictive capabilities for user decision patterns. Effective utilization of this common knowledge contributes to building a more powerful federated recommendation model. To achieve this, we take the pre-trained model as the foundation model for the federated recommendation system, enabling federated optimization from a favorable starting point.

To efficiently capture user-specific preferences, we propose a low-rank adapter that models user personalization from both the user-level and user-group-level perspectives. Additionally, we design an adaptive gate learning mechanism that effectively integrates common knowledge and personalized knowledge for better user modeling. The model architecture is illustrated in Figure 1. In this paper, we utilize an existing recommendation architecture as the base model, and it can be easily extended to other popular architectures.

3.2 Base Model

Given user attributes, item attributes, and user interaction records, we adopt a widely used two-tower recommendation model architecture. Specifically, the model consists of two input branches, one for learning user representations based on user attributes and the other for learning item representations based on item attributes. These representations are then fed into a prediction function to estimate user preferences for items. The user interaction records serve as supervision information to guide the model in updating its parameters.

**User Embedding Module.** There are certain attribute information available for the users, e.g., user active degree. For each attribute $i$, we construct a learnable embedding table $E_i \in \mathbb{R}^{p \times d}$, where $p$ is the total attribute categories and $d$ is the embedding dimension. Then, for each user $u$ with attributes $A_u$, we retrieve the embedding vectors from all the attribute embedding tables based on attribute values and ob-
tain the user representation \( u_r \) by concatenating them,

\[
\begin{align*}
    u_r &= \text{Concat}(E_i(A^u_{i,v})_{i=1}^{|A_u|}) \tag{1}
\end{align*}
\]

where \( |A_u| \) is the total attribute number of user \( u \).

**Item Embedding Module.** For each item \( v \), we adopt a similar approach as the user embedding module to construct the item embedding tables and obtain item representation \( v_r \):

\[
\begin{align*}
    v_r &= \text{Concat}(E_i(A^v_{i,v})_{i=1}^{|A_v|}) \tag{2}
\end{align*}
\]

and \( |A_v| \) is the total attribute number of item \( v \).

**Prediction Function Module.** Given user representation \( u_r \) and item representation \( v_r \), we use a simple MLP (Multi-Layer Perceptron) as the prediction function to estimate user preferences for items,

\[
\hat{Y}_{uv} = \text{MLP}(\text{Concat}(u_r, v_r)) \tag{3}
\]

**Loss Function.** To update the model parameters, we construct a loss function that encourages the model’s predictions to be as close as possible to the true labels. For common implicit feedback recommendation tasks, where the label value is \( Y_{uv} = 1 \) when a user \( u \) interacts with an item \( v \) and \( Y_{uv} = 0 \) otherwise, we use binary cross-entropy as the loss function,

\[
L(\theta_{\text{base}}) = -\frac{1}{|D|} \sum_{Y_{u,v} \in D} Y_{uv} \log \hat{Y}_{u,v} + (1 - Y_{uv}) \log (1 - \hat{Y}_{u,v}) \tag{4}
\]

where \( \theta_{\text{base}} \) is the model parameter, including user embedding tables \( \theta_{\text{emb}} \), item embedding tables \( \theta_{\text{emb}} \), and the MLP parameters \( \theta_{\text{mlp}} \). \( D \) is the user-item interaction record set, and \( |D| \) is the total interactions number.

### 3.3 Personalized Adapter

Existing federated recommendation methods typically model user personalization by preserving partial model parameters locally. However, it limits direct access from other clients and potentially hinders the collaborative context utilization. Moreover, the personal data on each client is generally limited, which can introduce biases and compromise model performance. To overcome the challenge, we propose to learn user personalization from two perspectives, i.e., user-level and user-group-level. Particularly, we devise a personalized adapter applied to the prediction function module due to its crucial role in predicting user preferences.

Drawing inspiration from research on neural network optimization, model parameters can be embedded within an intrinsic dimension [Li et al., 2018; Aghajanyan et al., 2020]. To this end, we propose a low-rank adapter that leverages low-rank matrices to model user-specific knowledge on each client. This approach offers two prominent advantages: First, it can learn user personalization from user-specific and user groups with similar characteristics, which enhances the system’s ability to model individual user preferences. Besides, the low-rank matrices introduce only a small number of parameters, making it a parameter-efficient solution.

**Low-Rank adapter.** For each layer \( l \) of MLP in the prediction function module, it maps the input \( x \in \mathbb{R}^d \) into a new space \( \xi \in \mathbb{R}^k \) with the weight matrix \( W_l \in \mathbb{R}^{k \times d} \). We intensify the personalization learning by adding a low-rank decomposition matrix \( W_{lr} = W_a W_b \), where \( W_a \in \mathbb{R}^{k \times r} \) and \( W_b \in \mathbb{R}^{r \times d} \), and the rank \( r \ll \min(d, k) \). Then, the forward pass of each layer can be modified as follows,

\[
\hat{\xi} = W_l x + W_a W_{lr} x \tag{5}
\]

where \( W_{lr} \) is responsible for learning user personalization. In the context of recommendation tasks, we develop two levels of personalization: user-level personalization and user-group-level personalization. They cater to individual user preferences as well as capturing patterns and preferences shared within specific user groups.

**User-Level Personalization.** For the user-level personalization, we aim to learn a specific low-rank adapter for each user so the parameters \( W_a^u \) and \( W_b^u \) would be preserved locally and not be shared globally. For the user \( u \), we formulate the user-specific low-rank adapter as follows,

\[
\hat{\xi}^u = W_a^u W_{lr}^u x \tag{6}
\]

**User-Group-Level Personalization.** For the user-group-level personalization, we aim to learn the same low-rank adapter for users in a specific group. In recommendation systems, users with similar characteristics tend to share similar preferences. To fully leverage this information, we learn multiple groups of low-rank adapters. For each user group \( g \), we formulate the group-specific low-rank adapter as follows,

\[
\hat{\xi}^g = W_a^g W_{lr}^g x \tag{7}
\]

Users within the same group share parameter \( W_a^g \) and \( W_b^g \). It is worth noting that users in the system can be grouped in multiple ways, meaning each user can belong to multiple groups \( \{g_i\}_{i=1}^{\text{total}} \). For example, a user can belong to the “young adults” group while also to the “highly active degree” group. By categorizing users from multiple orthometric perspectives, our model can learn more detailed personalized parameters and enhance the user preference capture.

### 3.4 Adaptive Gate Learning Mechanism

By incorporating the low-rank adapter, the prediction function module can learn both the shared decision patterns among users and the personalized decision logic at two granularities, i.e., user-level and user-group-level. To effectively combine common knowledge and personalized knowledge, we propose an adaptive gate learning mechanism that dynamically assigns weights to each decision branch. Specifically, we utilize a two-layer non-linear mapping to learn the weights for the branches based on the input. The fusion process can be formulated as follows,

\[
\bar{\xi} = \text{Sum}(\text{softmax}(W_2 \text{Relu}(W_1 x)) \odot [\hat{\xi}^u, \hat{\xi}^g, \{\hat{\xi}^h_{i=1}^{\text{total}}\}]) \tag{8}
\]

where \( W_1 \) and \( W_2 \) are the parameters of the adaptive gate learning mechanism, \( \hat{\xi} \) is the output of the MLP layer in the prediction function module. \( \odot \) and \( \text{Sum}(\cdot) \) denote element-wise multiplication and summation calculation, respectively. The parameters of the adaptive gate learning mechanism can be updated based on gradients along with other model parameters, eliminating the cumbersome manual setting of hyperparameters. This approach enhances the flexibility of multi-branch fusion, allowing for adaptive adjustment of the weights for all branches during the model’s training stages.
3.5 Optimization and Model Scalability

Optimization Objective
In the federated recommendation system, each user $u$ acts as a client and trains the local recommendation model according to the personal dataset $D_u$. Let $L_u(\theta_{all}^u)$ denote the local model’s loss function, where $\theta_{all}^u$ contains of base model parameter $\theta_{base}$, low-rank adapter parameter $\theta_{lr}$ and $\{\theta_{lr}^u\}_{l=1}^m$ and adaptive gate learning mechanism parameter $\theta_{gate}$. The overall optimization of the federated recommendation model can be formulated as follows,

$$\min_{\{\theta_{all}^u\}} \sum_{u=1}^{n} \frac{1}{n} L_u(\theta_{all}^u) \tag{9}$$

where $n$ is the total number of clients in the federated recommendation system. Here, we employ a naive average aggregation approach to optimize system parameters. It is also possible to use a more flexible weighted aggregation approach further to enhance the optimization objective [McMahan et al., 2017; Wang et al., 2020].

Efficient Parameter Update
To leverage the knowledge inherent in the pre-trained model effectively, we propose to freeze the item embedding module and prediction function module during federated optimization. However, since the federated recommendation system operates on different user sets, we continue to update the user embedding module to adapt to the specific characteristics. To summarize, during the federated optimization process, we focus solely on updating the parameters related to personalized user modeling, e.g., user embedding, low-rank adapter, and adaptive gate learning mechanism parameters. It effectively saves computational and communication costs.

Discussion about Model Scalability
The pre-trained recommendation models learned with abundant computational resources often have complex structures and large sizes in real-world scenarios. Deploying such models directly to the clients poses significant challenges in terms of limited storage and computational capability. To address this issue, we propose to leverage the knowledge distillation (KD) technique [Gou et al., 2021] to distill the pre-trained model into a smaller-sized recommendation model that can be accommodated by client devices. We then utilize this distilled model to warm up the federated recommendation system. This approach effectively enhances the scalability of the proposed framework in real-world applications.

3.6 Privacy-Preserving Enhanced FedPA

The distributed training nature of federated learning avoids direct exposure to private user data. To further mitigate the risk of the server inferring user privacy through model parameter reverse engineering, we integrate the Local Differential Privacy technique [Choi et al., 2018] into our method, whose basic insight is to add a zero-mean Laplacian noise to the shared model parameters before uploaded to the server. Our method’s shared model parameters include user group-level low-rank adapter and the adaptive gate learning mechanism parameters. By adjusting the intensity of noise, we can control the privacy protection capability of the system, that is, increasing the intensity of the noise enhances the effectiveness of privacy protection.

4 Experiment

In this section, we conduct comprehensive experiments to verify the efficacy and illustrate a deep analysis of various aspects of our proposed FedPA. Implementation code is available to ease reproducibility\(^1\).

4.1 Experimental Setup

Datasets
We evaluate FedPA on four practical industrial recommendation datasets collected from the short video platform Kuaishou\(^2\), i.e., KuaiRand\(^3\) (KuaiRand-Pure and KuaiRand-small) and KuaiSAR\(^4\) (KuaiSAR-S and KuaiSAR-R). For dataset split, we first divide each dataset into two subsets: one for training the foundation model and the other for training the federated recommendation system. The dataset for the federated recommendation system is further split into train, validation, and test sets for each user based on interaction timestamps, with a ratio of 6:2:2.

Evaluation Protocols
We evaluate the model performance by calculating the evaluation metrics on the test set of private users. Specifically, we take widely used AUC (Area Under Curve) and Precision as two evaluation metrics. All experimental results are in units of 1e-2 and the average values of five individual runs. For the user group level low-rank personalization, we group users based on their attribute values. It is important to note that the clients can update the corresponding user group low-rank adaptor parameters based on their own attribute values locally without exposing the attribute values to the server. Additionally, we have incorporated Local Differential Privacy technique to further protect user privacy, hence user privacy protection can be ensured in our method.

Baselines
This paper concentrates on developing a personalized federated recommendation system and investigating the assistance provided by the common knowledge contained in pre-trained models. To assess the feasibility and effectiveness of our proposed FedPA, we compare it with two types of baseline models: (1) Train and evaluate the model on the private user dataset without warm-starting from the pre-trained model, i.e., w/o Warm. (2) Warm-start the federated recommendation models with the pre-trained model, i.e., w/ Warm. Specifically, we select four representative personalized federated recommendation models, including FedNCF [Perifanis and Efraimidis, 2022], PFedNCF [Pillutla et al., 2022], FedRecon [Singhal et al., 2021] and PFedRec [Zhang et al., 2023a], and the corresponding warm-starting variants as baselines. Besides, we remove the warm-starting strategy

\(^1\) Code: https://github.com/Zhangcx19/IJCAI-24-FedPA
\(^2\) https://www.kuaishou.com/cn
\(^3\) https://kuairand.com/
\(^4\) https://kuaisar.github.io/
from our method, named FedPA w/o Warm, to assess the contribution of our personalization modeling insight for the federated recommendation system.

### 4.2 Comparison with Baselines

Table 1 shows the performance of AUC and Precision on four datasets. “w/o Warm” (“w/ Warm”) denotes training the federated recommendation system without (with) a pre-trained model. The best results are bold. “*” indicates the statistically significant improvements (i.e., two-sided t-test with p < 0.05) over the best baseline.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>KuaiRand-Pure</th>
<th>KuaiRand-small</th>
<th>KuaiSAR-S</th>
<th>KuaiSAR-R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Precision</td>
<td>AUC</td>
<td>Precision</td>
<td>AUC</td>
</tr>
<tr>
<td><strong>w/o Warm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PFedRec</td>
<td>68.17</td>
<td>73.33</td>
<td>69.35</td>
<td>65.69</td>
<td>55.19</td>
</tr>
<tr>
<td>PFedNCF</td>
<td>62.99</td>
<td>72.09</td>
<td>63.73</td>
<td>62.23</td>
<td>55.31</td>
</tr>
<tr>
<td>FedRecon</td>
<td>65.21</td>
<td>61.72</td>
<td>68.81</td>
<td>65.32</td>
<td><strong>58.56</strong></td>
</tr>
<tr>
<td>PFedRec</td>
<td>59.48</td>
<td>70.80</td>
<td>61.09</td>
<td>61.33</td>
<td>56.71</td>
</tr>
<tr>
<td><strong>FedPA w/o Warm</strong></td>
<td><strong>68.44</strong></td>
<td><strong>73.75</strong></td>
<td><strong>69.65</strong></td>
<td><strong>66.69</strong></td>
<td><strong>57.58</strong></td>
</tr>
<tr>
<td><strong>w/ Warm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warm_PFedNCF</td>
<td>69.90</td>
<td>74.15</td>
<td>70.54</td>
<td>65.83</td>
<td>59.95</td>
</tr>
<tr>
<td>Warm_PFedRec</td>
<td>62.78</td>
<td>71.75</td>
<td>62.58</td>
<td>61.59</td>
<td>57.86</td>
</tr>
<tr>
<td>Warm_FedRecon</td>
<td>70.02</td>
<td>74.22</td>
<td>70.75</td>
<td>66.56</td>
<td>55.68</td>
</tr>
<tr>
<td>Warm_PFedRec</td>
<td>62.71</td>
<td>71.73</td>
<td>63.96</td>
<td>61.83</td>
<td>59.74</td>
</tr>
<tr>
<td><strong>FedPA</strong></td>
<td><strong>70.28</strong></td>
<td><strong>75.12</strong></td>
<td><strong>71.14</strong></td>
<td><strong>66.86</strong></td>
<td><strong>61.99</strong></td>
</tr>
</tbody>
</table>

Table 1: Experimental results of baselines and our method on four datasets. “w/o Warm” (“w/ Warm”) denotes training the federated recommendation system without (with) a pre-trained model. The best results are bold. “*” indicates the statistically significant improvements (i.e., two-sided t-test with p < 0.05) over the best baseline.

### 4.3 Low-Rank Personalization Analysis

Our FedPA enables the learning of low-rank personalization at both the user and user-group levels, providing comprehensive modeling of user preferences from multiple perspectives. To make an in-depth analysis of the efficacy of the two forms of personalization, we conduct two model variants: one focusing on user-level personalization and the other on user-group-level personalization. Specifically, we take FedNCF as the baseline and incorporate two forms of personalization, denoted as w/ UP (with user-level) and w/ GP (with user-group-level), respectively. Given the user grouping based on their attributes, we conduct the experiments according to multiple user attributes to make a comprehensive analysis. As shown in Table 2, incorporating either user-level or user-group-level low-rank personalization into FedNCF can improve its performance. Therefore, combining these two forms of personalization allows them to complement each other and achieve superior performance.

### 4.4 Effect of Common Knowledge on Federated Optimization

In the recommendation model, different modules have specific roles. User embedding and item embedding focus on attribute information learning, while the prediction function captures user decision patterns. To further explore the impact of the common knowledge inherent in the pre-trained model on federated optimization, we assess the specific effects of each module in the recommendation model on model performance.
Table 2: Experimental results for user-level (w/ UP) and user-group-level (w/ GP) low-rank personalization analysis on KuaiRand-Pure. Specifically, we select multiple user attributes, e.g., user active degree, to guide the user grouping in user-group-level personalization.

<table>
<thead>
<tr>
<th>Model</th>
<th>Warm_FedNCF w/ UP</th>
<th>w/ GP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td></td>
</tr>
<tr>
<td>AUC</td>
<td>69.90</td>
<td>70.01</td>
</tr>
<tr>
<td>Precision</td>
<td>74.15</td>
<td>74.26</td>
</tr>
</tbody>
</table>

Table 3: Effect of freezing different modules of the pre-trained model on federated optimization on KuaiRand-Pure. “FZ,UE”, “FZ,IE” and “FZ,PF” denote freezing user embedding, item embedding and prediction function, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>FZ,UE</th>
<th>FZ,IE</th>
<th>FZ,PF</th>
<th>FedPA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUC</td>
<td>65.55</td>
<td>70.31</td>
<td>70.28</td>
<td>70.28</td>
</tr>
<tr>
<td>Precision</td>
<td>72.67</td>
<td>75.16</td>
<td>75.27</td>
<td>75.12</td>
</tr>
</tbody>
</table>

Table 4: Experimental results of warm-starting the federated recommendation system with different-sized models by knowledge distillation on KuaiRand-Pure.

<table>
<thead>
<tr>
<th>Model</th>
<th>8-(32, 8, 1)</th>
<th>8-(8, 1)</th>
<th>4-(32, 8, 1)</th>
<th>4-(8, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>AUC</td>
<td>Precision</td>
<td>AUC</td>
</tr>
<tr>
<td>AUC</td>
<td>70.28</td>
<td>70.06</td>
<td>71.41</td>
<td>71.12</td>
</tr>
<tr>
<td>Precision</td>
<td>75.12</td>
<td>75.47</td>
<td>77.49</td>
<td>76.81</td>
</tr>
</tbody>
</table>

Table 5: Experimental results of privacy-protection enhanced FedPA with various noise intensity on KuaiRand-Pure.

<table>
<thead>
<tr>
<th>Intensity</th>
<th>0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>70.28</td>
<td>70.06</td>
<td>69.93</td>
<td>69.92</td>
<td>69.80</td>
<td>69.89</td>
</tr>
<tr>
<td>Precision</td>
<td>75.12</td>
<td>74.17</td>
<td>74.35</td>
<td>74.28</td>
<td>74.20</td>
<td>74.16</td>
</tr>
</tbody>
</table>

4.5 Lightweight FedPA with KD

In the physical setting, the service provider generally learns a large-scale pre-trained model, which poses challenges in terms of computational and storage capabilities when deployed directly to clients. To fill the gap, we develop a lightweight FedPA with knowledge distillation technique. Specifically, we first distill a small-scale model from the pre-trained large-scale model and then deploy it on each client as the base model. For a comprehensive investigation, we distill three different sizes, i.e., 8-(8, 1), 4-(32, 8, 1) and 4-(8, 1), from the original model whose size is 8-(32, 8, 1) by adjusting the embedding dimension or the prediction function architecture. As shown in Table 4, employing the distilled small-scale models for warm-starting the federated recommendation system not only preserves the model’s performance but also yields performance improvements. This finding further strengthens our method’s viability in real-world applications.
Acknowledgments

Chunxu Zhang and Bo Yang are supported by the National Key R&D Program of China under Grant No. 2021ZD0112500; the National Natural Science Foundation of China under Grant Nos. U22A2098, 62172185, 62206105 and 62202200; the Fundamental Research Funds for the Central Universities, JLU; the Key Science and Technology Development Plan of Jilin Province under Grant No. 2024032078GX; Jilin Province Capital Construction Fund Industry Technology Research and Development Project No. 2022C047-1. Yang Liu and Chunxu Zhang are supported by the Tsinghua-Kuaishou Joint Research Program.

References


