# Adaptive Personalized Federated Recommendation with Global Knowledge Distillation

Jianzhe Zhao, Lingyan He(✉), Fanzhe Lin, Jiaqi Ding, Xiaxue Zhu, and Guibing Guo

Software College, Northeastern University, Chuangxin Road 195, Shenyang, Liaoning 110169, China
zhaojz@swc.neu.edu.cn,
{2471425,20216976,20216584,20216683}@stu.neu.edu.cn, guogb@swc.neu.edu.cn

**Abstract.** Personalized federated recommendation (PFR) enhances the recommendation performance of client models by personalizing the modeling of heterogeneous data. Existing methods achieve personalization by incorporating local item embeddings on global item embedding. However, two core challenges persist. First, they lack fine-grained differentiation in data heterogeneity across datasets, leading to inconsistent performance when applying the same modeling approach. Second, solely modeling global knowledge through shared item embeddings is insufficient, reducing model performance, as client selection further exacerbates knowledge forgetting. In this work, we highlight the dataset heterogeneity and observe the impact of this heterogeneity and knowledge forgetting on model performance. We propose an Adaptive Personalized Federated Recommendation through Global Knowledge Distillation (APFR-GKD) to address these challenges. This approach introduces an adaptive personalization mechanism that adjusts the algorithm's personalization level based on the degree of data heterogeneity. Additionally, we enhance global knowledge integration by sharing item embeddings and scoring functions, incorporating global historical models, and guiding local model updates through knowledge distillation. Comprehensive experiments on four datasets demonstrate that our method consistently outperforms baseline algorithms on heterogeneous datasets. Ablation studies further validate the effectiveness of our approach.

**Keywords:** Federated recommendation · Adaptive personalization · Global knowledge distillation

## 1 Introduction

Compared to large-scale training that centralizes the data on the server, federated recommendation systems (FRs) enable lightweight recommendation services on the client side and protect the user's privacy constraints. However, data heterogeneity (non-Independent and Identically Distributed, non-IID) causes local model drift, leading to a decline in performance. The federated paradigm comprises a central server and a series of local clients. A typical federated recommendation system usually includes three modules: user embeddings, item embeddings, and a scoring function. Among them, the user embedding module is kept locally for privacy protection, and the global information is obtained through co-training of item embeddings and scoring functions at the server. In FRs [1,2,3,4], the local clients are equivalent to individual users, and data heterogeneity is often reflected in the differences in user preferences. Therefore, the client-side recommendation model in FRs needs to be adapted to the recommendation tasks of individual users. Ignoring the heterogeneity that exists when individual users rate the same item will lead to the degradation of the performance of the end-side recommendation model.

Motivated by this problem, many methods have been proposed to correct or adapt to this data heterogeneity. Some researches improve the generalization ability of global models by pulling drifting clients [5,6]. However, compared with the powerful global model, improving the performance of the end-to-side recommendation model can provide lightweight and high-quality recommendation services for individual users. Personalized federated recommendation (PFR) focuses on personalization mechanisms to capture user preferences to enhance recommendation performance [7,8]. The state-of-the-art methods include dual personalization (PFedRec) [7] and additive personalization (FedRAP) [8] mechanisms, which enhance the personalized recommendation performance of local models by integrating global item embeddings with local item embeddings. The commonality is that they share the global item embedding on the server side and the private item embeddings and score function models of users are saved locally like FedNCF [4]. The difference between the two lies in the number of updates applied to the local item embeddings using local data, resulting in varying levels of personalization. The dual personalization mechanism updates local item embeddings with the local dataset twice, but the additive personalization mechanism performs local training only once. In the dual mechanism, firstly, the locally trained item embeddings are shared and aggregated as global ones, then the global item embedding is replaced by the local item embeddings; secondly, fine-turning the local item embeddings using the local data. In the additive mechanism, the local training item embeddings are shared and aggregated globally; then, the global item embedding is integrated with the local item embeddings again via matrix addition. Therefore, because the dual mechanism makes more extensive use of local data for updates, it demonstrates a higher degree of personalization than additive personalization.

Although these methods have progressed in the personalized representation of item embeddings, two core challenges remain. First, the degree of data heterogeneity varies across different datasets (recommendation scenarios), a concept we refer to as dataset heterogeneity in this work, which yields inconsistent performance of specific personalization mechanisms on these datasets. Second, existing approaches only share local item embeddings, and client selection in federated settings can lead to knowledge forgetting, resulting in insufficient distillation of global knowledge and, consequently, a decline in model performance. To further validate our assumptions, we conduct experiments on four real datasets and compare the performance of baselines in Assumption Validation. The results show that (1) recommendation scenarios exhibit dataset heterogeneity, and a uniform personalized approach cannot be applied to heterogeneous datasets; (2) merely sharing item embeddings and selecting clients can lead to insufficient global knowledge acquisition, thereby causing a decline in performance.

Therefore, this paper introduces a novel approach, Adaptive Personalized Federated Recommendation with Global Knowledge Distillation (APFR-GKD). This approach stands out for its adaptive personalization mechanism, which adjusts the algorithm's degree of personalization to suit different recommendation scenarios. Unlike existing approaches that only upload local item embeddings to the server, we also share the parameters of the scoring function model and integrate the historical global models to distill global knowledge. This unique self-distillation technique refines global knowledge into local models, thereby enhancing the performance of client models. Specifically, the paper's contributions are as follows:

- We highlight the data heterogeneity across the datasets and design an adaptive personalization mechanism to adjust the personalized recommendation algorithm, adapting to different scenarios.
- We propose a global knowledge distillation method for PFRs. In this method, global knowledge should be better distilled by integrating local item embeddings, local score function models, and global history models on the server side. The powerful global model guides the local training via KD.
- We conduct comprehensive experiments on four real datasets, each representing a different scenario, and compare the results with benchmarks. The results demonstrate the effectiveness of our method.

## 2 Assumption Validation

In this section, we will conduct three sets of experiments to validate the following two assumptions.

**Assumption 1**: In the PFR environments, there exists a disparity in data heterogeneity across different datasets, i.e., dataset heterogeneity. The specific personalization mechanism may yield varying performance outcomes on heterogeneous datasets.

**Assumption 2**: In PFR, the model's performance is influenced by the extent to which clients share knowledge and the proportion of participating clients.

Specifically, for Assumption 1, we conduct Experiments 1 and 2, for analyzing dataset heterogeneity on the ML-100K [9], ML-1M [9], Lastfm-2K [10], and Amazon [11] datasets, and comparing the performance (HR@10 and NDCG@10) of PFedRec [7] and FedRAP [8] on these datasets. For Assumption 2, we design Experiment 3 using the FedNCF algorithm on ML-100K to observe the effects of different knowledge-sharing intensities and client participation rates on model performance, respectively. The results of the experiments, conducted with meticulous attention to detail, show that both of our assumptions hold in real applications.

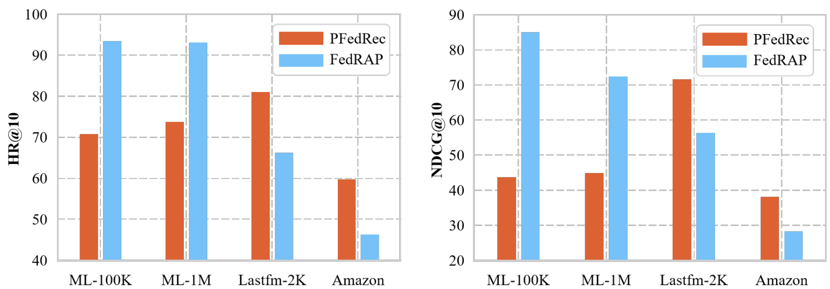
**Experiment 1**: Statistical Analysis of Data Heterogeneity. We perform a K-means clustering analysis of user preferences across four datasets and visualize the variations in preferences among user clusters (see Appendix, Fig. **??**). In general, significant differences in preferences between different user clusters within a dataset suggest significant individual user preference differences, indicating high data heterogeneity. Conversely, slight differences in preferences point to low data heterogeneity. We calculate the average Euclidean distance between user clusters (AED) and the ratio of items per user (RIU) for the four datasets, as shown in Table 1. These two indicators can reflect the heterogeneity of data to a certain extent. The results indicate that the clusters/items in ML-100K and ML-1M are relatively dense, reflecting low data heterogeneity, while in Lastfm-2K and Amazon, the clusters/items are relatively sparse, indicating high data heterogeneity. These findings highlight differences in the degree of user preference variation, i.e., the degree of data heterogeneity of the dataset, referred to as dataset heterogeneity.

Table 1: Dataset statistic, AED and RIU indicate the level of data heterogeneity.

| Dataset | User Num | Item Num | AED | RIU |
|---|---|---|---|---|
| ML-100K | 943 | 1,682 | 1.939 | 1.78 |
| ML-1M | 6,040 | 3,706 | 1.905 | 0.61 |
| Lastfm2K | 1,600 | 12,454 | 2.106 | 7.78 |
| Amazon | 8,072 | 11,830 | 2.041 | 1.46 |

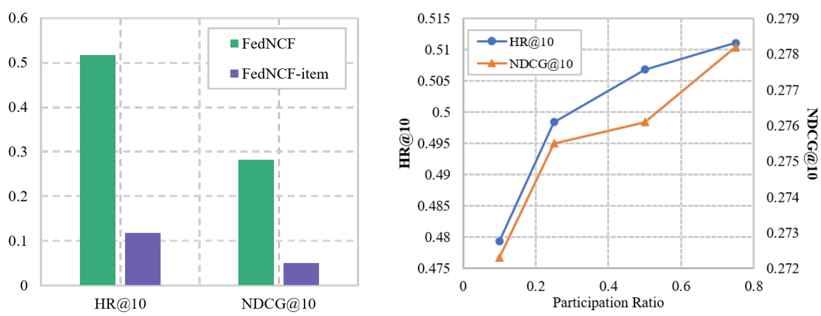
**Experiment 2**: Performance Analysis on Heterogeneous Datasets. We analyze the performance of advanced personalized mechanisms on heterogeneous datasets. As shown in Fig. 1, PFedRec, with greater personalization, outperforms others on the heterogeneous Lastfm-2K and Amazon datasets. In contrast, FedRAP excels on the less heterogeneous ML-100K and ML-1M datasets. These findings indicate that no single mechanism is universally effective.

**Experiment 3**: Impact of Knowledge Sharing Intensity and Client Participation Proportion. We analyze the effects of varying levels of knowledge sharing by uploading only item embeddings or by simultaneously uploading item embeddings and scoring functions. As depicted in Fig. 2(a), on the ML-100K dataset, FedNCF [4] achieves the highest accuracy when both item embeddings and scoring functions are shared, with sharing only item embeddings yielding the second-best accuracy. Additionally, Fig. 2(b) shows that model accuracy increases monotonically with the proportion of participating clients, indicating the significance of client participation in enhancing model performance.




(a) HR (b) NDCG

Fig. 1: Performance comparison on different datasets.




(a) (b)

Fig. 2: Model performance vs. (a) knowledge sharing intensity and (b) proportion of participating clients of FedNCF on ML-100K.

## 3 Related works

### 3.1 Personalized Federated Learning

Personalized Federated Learning (PFL) aims to train models that adapt to local clients, particularly with non-IID data. Some approaches enhance the global model's generalization for better personalization using data augmentation [12,13] and strategic client selection [14,15] to reduce data distribution heterogeneity. Other methods improve the global model through regularized local loss [16,17,18], meta-learning [19,20], and transfer learning [21]. Additionally, studies focus on customizing models for local clients, such as LG-FedAvg [22], which combines global models with local data to enhance adaptability, and customer relationship modeling [23,24,25], enabling similar clients to learn analogous models.

Knowledge distillation has also been successfully applied to PFR tasks [26,27,28]. Zhu et. al introduced FedGen, which aggregates user information data-free to regulate local training [27]. Lin et. al proposed a robust aggregation scheme using ensemble distillation for model fusion, which leverages unlabeled data or artificially generated examples to combine knowledge from heterogeneous client models [28]. The self-distillation method has proven effective for label smoothing and enhancing model generalization performance [29]. Yao et. al applied self-distillation to integrate knowledge from historical global models to guide local training, addressing the "client drift" issue [30]. However, their approach is tailored for simple classification tasks, which differs from FRs that locally maintain a recommendation model for each user.

### 3.2 Federated Recommendation Systems

In recent years, privacy protection has gained significant attention, leading to studies that integrate federated learning (FL) with recommendation systems. Most federated recommendation systems utilize matrix factorization and deep learning models. For example, FCF [1] adapted a collaborative filtering model with implicit feedback for FL. Some research combines matrix factorization with FL [2] and employs homomorphic encryption for privacy enhancement. Deep learning has also been widely used in recommendation systems to improve performance [16], and scholars have extended NCF [4] to federated settings.

Given that FRs emphasize personalization, research has focused on methods to maintain personalized models. Some studies enhance personalized recommendations using clustering methods [5,6], while others improve local model performance by combining global and local item embeddings [7,8]. This paper is more relevant to the second solution. However, these methods often overlook varying levels of user preference heterogeneity across different datasets, leading to inconsistent performance. Moreover, sharing only item embeddings between servers and clients can result in a loss of global information, which hampers recommendation performance. We address these issues by capturing global information more effectively and adapting recommendation algorithms based on dataset heterogeneity.

## 4 Problem Statement

### 4.1 Notations

The system described in this paper consists of a main server $S$ and $N$ clients $C_1, C_2, C_3, \ldots, C_N$, and each client represents a user, retains a local dataset $D_i$ for local training. Each client maintains a personalized recommendation model, which is expressed as $\theta_i = \{\theta_i^m, \theta_i^s\}$. Here, $\theta_i^m$ and $\theta_i^s$ represent the $i$-th client's item embedding and scoring function, respectively. During training, the item embeddings and score functions are exchanged between the clients and the server. The conventional FL algorithm, FedAvg [31], is applied for local model aggregation. The global model aggregated on the server is expressed as $\theta = \{\theta^m, \theta^s\}$. In our framework, client selection is used to improve the training efficiency[30,32].

In each round of training, $N$ clients are randomly selected from $n$ to participate in training. Therefore, in the $t$ round, the global model is aggregated as Equ. 1.

$$\bar{\theta} = \frac{1}{n} \sum_{i=1}^{n} \theta_i \tag{1}$$

### 4.2 Personalized Recommendation System

The training task in this paper is a PFR problem, and the training objective is to minimize the global joint loss function in Equ. 2.

$$\min_{\theta, \{\theta_i\}_{i=1}^{n}} \sum_{i=1}^{n} \alpha_i \mathcal{L}_i \left(\theta, \theta_i\right) \tag{2}$$

Here, $\mathcal{L}_i \left(\theta, \theta_i\right)$ is the training loss of the $i$-th client. $\alpha_i$ is the weight loss of the $i$-th client. In our work, we define $\alpha_i := |D_i| / \sum_{j=1}^{N} |D_j|$ by using FedAvg. After training, each client will have a personalized recommendation model.

Regarding the loss function, we combine the above setting of the federated recommendation system and define the score of the $i$-th client on the $j$-th item using its local recommendation model as $r_{ij}$. This paper focuses on the use of a user-item interaction matrix for recommendation, chooses the multi-layer perceptual model (MLP) in NCF [33] as the basic recommendation model, and discusses the implicit feedback recommendation task. When the $i$-th user interacts with the $j$-th item, its score $r_{ij} = 1$; when it does not interact, its score $r_{ij} = 0$. From this, we choose the binary cross-entropy loss (BCE-loss) function as the loss function of the local client.

$$\mathcal{L}_i \left(\theta, \theta_i; r, \hat{r}\right) = -\sum_{(i,j) \in D_i} \log \hat{r}_{ij} - \sum_{(i,j') \in D_i^-} \log \left(1 - \hat{r}_{ij'}\right) \tag{3}$$

where $D_i^-$ is the negative sample set, we remove the item that the $i$-th user interacted with from all the item lists, and extract negative samples from the remaining items according to the sampling ratio. Specifically, after each round of training, the selected client receives the global model $\theta$ sent by the server, replaces the local item embedding with the global item embedding in it, and then performs local training combined with $D_i$ to minimize the above joint loss function.

## 5 Proposed APFR-GKD

### 5.1 Overview

We present an overview of the APFR-GKD in Fig. 3 and Algorithm 1. Our primary objective is to address the challenge of personalization in PFR. One of the core issues is adaptability to the degree of data heterogeneity in PFR.

We propose an adaptive personalization mechanism that adjusts the algorithm's level of personalization based on the degree of data heterogeneity. Another core issue is how to gather local information and assemble it into a global model to guide local model training. We propose a global knowledge distillation method for PFRs to reduce the loss of global information during personalized local training.

The adaptive personalization mechanism is implemented by adding personalization adaptation components to the algorithm, the green parts in Fig. 3. First, the Personalization Adapter determines the algorithm's level of personalization based on the degree of data heterogeneity. Second, the dual training part in the green dashed line dictates whether the algorithm undergoes dual personalization updates. In APFR-GKD with Dual, a fine-tuning strategy retrains the global model using the local dataset, enhancing the algorithm's personalization. In contrast, APFR-GKD with Unary directly uses the aggregated global model as the teacher model to guide local updates, reducing the degree of personalization.

A global knowledge distillation method is introduced for local training in APFR-GKD. First, unlike existing PFR, our approach shares and integrates both the item embedding module and the scoring module into the global model, following the setup from FedNCF [4]. Second, we incorporate a self-distillation technique, leveraging historical models stored in a buffer to recover accuracy that might be lost while pursuing federated efficiency. Finally, the global model guides local model training through knowledge distillation, where the combined features from the global and local models are used to match the output of the global model, thereby enhancing the model's generalization ability.

### 5.2 Adaptive Personalization Mechanism

In the PFR, data heterogeneity manifests as differences in user preference. Therefore, we first perform $k$-means cluster analysis on user and item interaction, then measure the preference differences by the distances of user clusters. Hence, the degree of data heterogeneity $P$ could be calculated by the average Euclidean distance of $k$ cluster centers.

$$P = \frac{\sum_{i=1}^{k} \sum_{j=i+1}^{k} d(c_i, c_j)}{k(k-1)/2} \tag{4}$$

$$d(c_i, c_j) = \sqrt{\sum_{d=1}^{m} (c_{i_d} - c_{j_d})^2} \tag{5}$$

where $c_i$ and $c_j$ are the centers of the two clusters, $d(c_i, c_j)$ is Euclidean distance. $c_{i_d}$ and $c_{j_d}$ are the coordinate values in the $d$ dimension, $m$ is the number of dimensions. In APFR-GKD, we can calculate $P$ by sampling the clients' data. When sampling client data, the differential privacy strategy can be combined to avoid destroying the privacy settings. Other methods can also be performed, such as a simple analysis based on the density of users and items (RIU). When

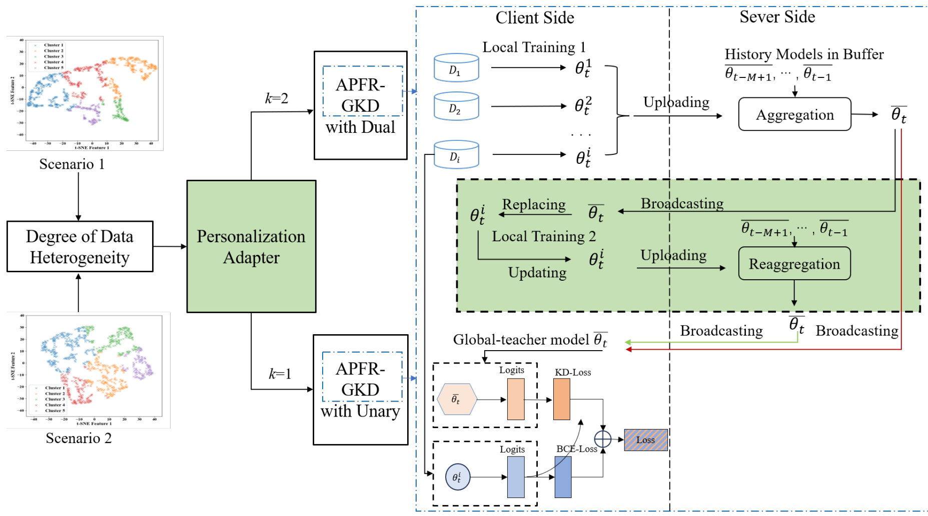



Fig. 3: Overview of APFR-GKD: It includes a Personalized Adapter to select strategies based on dataset heterogeneity, and a personalized recommendation module based on knowledge distillation.

the density of items is low, users may have fewer shared preferences and fewer preference differences.

For APFR-GKD with Dual, we implement a fine-tuning strategy with local data to enhance the personalization. We update the scoring function by replacing the local item embedding with the global item embedding from the first aggregation. Then, the scoring function is frozen, the item embeddings are updated with the local dataset again, and the resulting local model is uploaded for re-aggregation. The fine-tuning process on local data makes the feature space of the global model focus on local user features, improving the degree of personalization of the global teacher model and making it more suitable for scenarios with a significant degree of data heterogeneity. Conversely, APFR-GKD with Unary, which skips the fine-tuning module, adapts to scenarios with low data heterogeneity. The two-phase personalized adaptive mechanism enables APFR-GKD to achieve better recommendation performance in different scenarios.

### 5.3 GKD based PFR

APFR-GKD uses a client-server architecture where server-side global models guide local model updates on each client. It builds on the FedNCF framework, following the standard federated learning process of uploading local parameters, server aggregation, and broadcasting the global model for local updates. Each client maintains a Neural Collaborative Filtering (NCF) model, with the scoring function implemented via MLP. To enrich global knowledge, we share the item embeddings $\theta_i^m$ and scoring model parameters $\theta_i^s$ for global model aggregation, while the user embeddings $\theta_i^u$ are retained locally to preserve privacy. To address the issue of knowledge forgetting caused by client selection, APFR-GKD incorporates a self-distillation technique that aggregates historical global models. Specifically, in each round $t$, we aggregate $M$ historical global models, denoted as $\bar{\theta}_{t-M+1}, \cdots, \bar{\theta}_{t-1}, \bar{\theta}_t$, using a buffer of length $M$. During the historical global model aggregation process, the current global model $\bar{\theta}_t$ is replaced by a version that integrates this accumulated historical knowledge, as formalized in Equ. 6.

$$\bar{\theta}_t = \frac{1}{M} \sum_{m=1}^{M} \bar{\theta}_{t-m+1} \tag{6}$$

The knowledge distillation-based local model training only achieves global knowledge sharing by replacing or adding global item embeddings to the local model. Instead, it enhances the global teacher model to encapsulate all user feature information, effectively transferring global knowledge to the local models. This is achieved by incorporating the knowledge distillation loss (KD-loss) into the local loss (BCE-loss), thereby improving recommendation performance by aligning the global information with the local models. Assume $x$ is the input during model training, and $h(\theta, x)$ is the predicted score output by the logits layer vector $z(\theta, x)$ of model $\theta$. Here, the temperature parameter $\tau$ softens the probability distribution[34], as shown in Equ. 7. Accordingly, the loss function for local training in APFR-GKD is defined in Equ. 8.

$$h\left(\theta, x\right) = \frac{\exp\left(z\left(\theta, x\right)/\tau\right)}{\sum_d \exp\left(z\left(\theta, x\right)/\tau\right)} \tag{7}$$

$$loss = \min_{\theta, \{\theta_i\}_{i=1}^n} \sum_{i=1}^{n} \alpha_i \mathcal{L}_i\left(\theta, \theta_i\right) + \mu\tau^2 \sum_{i=1}^{n} KL\left(h\left(\bar{\theta}, x\right) \middle\| h\left(\theta_i, x\right)\right) \tag{8}$$

Here, $\sum_{i=1}^{n} KL(h(\bar{\theta}, x) \parallel h(\theta_i, x))$ denotes KD-loss which is calculated by Kullback-Leibler Divergence, while $\mathcal{L}_i\left(\theta, \theta_i\right)$ represents BCE-loss. The hyper-parameter $\mu$ controls the trade-off between the local and distillation losses.

## 6 Experiments

### 6.1 Datasets and Evaluation Index

In this study, we utilize four widely adopted real-world datasets: ML-100K, ML-1M, Lastfm-2K, and Amazon to assess the performance of the proposed APFR-GKD algorithm. To ensure a comprehensive evaluation, we compare the APFR-GKD algorithm against selected baseline algorithms using two key metrics: HR@10 and NDCG@10. HR@10 is the proportion of the top 10 recommendations of users in the test set that contain at least one ground-truth item. NDCG@10 evaluates the ranking rationality of the recommendation list by calculating the ratio of the weighted gain of the top 10 recommendations to the desired ranking gain.

**Algorithm 1** APFR-GKD

**Input**: Total number of clients $N$, total communication rounds $T$, local epoch $E$, initialized global model $\theta_0$, learning rate $\eta$, buffer size $M$, samples for personalization adapter $D_s$
**Output**: Client Models $\{\theta^1, \theta^2, \theta^3, \ldots, \theta^N\}$

**ServerExecution:**

Initialize $\theta_0$
$k = PersonalizationAdapter(D_s)$
**for** iteration $t = 1, 2, \ldots, T$ in parallel **do**
  $S_t \leftarrow ClientsRandomlySelect(N, n)$
  Send the global model $\bar{\theta}_t$ to the selected clients
  **if** $k = 2$ **then**
    $TrainingForDual(S_t, n)$
    $\bar{\theta}_t \leftarrow \frac{1}{n}\sum_{i=1}^{n}\theta_t^i$
    Save global model $\bar{\theta}_t$ to the buffer
    Ensemble the historical models in the buffer as global model $\bar{\theta}_t$
  **end if**
  $LocalTrainingWithKD(S_t, n)$
  $\bar{\theta}_t \leftarrow \frac{1}{n}\sum_{i=1}^{n}\theta_t^i$
  Save global model $\bar{\theta}_t$ to the buffer
  Ensemble the historical models in the buffer as global model $\bar{\theta}_t$
**end for**

**PersonalizationAdapter($D_s$):**

Calculate the degree of data heterogeneity $P$ via Equs. 4 and 5
**if** $P < \varepsilon$ **then**
  **return** 1
**else**
  **return** 2
**end if**

**TrainingForDual($S_t, n$):**

**for** each client index $i = 1, 2, \ldots, n$ in parallel **do**
  Download $\bar{\theta}_t$ from the server
  $\nabla\theta_t^i \leftarrow \nabla\mathcal{L}(\bar{\theta}_t, \theta_t^i \; ; r, \hat{r})$ ▷ $Update\nabla\ \theta_t^i\ via\ Equ.2$
  $\theta_t^i \leftarrow \theta_{t-1}^i - \eta\nabla\theta_t^i$
  Upload $\theta_t^i$ to the server
**end for**

**LocalTrainingWithKD($S_t, n$):**

**for** each client index $i = 1, 2, \ldots, n$ in parallel **do**
  Download $\bar{\theta}_t$ from the server
  $\nabla\theta_t^i \leftarrow \nabla\mathcal{L}(\bar{\theta}_t, \theta_t^i \; ; r, \hat{r})$ ▷ $Update\nabla\ \theta_t^i\ via\ Equ.8$
  $\theta_t^i \leftarrow \theta_{t-1}^i - \eta\nabla\theta_t^i$
  Upload $\theta_t^i$ to the server
**end for**

### 6.2 Baselines

- FedNCF [4]: This is the federated version of NCF. In this approach, each user updates their user embeddings, while item embeddings and recommendation model parameters are uploaded to the server for aggregation.
- PFedRec [7]: This approach only uploads item embeddings to the server for aggregation, while retaining user embeddings and recommendation model parameters locally. The proposed dual personalization framework is applied to enhance the recommendation performance of the local model.
- FedRAP [8]: FedRAP strikes a balance between global knowledge sharing and local personalization by integrating private local item embeddings with global item embeddings.

Table 2: Performance of HR@10 and NDCG@10 on four datasets. The table presents the experimental data of APFR-GKD and three baselines. The results are the mean value of five repeated trials.

| | ML-100K | | ML-1M | | Lastfm-2K | | Amazon | |
|---|---|---|---|---|---|---|---|---|
| | HR@10 | NDCG@10 | HR@10 | NDCG@10 | HR@10 | NDCG@10 | HR@10 | NDCG@10 |
| APFR-GKD | **0.9544** | **0.867** | 0.8978 | **0.7424** | **0.8218** | **0.7262** | **0.6067** | **0.3967** |
| FedNCF | 0.4994 | 0.2741 | 0.5076 | 0.2842 | 0.6056 | 0.5288 | 0.4467 | 0.2152 |
| PFedRec | 0.7083 | 0.4362 | 0.7372 | 0.4482 | 0.8100 | 0.7162 | 0.5978 | 0.3808 |
| FedRAP | 0.9343 | 0.8507 | **0.9310** | 0.7232 | 0.6619 | 0.5634 | 0.4628 | 0.2829 |
| Improve | 0.0200 | 0.0160 | -0.0332 | 0.0190 | 0.0120 | 0.0100 | 0.0090 | 0.0160 |

Both PFedRec and FedRAP were implemented using the code provided in the original papers. For FedNCF, since no official implementation is available, we impFLlemented it based on the original concept, excluding the differential privacy component, for comparative experiments.

### 6.3 Experimental Setting

We compare APFR-GKD with three baseline algorithms, setting the maximum client-server communications to 100 and client training iterations to 10. User and item embeddings are fixed at 32 dimensions, with other settings adjusted per the original papers. Following the methodology in [7,8], we employ leave-one-out cross-validation for dataset splitting and evaluation. Each user is paired with 99 non-interacted items, and test items are ranked among 100 samples, with four negative samples randomly selected per positive sample. In our personalized FL setup, each user corresponds to a client. For our approach, we used a single-layer MLP as the scoring function. Experiments are conducted using PyTorch, repeated five times, with final results reported as the average. All experiments are conducted on L20. and we use a 20-core CPU, Xeon(R) Platinum 8457C, with 100 GB of memory.

### 6.4 Comparison and Analysis

We conduct experiments on four datasets and compare the results with benchmark algorithms, as shown in Table 2. FedRAP demonstrates superior performance on the ML-100K and ML-1M datasets, where the degree of data heterogeneity is low. This is attributed to its penalization mechanism that operates without fine-tuning on local datasets. On the other hand, PFedRec performs better on the Lastfm-2K and Amazon datasets, which exhibit high data heterogeneity. This is because its dual personalization approach leverages local personalized information twice, enhancing its performance in such scenarios. However, due to the lack of consideration for the degree of algorithm personalization, the performance of these algorithms varies significantly across different contexts.

Table 3: Ablation study of APFR-GKD. APFR-GKD with Dual (Dual) means that the dual mechanism is used for recommendation, and APFR-GKD with Unary (Unary) means that only knowledge distillation is used.

| | ML-100K | | ML-1M | | Lastfm-2K | | Amazon | |
|---|---|---|---|---|---|---|---|---|
| | HR@10 | NDCG@10 | HR@10 | NDCG@10 | HR@10 | NDCG@10 | HR@10 | NDCG@10 |
| APFR-GKD | **0.9544** | **0.867** | **0.8978** | **0.7424** | **0.8218** | **0.7262** | **0.6067** | **0.3967** |
| Dual | 0.6542 | 0.3751 | 0.5604 | 0.3199 | 0.8218 | 0.7262 | 0.6067 | 0.3967 |
| Decrease | ↓31.45% | ↓56.74% | ↓33.17% | ↓56.91% | —— | —— | —— | —— |
| Unary | 0.9544 | 0.8670 | 0.8385 | 0.7424 | 0.6606 | 0.6190 | 0.1264 | 0.0595 |
| Decrease | —— | —— | —— | —— | ↓19.62% | ↓14.76% | ↓79.17% | ↓85.00% |

In contrast, except for the HR@10 index on the ML-1M dataset, which is slightly lower than FedRAP, our proposed algorithm outperforms all baselines across all four datasets, demonstrating consistent performance. This robustness is largely due to our adaptive personalization mechanism, which dynamically adjusts to varying levels of data heterogeneity. Additionally, our approach benefits from the introduction of global knowledge distillation, which further enhances model performance. To further validate the effectiveness of our method, we conducted ablation studies focusing on the impact of adaptive personalization mechanism and global knowledge distillation.

### 6.5 Ablation Study

To further validate and analyze the impact of the adaptive module and GKD method in APFR-GKD, we conduct specific experiments to address the following questions:

- Question 1: Why does incorporating the adaptive module lead to better personalized recommendation performance?
- Question 2: What are the advantages of the GKD method over existing algorithms on both datasets?

To address Question 1, to explore the influence of the adaptive module, we removed it from APFR-GKD and observed how recommendation performance changes without tailored adaptation to various recommendation scenarios. Specifically, we introduced two control groups: one using APFR-GKD with Dual across all datasets, and the other using APFR-GKD with Unary across all datasets. The experiments were conducted on four datasets under the same federated settings, with the results summarized in Table 3. When using APFR-GKD with Dual alone, HR@10 and NDCG@10 decreased significantly for the Lastfm-2K dataset, which has considerable data heterogeneity. The decline was even more pronounced for the Amazon dataset, where both HR@10 and NDCG@10 dropped, likely due to the high heterogeneity. Conversely, when using APFR-GKD with Unary alone, HR@10 decreased by approximately 32%, and NDCG@10 dropped by about 56% for the ML-100K and ML-1M datasets, which have low data heterogeneity. These findings indicate that neglecting the degree of data heterogeneity and either overemphasizing personalization or focusing solely on global information can adversely affect client-side recommendation performance. Our model, through the adaptive module, effectively captures user preference information by applying appropriate recommendation strategies based on the heterogeneity of the dataset.

To address Question 2, we compare APFR-GKD with baseline algorithms from Table 2. In low data heterogeneity scenarios, our approach enhances personalization by utilizing local information twice and incorporating knowledge distillation, thereby improving local model recommendations with global data. Compared to FedRAP, our method better integrates global insights, as the aggregated historical global teacher model offers richer information while knowledge distillation balances global and local aspects. Overall, APFR-GKD consistently achieves optimal recommendation performance across various datasets.

## 7 Conclusions and Future Work

This paper demonstrates the impact of dataset heterogeneity on PFR algorithms through a series of experiments. Based on these findings, we propose APFR-GKD, which uses an adaptive personalization mechanism to adjust local model personalization based on dataset heterogeneity. Integrating historical global models as teacher models through knowledge distillation further enhances performance. Experimental results show that our method outperforms baselines across four datasets. Future work will consider in-depth discussions on computational overhead, privacy protection, and scalability. At the same time, in order to achieve better recommendation results, our method is slightly more complex, and we will improve this point.

### Funding

This work was supported by the National Natural Science Foundation of China under Grant No. 62102074 and the Natural Science Foundation of Liaoning Province No. 2024-MSBA-49.

## References

1. Yi, J., Wu, F., Wu, C., Liu, R., Sun, G., Xie, X.: Efficient-fedrec: Efficient federated learning framework for privacy-preserving news recommendation. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. pp. 2814–2824. Association for Computational Linguistics (2021)
2. Chai, D., Wang, L., Chen, K., Yang, Q.: Secure federated matrix factorization. IEEE Intell. Syst. **36**(5), 11–20 (2021)

3. Wu, C., Wu, F., Cao, Y., Huang, Y., Xie, X.: A federated graph neural network framework for privacy-preserving personalization. Nature Communications **13** (2021)
4. Perifanis, V., Efraimidis, P.S.: Federated neural collaborative filtering. Knowl. Based Syst. **242**, 108441 (2022)
5. He, X., Liu, S., Keung, J., He, J.: Co-clustering for federated recommender system. In: Chua, T., Ngo, C., Kumar, R., Lauw, H.W., Lee, R.K. (eds.) Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024. pp. 3821–3832. ACM (2024)
6. Zhang, H., Li, H., Chen, J., Cui, S., Yan, K., Wuerkaixi, A., Zhou, X., Shen, Z., Li, Y.: Beyond similarity: Personalized federated recommendation with composite aggregation. CoRR **abs/2406.03933** (2024)
7. Zhang, C., Long, G., Zhou, T., Yan, P., Zhang, Z., Zhang, C., Yang, B.: Dual personalization on federated recommendation. In: Proceedings of IJCAI 2023, 19th-25th August 2023, Macao, SAR, China. pp. 4558–4566. ijcai.org (2023)
8. Li, Z., Long, G., Zhou, T.: Federated recommendation with additive personalization. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net (2024)
9. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2016)
10. Cantador, I., Brusilovsky, P., Kuflik, T.: Second workshop on information heterogeneity and fusion in recommendersystems (hetrec2011). In: Mobasher, B., Burke, R.D., Jannach, D., Adomavicius, G. (eds.) Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys2011, Chicago, IL, USA, October 23-27, 2011. pp. 387–388. ACM (2011)
11. Ni, J., Li, J., McAuley, J.: Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on EMNLP-IJCNLP. pp. 188–197. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
12. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. CoRR **abs/1806.00582** (2018)
13. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.: Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. CoRR **abs/1811.11479** (2018)
14. Yang, M., Wang, X., Zhu, H., Wang, H., Qian, H.: Federated learning with class imbalance reduction. In: 29th EUSIPCO 2021, Dublin, Ireland, August 23-27, 2021. pp. 2174–2178. IEEE (2021)
15. Li, L., Duan, M., Liu, D., Zhang, Y., Ren, A., Chen, X., Tan, Y., Wang, C.: Fedsae: A novel self-adaptive federated learning framework in heterogeneous systems. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–10. IEEE (2021)
16. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Dhillon, I.S., Papailiopoulos, D.S., Sze, V. (eds.) Proceedings of the Third Conference on Machine Learning and Systems, MLSys 2020, Austin, TX, USA, March 2-4, 2020. mlsys.org (2020)
17. Kirkpatrick, J., Pascanu, R., Rabinowitz, N.C., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. CoRR **abs/1612.00796** (2016)

18. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: SCAFFOLD: stochastic controlled averaging for federated learning. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 5132–5143. PMLR (2020)
19. Dinh, C.T., Tran, N.H., Nguyen, T.D.: Personalized federated learning with moreau envelopes. In: Advances in Neural Information Processing Systems 33, NeurIPS 2020, December 6-12, 2020, virtual (2020)
20. Khodak, M., Balcan, M., Talwalkar, A.: Adaptive gradient-based meta-learning methods. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32,NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 5915–5926 (2019)
21. Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. CoRR **abs/1910.03581** (2019)
22. Liang, P.P., Liu, T., Liu, Z., Salakhutdinov, R., Morency, L.: Think locally, act globally: Federated learning with local and global representations. CoRR **abs/2001.01523** (2020)
23. Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., Zhang, Y.: Personalized cross-silo federated learning on non-iid data. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021. pp. 7865–7873. AAAI Press (2021)
24. Mansour, Y., Mohri, M., Ro, J., Suresh, A.T.: Three approaches for personalization with applications to federated learning. CoRR **abs/2002.10619** (2020)
25. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. IEEE Trans. Inf. Theory **68**(12), 8076–8091 (2022)
26. Seo, H., Park, J., Oh, S., Bennis, M., Kim, S.: Federated knowledge distillation. CoRR **abs/2011.02367** (2020)
27. Zhu, Z., Hong, J., Zhou, J.: Data-free knowledge distillation for heterogeneous federated learning. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 12878–12889. PMLR (2021)
28. Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33, NeurIPS 2020, December 6-12, 2020, virtual (2020)
29. Kim, K., Ji, B., Yoon, D., Hwang, S.: Self-knowledge distillation with progressive refinement of targets. In: 2021 IEEE/CVF ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 6547–6556. IEEE (2021)
30. Yao, D., Pan, W., Dai, Y., Wan, Y., Ding, X., Yu, C., Jin, H., Xu, Z., Sun, L.: Fedgkd: Toward heterogeneous federated learning via global knowledge distillation. IEEE Trans. Computers **73**(1), 3–17 (2024)
31. McMahan, H.B., Moore, E., Ramage, D., y Arcas, B.A.: Federated learning of deep networks using model averaging. CoRR **abs/1602.05629** (2016)
32. Fu, L., Zhang, H., Gao, G., Zhang, M., Liu, X.: Client selection in federated learning: Principles, challenges, and opportunities. IEEE Internet of Things Journal **10**(24), 21811–21819 (2023)
33. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: Barrett, R., Cummings, R., Agichtein, E., Gabrilovich, E. (eds.) Proceedings of the 26th WWW 2017, Perth, Australia, April 3-7, 2017. pp. 173–182. ACM (2017)
34. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015)