Breaking the Aggregation Bottleneck in Federated Recommendation: A Personalized Model Merging Approach

Jundong Chen[1,2], Honglei Zhang[1,2], Chunxu Zhang[3], Fangyuan Luo[4], Yidong Li[1,2*]

[1]Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, China
[2]Beijing Jiaotong University, China
[3]Jilin University, China
[4]Beijing University of Technology, China
{jundongchen, honglei.zhang, ydli}@bjtu.edu.cn, cxzhang19@mails.jlu.edu.cn, luofangyuan@bjut.edu.cn

Abstract

Federated recommendation (FR) facilitates collaborative training by aggregating local models from massive devices, enabling client-specific personalization while ensuring privacy. However, we empirically and theoretically demonstrate that server-side aggregation can undermine client-side personalization, leading to suboptimal performance, *i.e.*, the aggregation bottleneck. This issue stems from the inherent heterogeneity across numerous clients in FR, which drives the global model to deviate from local optima. To this end, we propose FedEM, which elastically merges the global and local models to compensate for impaired personalization. Unlike existing personalized federated recommendation (pFR) methods, FedEM (1) investigates the aggregation bottleneck in FR through theoretical insights, rather than relying on heuristic analysis; (2) leverages off-the-shelf local models rather than designing additional mechanisms to boost personalization. Extensive experiments demonstrate that our method preserves client personalization during collaborative training, outperforming state-of-the-art baselines.

Introduction

Federated recommendation (FR), as an emerging on-device learning paradigm, ensures that clients' raw data remains local during the training process, thus protecting user privacy (Sun et al. 2024; Yin et al. 2024). Pioneering works include FedMF (Chai et al. 2020) and FedNCF (Perifanis and Efraimidis 2022), which apply matrix factorization and neural collaborative filtering, respectively, within the federated framework. Conceptually, the clients upload locally trained models for global aggregation and then download the aggregated model for the next round (Chen et al. 2023).

However, due to the varying user preferences in FR, the interaction data from different clients are not independent and identically distributed (Non-IID), which naturally leads to data heterogeneity (Sun et al. 2024). Traditional FR methods fail to address this issue by sharing one same global model across all clients (Zhang et al. 2023b). Hence, personalized federated recommendation (pFR) has been introduced to tailor client-specific models. For example, PFedRec (Zhang et al. 2023a) enables dual personalization of both local and global components, while FedRAP (Li, Long, and Zhou 2024) trains an extra personalized model locally. Although effective in practice, existing pFR methods suffer from two key limitations: (1) They overlook the degradation of local personalization caused by global aggregation in FR; (2) They rely on heuristically designed personalization mechanisms, which limit their compatibility.

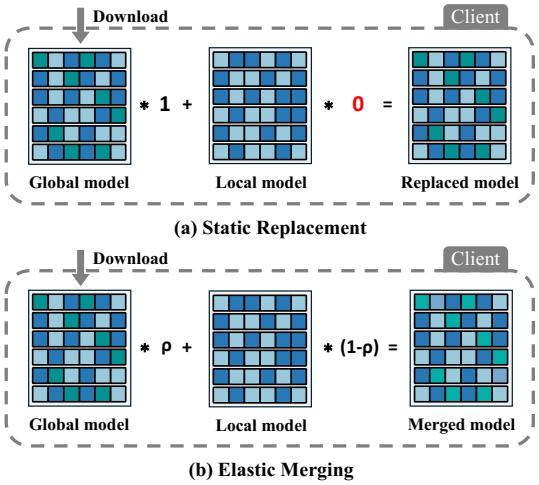

Figure 1: Traditional FR methods directly replace the local model with the global model, while our FedEM can elastically merge both the global and local models with the weight ρ, delicately balancing collaboration and personalization.

For the first limitation, we theoretically reveal that aggregation can cause a loss of local information in the global model. This issue is exacerbated by the unique nature of FR tasks, which often involve millions of clients, several orders of magnitude more than typical federated learning scenarios. Besides, existing FR methods typically adopt a `static replacement` scheme, as illustrated in Figure 1(a), where the global model replaces the local one for both training and inference. Such a scheme propagates the impact of aggregation, causing optimization to deviate from the client-specific optimum and ultimately undermining local personalization. We refer to this issue as the aggregation bottleneck.

*Corresponding author (ydli@bjtu.edu.cn).

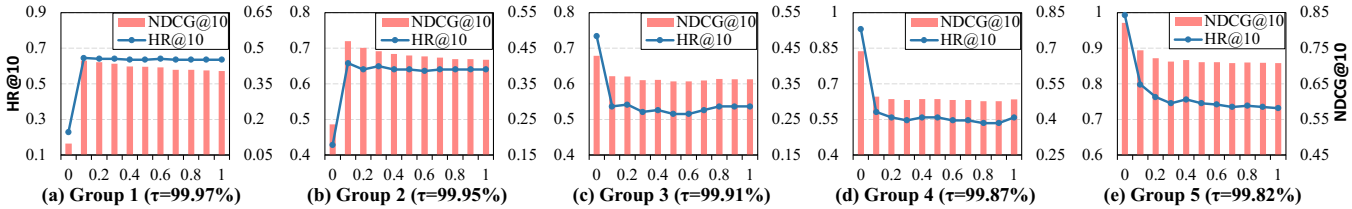

Figure 2: The performance of different client groups under varying merging weights ρ. The degree of data sparsity τ decreases from group 1 to group 5. Here, $\rho = 1$ stands for the traditional static replacement scheme.

For the second limitation, we explore a simpler and more general solution. Grounded in model merging theory (Zhou et al. 2024; Yang et al. 2024), we aim to compensate for local information loss with the off-the-shelf local model, thereby breaking the aggregation bottleneck. Specifically, we adopt an `elastic merging` scheme, as illustrated in Figure 1(b), where ρ denotes the weight of the global model parameters, while $1 - \rho$ stands for the weight of the local ones. We conduct empirical validation based on FedMF. To account for client heterogeneity, we divide the clients into five groups according to privacy-insensitive statistics of their local data, *e.g.*, data sparsity. For each group, we report the average performance metrics (HR@10 and NDCG@10) under different ρ. As shown in Figure 2, we observe that: (1) Static replacement scheme, *i.e.*, $\rho = 1$, yields suboptimal performance, which is consistent with our theoretical analysis; (2) Elastic merging scheme effectively enhances model performance. Owing to heterogeneity across clients, each group achieves optimal trade-offs under different merging weights ρ. Given this, we adopt client-specific merging that elastically adjusts to local personalization needs.

Taking both limitations into account, we propose a simple yet theoretically guaranteed pFR method called **Fed**erated recommendation via **E**lastic **M**erging (**FedEM**). It merges the aggregated global model with the off-the-shelf local model in a balanced way, effectively absorbing collaborative information while preserving client-specific personalization. To sum up, our main contributions are as follows:

- To the best of our knowledge, we are the first to theoretically analyze the aggregation bottleneck in FR scenarios, *i.e.*, the loss of local information caused by global aggregation, which further harms local personalization.
- Based on model merging, we propose FedEM to bridge the gap between global collaboration and local personalization. Unlike existing heuristic methods, our approach is theoretically grounded and empirically validated.
- The proposed elastic merging module is model-agnostic and can be seamlessly integrated as a plug-in to enhance FR/pFR methods constrained by aggregation bottleneck.
- Extensive experiments demonstrate that FedEM consistently outperforms state-of-the-art (SOTA) methods.

Related Work

Personalized Federated Learning

Federated learning (FL) is a distributed learning paradigm to mitigate privacy issues (Feng et al. 2018; Huang et al. 2024; Feng et al. 2025). Traditional FL methods, such as FedAvg (McMahan et al. 2017), struggle to derive a global model generalized for each client when the local data is Non-IID (Huang et al. 2021). To that end, some personalized federated learning (pFL) methods aim to fine-tune the globally aggregated model for each client to obtain the personalized ones (Collins et al. 2021; Fallah, Mokhtari, and Ozdaglar 2020). For instance, FedALA exploits the general information of global model to enhance the capability for local models targeting federated vision tasks (Zhang et al. 2023c). Still other methods tend to locally learn a personalized model for each client (Li et al. 2021), *e.g.*, pFedMe (T Dinh, Tran, and Nguyen 2020) uses Moreau envelopes as the client regulation loss to decouple personalized model optimization from the global model learning. Other methods take a global view to boost personalization (Ji et al. 2019; Zhang et al. 2021; Zheng et al. 2025), *e.g.*, pFedGraph (Ye et al. 2023) optimizes the collaboration graph to perform weighted aggregation for different clients.

Personalized Federated Recommendation

In federated recommendation (FR), there is natural heterogeneity among clients due to their different preferences, thus necessitating personalization of the local model (Yin et al. 2024). Similar to pFL, personalized federated recommendation (pFR) methods can also be categorized into three research lines. (1) Fine-tune the global model (Zhang et al. 2026): PFedRec (Zhang et al. 2023a) personalizes both the score function and global model to alleviate data heterogeneity. FedCIA (Han et al. 2025) leverages the aggregated item similarity matrix to guide local model training, aiming to achieve global fine-tuning. (2) Learn an additional personalized model (Chen et al. 2025): FedRAP (Li, Long, and Zhou 2024) applies an additive model to item embeddings to enhance personalization. (3) Perform personalized global aggregation (Luo, Xiao, and Song 2022; Zhang et al. 2024b): FedFast (Muhammad et al. 2020) performs global aggregation by identifying representatives from different clusters based on user profile similarities. GPFedRec (Zhang et al. 2024a) performs graph-guided aggregation to recover inter-client correlations, generating client-specific global models. Unlike the above methods, which follow the traditional static replacement scheme and rely on heuristic personalization mechanisms, FedEM adopts an elastic merging scheme, offering a simple yet theoretically grounded solution. During global collaborative training, it leverages the off-the-shelf local model to enhance personalization for each client.

Preliminaries

Let $\mathcal{U}$ denote the set with n users/clients, $\mathcal{I}$ the set with m items. Then $\mathcal{D}_u = \{(u, i, r_{ui} | i \in \mathcal{I}_u)\}$ denotes the local interaction dataset of client u, where $\mathcal{I}_u$ is for the items observed by client u, and each entry $r_{ui} \in \{0, 1\}$ is for the label on item i by client u. The goal of FR is to predict $\hat{r}_{ui}$ of client u for each unobserved item $i \in \mathcal{I} \setminus \mathcal{I}_u$ on local devices. Formally, the global optimization objective over n clients of FR tasks is:

$$\min_{(\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n; \mathbf{Q}_1^l, \mathbf{Q}_2^l, \cdots, \mathbf{Q}_n^l)} \sum_{u=1}^{n} p_u \mathcal{L}_u \left(\mathbf{p}_u, \mathbf{Q}_u^l; \mathcal{D}_u\right), \quad (1)$$

where $\mathbf{p}_u \in \mathbb{R}^d$ and $\mathbf{Q}_u^l \in \mathbb{R}^{m \times d}$ denote local user embedding and local item embedding table for client u, respectively, d is the dimension of the embedding vector. In this work, we treat item embedding table $\mathbf{Q}$ as the intermediate model parameters for uploads and downloads, since it is the standard configuration in embedding-based FRs (Ammad-Ud-Din et al. 2019). The server aggregates local parameters $\mathbf{Q}_u^l$ with weight p_u to derive global model $\mathbf{Q}^g$, *e.g.*, $p_u = |\mathcal{D}_u| / \sum_{v=1}^{n} |\mathcal{D}_v|$ in FedAvg (McMahan et al. 2017) and $p_u = 1/n$ in FCF (Ammad-Ud-Din et al. 2019). $\mathcal{L}_u$ is the task-specific objective to facilitate local training.

In this work, we focus on the typical recommendation task with implicit feedback, where the rating r_{ui} of item i by user u is either 1 or 0, indicating interested versus uninterested interaction, respectively. Therefore, we adopt the binary cross-entropy (BCE) loss (He et al. 2017), which is commonly used in binary-value problems, as the local objective $\mathcal{L}_u$. Formally, the BCE loss is defined as:

$$\begin{aligned} \mathcal{L}_u(\mathbf{p}_u, \mathbf{Q}_u^l; \mathcal{D}_u) = - \sum_{(u,i,r_{ui}) \in \mathcal{D}_u} \big[& r_{ui} \log \hat{r}_{ui} \\ & + (1 - r_{ui}) \log (1 - \hat{r}_{ui}) \big], \end{aligned} \quad (2)$$

where $\hat{r}_{ui} = \phi(\mathbf{p}_u \mathbf{q}_i^\top)$ denotes the predicted rating of item i by user u, where $\mathbf{q}_i \in \mathbf{Q}_u^l$ and $\phi(\cdot)$ is the Sigmoid function.

Methodology

Motivation

Aggregation Bottleneck. Most existing FR methods adopt fixed-weight aggregation on the server side to enable collaborative training across clients. Taking FCF (Ammad-Ud-Din et al. 2019) for example, that is, $p_u = 1/n$, the global model derived by aggregation in round t can be represented as:

$$\begin{aligned} \mathbf{Q}^{g(t)} = \frac{1}{n} \sum_{u=1}^{n} \mathbf{Q}_u^{l(t)} = \frac{1}{n} \mathbf{Q}_u^{l(t)} + \frac{n-1}{n} \mathbf{Q}_r^{g(t)}, \\ \text{where} \quad \mathbf{Q}_r^{g(t)} = \frac{1}{n-1} \sum_{v \neq u}^{n} \mathbf{Q}_v^{l(t)}. \end{aligned} \quad (3)$$

Here, $\mathbf{Q}_r^{g(t)}$ denotes the aggregated model over the remaining clients except for ego Client u. However, unlike typical FL settings, FR tasks often involve tens of thousands or even millions of clients. Therefore, we have:

$$\lim_{n \to \infty} \frac{1}{n} = 0, \ \lim_{n \to \infty} \frac{n-1}{n} = 1, \quad (4)$$

which means $\mathbf{Q}^{g(t)} \approx \mathbf{Q}_r^{g(t)}$, indicating a loss of local information in the global model. At this point, for client u, adopting the static replacement scheme may lead to the aggregation bottleneck formally established in Lemma 1.

Assumption 1 (Smoothness and Convexity) *The local objective $\mathcal{L}_u(\mathbf{Q}_u)$ is assumed to be continuously differentiable and L-smooth. Moreover, it satisfies an approximate μ-strong convexity condition induced by ℓ_2 regularization in the BCE loss (Boyd and Vandenberghe 2004; Bottou, Curtis, and Nocedal 2018). Such surrogate assumptions are standard in theoretical analyses of federated optimization (McMahan et al. 2017; T Dinh, Tran, and Nguyen 2020) and can still provide qualitative insights into the optimization direction under locally non-convex objectives.*

Lemma 1 (Bottleneck of Global Aggregation) *Based on Assumption 1, let $\mathbf{Q}_u^*$ denote the local optimal model. Then, following the static replacement scheme, i.e., using the global model $\mathbf{Q}^{g(t)}$ for local training, may lead to:*

$$\langle \nabla \mathcal{L}_u(\mathbf{Q}^{g(t)}), \mathbf{Q}^{g(t)} - \mathbf{Q}_u^* \rangle \leq 0, \quad (5)$$

which shows that the globally aggregated model can cause training optimization to deviate from the client-specific optimum, thereby undermining local personalization. This issue becomes more pronounced under the high inherent client heterogeneity in FR, leading to a performance bottleneck.

Framework

As illustrated in Figure 3, the overall framework of FedEM consists of the following main steps in each round t:
1) Elastic Merging: client u merges aggregated global model $\mathbf{Q}_u^{g(t-1)}$ and off-the-shelf local model $\mathbf{Q}_u^{l(t-1)}$ to derive a merged model $\mathbf{Q}_u^{m(t)}$ at the beginning of this round t. Here, existing FR methods follow the static replacement scheme and directly discard the local model.
2) Local Training: client u utilizes its own data $\mathcal{D}_u$ to update the merged model $\mathbf{Q}_u^{m(t)}$ and local parameters like $\mathbf{p}_u^{(t-1)}$. Then, client u uploads the updated model $\mathbf{Q}_u^{l(t)}$ to the server.
3) Global Aggregation: the server performs aggregation based on the similarity over uploaded $\{\mathbf{Q}_1^{l(t)}, \cdots, \mathbf{Q}_n^{l(t)}\}$, generating the global model $\mathbf{Q}_u^{g(t)}$ for client u.
Notably, both $\mathbf{Q}_u^{g(0)}$ and $\mathbf{Q}_u^{l(0)}$ represent randomly initialized models when $t = 1$. After T rounds above, each client can get its own personalized model.

Elastic Merging

Theoretical Analysis. Existing pFR methods, *e.g.*, PFedRec (Zhang et al. 2023a), FedRAP (Li, Long, and Zhou 2024), and FedCIA (Han et al. 2025), heuristically design additional personalization mechanisms. Although empirically effective, these methods fail to identify the aggregation bottleneck as the underlying cause of insufficient personalization in FR, thereby limiting their potential for further improvement. Additionally, their complexity and limited compatibility pose challenges for practical deployment.

Inspired by the model merging theory (Zhou et al. 2024; Yang et al. 2024), we merge the global model $\mathbf{Q}^{g(t)}$ and the

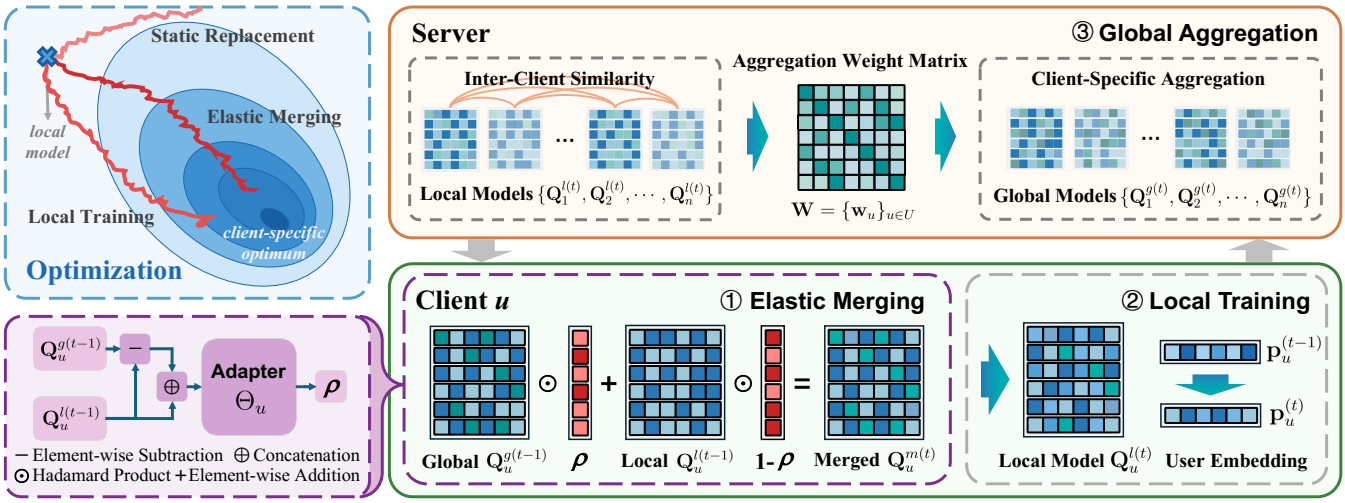

Figure 3: The framework of FedEM. Unlike the static replacement scheme in existing FR methods, we propose an elastic merging scheme to mitigate the optimization deviation caused by global aggregation, effectively breaking the performance bottleneck. Additionally, the server can perform similarity-based aggregation to further alleviate the aggregation bottleneck.

off-the-shelf local model $\mathbf{Q}_u^{l(t)}$ to overcome the aggregation bottleneck. For embedding-based FRs, $\forall \rho \in [0,1]$,

$$f(\rho \mathbf{Q}^{g(t)} + (1-\rho)\mathbf{Q}_u^{l(t)}) = \rho f(\mathbf{Q}^{g(t)}) + (1-\rho) f(\mathbf{Q}_u^{l(t)}), \tag{6}$$

where $f(\mathbf{Q})$ denotes the predicted rating, *i.e.*, $\mathbf{p}_u \cdot \mathbf{Q}$. Thus, the two models $\mathbf{Q}^{g(t)}$ and $\mathbf{Q}_u^{l(t)}$ satisfy the model merging conditions in the parameter space (Zhou et al. 2024). We define the merged model as:

$$\mathbf{Q}_u^{m(t+1)} = \rho \mathbf{Q}^{g(t)} + (1-\rho)\mathbf{Q}_u^{l(t)}, \tag{7}$$

which is used as the initialization for local training in round $t+1$. By following the elastic merging scheme, we compensate for the loss of local information in the aggregated model. The merged model benefits from global collaborative training while mitigating the risk of deviating from the client-specific optimum, thus preserving personalization.

Lemma 2 (Compensation of Elastic Merging) *Under Assumption 1, the merged model can compensate for the optimization deviation caused by aggregation, guiding the update toward the client-specific optimum, i.e., it satisfies:*

$$\begin{aligned} &\langle \nabla \mathcal{L}_u(\mathbf{Q}_u^{m(t+1)}),\ \mathbf{Q}_u^{m(t+1)} - \mathbf{Q}_u^{*} \rangle \\ &\geq (1-\rho)\langle \nabla \mathcal{L}_u(\mathbf{Q}_u^{l(t)}),\ \mathbf{Q}_u^{l(t)} - \mathbf{Q}_u^{*} \rangle - \rho C, \end{aligned} \tag{8}$$

where C denotes the deviation introduced by global aggregation. This deviation can be lower-bounded by both the state of the local model $\mathbf{Q}_u^{l(t)}$ and the global-local discrepancy $\mathbf{Q}_u^{\Delta(t)} = \mathbf{Q}^{g(t)} - \mathbf{Q}_u^{l(t)}$, i.e., it is approximately $\|\nabla \mathcal{L}_u(\mathbf{Q}_u^{l})\|(\|\mathbf{Q}_u^{l} - \mathbf{Q}_u^{*}\| + \|\mathbf{Q}_u^{\Delta}\|)$. The first term on the right-hand side of the inequality, $A = \langle \nabla \mathcal{L}_u(\mathbf{Q}_u^{l(t)}),\ \mathbf{Q}_u^{l(t)} - \mathbf{Q}_u^{*} \rangle > 0$, reflects the preservation of client-specific optimization signal. For effective local personalization, the update must remain aligned with the client-specific optimum, which requires the left-hand side of the inequality to be positive. To achieve this, we need to adaptively control the merging weight $0 \leq \rho < \frac{A}{A+C}$ based on the deviation C in each round. Accordingly, we propose the Elastic Merging module, which learns ρ from $\mathbf{Q}_u^{l(t)}$ and $\mathbf{Q}_u^{\Delta(t)}$, ensuring client-wise alignment and mitigating the aggregation bottleneck.

Practical Implementation. On the client side, we introduce the Elastic Merging (EM) module to dynamically preserve personalized information from the local model during global collaborative training. Furthermore, we utilize the vector $\boldsymbol{\rho} \in \mathbb{R}^m$ for fine-grained merging at the item level. Based on the analysis in Lemma 2, we first compute the global-local model discrepancy $\mathbf{Q}_u^{\Delta(t-1)} = \mathbf{Q}_u^{g(t-1)} - \mathbf{Q}_u^{l(t-1)}$, and then concatenate it with the local model $\mathbf{Q}_u^{l(t-1)}$ to obtain $\mathbf{Q}_u^{\text{cat}(t-1)} = \mathbf{Q}_u^{\Delta(t-1)} \oplus \mathbf{Q}_u^{l(t-1)}$. We feed $\mathbf{Q}_u^{\text{cat}(t-1)} \in \mathbb{R}^{m \times 2d}$ into the core component of the EM module, *i.e.*, the adapter, to generate the elastic vector $\boldsymbol{\rho}$ for each client u.

Here, we implement the adapter with an L-layer multilayer perceptron (MLP). Formally, we have:

$$\boldsymbol{\rho} = \phi_{output}(\Theta_u^L \bullet (\cdots \phi(\Theta_u^2 \bullet (\phi(\Theta_u^1 \bullet \mathbf{Q}_u^{\text{cat}(t-1)}))))), \tag{9}$$

where $\phi(\cdot)$ and $\phi_{output}(\cdot)$ denote the mapping functions for the intermediate and output layers, respectively. Here, we employ ReLU for ϕ and Sigmoid for ϕ_{output} to ensure that each element of vector $\boldsymbol{\rho}$ is in the range $[0,1]$. Besides, notation $\bullet$ denotes the matrix product operation, and Θ_u^i denotes the parameters of the i-th layer. Then, the merged model can be formulated as:

$$\begin{aligned} \mathbf{Q}_u^{m(t+1)} &= \boldsymbol{\rho} \odot \mathbf{Q}_u^{g(t-1)} + (\mathbf{1} - \boldsymbol{\rho}) \odot \mathbf{Q}_u^{l(t-1)} \\ &= \mathbf{Q}_u^{l(t-1)} + \boldsymbol{\rho} \odot \mathbf{Q}_u^{\Delta(t-1)}, \end{aligned} \tag{10}$$

where $\odot$ denotes the Hadamard product. Hence, by preserving the local information in $\mathbf{Q}_u^{l(t)}$, $\boldsymbol{\rho}$ serves to selectively absorb the collaborative information in $\mathbf{Q}_u^{\Delta(t)}$, thus preventing the aforementioned aggregation bottleneck.

Specifically, the adapter parameters Θ_u can be updated using stochastic gradient descent (SGD) as follows:

$$\Theta_u = \Theta_u - \beta \cdot \frac{\partial \mathcal{L}_u(\mathbf{p}_u^{(t-1)}, \mathbf{Q}_u^{m(t)}; \mathcal{D}_u)}{\partial \Theta_u}, \quad (11)$$

where β is the learning rate for the adapter. Notably, only the parameters Θ_u are updated with $\boldsymbol{\rho}$ varied in this step, while other model parameters, *i.e.*, $\mathbf{p}_u^{(t-1)}$, $\mathbf{Q}_u^{l(t-1)}$ and $\mathbf{Q}_u^{g(t-1)}$, are frozen. Unlike other FR methods, $\mathbf{Q}_u^{m(t)}$ is utilized to initialize the local model for local training rather than $\mathbf{Q}_u^{g(t-1)}$.

Local Training

The objective of local training is to update the user embedding $\mathbf{p}_u^{(t-1)}$ and the item embedding table $\mathbf{Q}_u^{m(t)}$ to obtain $\mathbf{p}_u^{(t)}$ and $\mathbf{Q}_u^{l(t)}$. Note that the adapter Θ_u is not involved during this process. Formally, the update can be written as:

$$\mathbf{p}_u = \mathbf{p}_u - \eta \cdot \frac{\partial \mathcal{L}_u}{\partial \mathbf{p}_u}, \quad \mathbf{Q}_u^l = \mathbf{Q}_u^l - \eta \cdot \frac{\partial \mathcal{L}_u}{\partial \mathbf{Q}_u^l}, \quad (12)$$

where η is the learning rate for local training. To protect privacy, client u only uploads the trained $\mathbf{Q}_u^{l(t)}$ for aggregation.

Global Aggregation

As shown in Lemma 1, traditional fixed-weight aggregation is ineffective in FR, and the aggregation bottleneck is further exacerbated by inherent client heterogeneity. In addition to local elastic merging, we further perform global similarity-based aggregation to alleviate the impact of such heterogeneity. Inspired by the previous work (Ye et al. 2023) in FL, we additionally incorporate client similarity, generating a client-specific weight vector $\mathbf{w}_u = [w_{u1}, w_{u2}, \ldots, w_{un}] \in \mathbb{R}^n$ to conduct tailored aggregation for each client u. Specifically, $\mathbf{w}_u$ is obtained by minimizing the following objective:

$$\mathcal{L}_g = \sum_{v=1}^{n}((w_{uv} - p_v)^2 + \alpha(w_{uv} - \sigma(\mathbf{Q}_u^l, \mathbf{Q}_v^l))^2), \quad \text{s.t.} \quad \mathbf{1}^T\mathbf{w}_u = 1 \text{ and } \mathbf{w}_u > \mathbf{0}, \quad (13)$$

where $p_u = |\mathcal{D}_u| / \sum_{v=1}^{n} |\mathcal{D}_v|$ denotes the aggregation weight used in the backbone FedMF (Chai et al. 2020). Here, $\sigma(\mathbf{Q}_u^l, \mathbf{Q}_v^l) = 1/(1 + \|\mathbf{Q}_u^l - \mathbf{Q}_v^l\|^2)$ denotes the similarity function. This term ensures that the aggregation weight w_{uv} increases when the two client models are highly similar. Besides, α is a similarity-related hyperparameter. After global aggregation, each client u can download the global model $\mathbf{Q}_u^{g(t)} = \sum_{v=1}^{n} w_{uv}\mathbf{Q}_v^{l(t)}$ to start the next round $t+1$.

Discussions

Complexity Analysis

Given n clients and m items, with embedding dimension d, the computational complexity of the local backbone model is $\mathcal{O}((m+1)d)$. Our proposed EM module is implemented as an L-layer MLP, whose complexity is $\mathcal{O}(Ld^2)$. Since $m \gg d > L$ in practice, the EM module introduces negligible overhead to the local device, making it well-suited for on-device recommendation.

Privacy Analysis

FedEM naturally preserves user privacy under the federated paradigm. Moreover, the local merging parameters Θ_u and $\boldsymbol{\rho}$ are not shared with the server, reducing the risk of privacy leakage (Chai et al. 2020). To further enhance privacy, we incorporate local differential privacy (LDP) (Qi, Wang, and Huang 2024) into our method. The privacy budget ε is guaranteed by $\mathcal{S}_u/\delta$, where δ is the noise strength and $\mathcal{S}_u$ denotes the global sensitivity of client u. We upper-bound $\mathcal{S}_u$ by $2p_u\eta Z$, where Z is the gradient clipping threshold.

Experiments

Experimental Settings

Datasets. We evaluate our proposed method on four datasets with varying client scale and data sparsity: Filmtrust (Guo, Zhang, and Yorke-Smith 2013), ML-100K (Harper and Konstan 2015), ML-1M (Harper and Konstan 2015), and LastFM-2K (Cantador, Brusilovsky, and Kuflik 2011).

Evaluation Protocols. We follow the popular *leave-one-out* evaluation (Bayer et al. 2017; He et al. 2017), and report the performance by *Hit Ratio* (HR@10) and *Normalized Discounted Cumulative Gain* (NDCG@10) (He et al. 2015).

Compared Baselines. We compare FedEM with SOTA centralized methods, *e.g.*, MF (Koren, Bell, and Volinsky 2009), NCF (He et al. 2017), LightGCN (He et al. 2020), and federated methods, *e.g.*, FedMF (Chai et al. 2020), FedNCF (Perifanis and Efraimidis 2022), FedFast (Muhammad et al. 2020), PFedRec (Zhang et al. 2023a), CoLR (Nguyen et al. 2024), GPFedRec (Zhang et al. 2024a), FedRAP (Li, Long, and Zhou 2024), FedCIA (Han et al. 2025).

Implementation Details. For a fair comparison, we set the global round $T = 100$, batch size $B = 256$, and embedding dimension $d = 16$ for all methods. For baselines, we adopt the optimal settings for other hyperparameters as reported in their original papers. For our method, we set the learning rate $\eta = \beta = 0.1$, and local epoch $E = 10$ with the Adam optimizer. All methods converge under the given settings.

Overall Performance

The performance comparison of different methods is summarized in Table 1. FedEM outperforms centralized methods on most datasets. As a pFR method, FedEM customizes item embeddings for each client, enabling it to better capture user preferences compared to centralized methods that share the same item embeddings across all clients. FedEM also surpasses other federated methods. Existing pFR methods heuristically design additional personalization mechanisms without tackling the aggregation bottleneck, *i.e.*, the loss of local personalization caused by global aggregation. In contrast, our method effectively and efficiently leverages off-the-shelf personalized information from the local model to bridge this gap. Additionally, on relatively dense datasets such as ML-100K and ML-1M, pFR methods, including FedEM, can surpass centralized methods due to the locally sufficient training of personalized models. However, for highly sparse datasets such as LastFM-2K, with a sparsity level of up to 99.07%, the federated setting severely limits local

Datasets	Methods	CenRec			FedRec								Ours
		MF	NCF	LightGCN	FedMF	FedNCF	FedFast	PFedRec	CoLR	GPFedRec	FedRAP	FedCIA	FedEM
Filmtrust	HR@10	0.6936	0.6856	0.7804	0.6577	0.6597	0.6637	0.6756	0.6377	0.6836	0.8024	0.8543	**0.8932**
	NDCG@10	0.5341	0.5476	0.6475	0.5290	0.5337	0.4951	0.5398	0.5105	0.5425	0.5455	0.6992	**0.7701**
ML-100K	HR@10	0.6585	0.6066	0.8356	0.4889	0.4252	0.4687	0.6882	0.4952	0.7010	0.8897	0.9512	**0.9958**
	NDCG@10	0.3781	0.3398	0.5754	0.2721	0.2290	0.2702	0.3913	0.2772	0.4069	0.7950	0.7741	**0.9427**
ML-1M	HR@10	0.6053	0.5897	0.8217	0.4871	0.4230	0.4137	0.6730	0.4533	0.6776	0.8619	0.9012	**0.9507**
	NDCG@10	0.3376	0.3325	0.5478	0.2733	0.2285	0.2333	0.3898	0.2475	0.3973	0.7661	0.7890	**0.8392**
LastFM-2K	HR@10	0.8440	0.7904	0.8463	0.5934	0.4996	0.5225	0.7833	0.5335	**0.7975**	0.6210	0.7108	0.7935
	NDCG@10	0.6161	0.6024	0.6890	0.3963	0.3307	0.3239	0.6822	0.3580	0.6690	0.5923	0.6601	**0.7151**

Table 1: Performance comparison on four datasets, reported by HR@10 and NDCG@10. CenRec and FedRec represent centralized and federated recommendation methods, respectively. The best FedRec results are bold, and the second ones are underlined.

data availability, making it difficult to outperform centralized methods. Nevertheless, FedEM still achieves superior performance compared to all pFR methods under such challenging scenarios, demonstrating its general applicability.

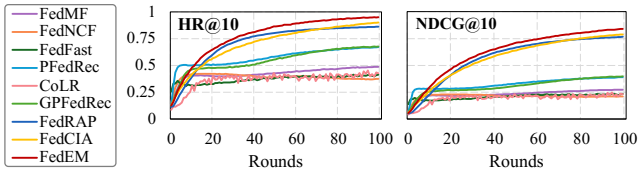

Figure 4: Convergence comparison on the ML-1M dataset.

We further compare the convergence of different methods, with results on the ML-1M dataset shown in Figure 4. Compared with other methods, FedEM demonstrates a faster convergence in the early stages and achieves more stable and superior performance in the later stages. This is attributed to the elastic merging scheme, which retains the local information from the previous round, thereby enhancing both performance and convergence stability.

Datasets	Metrics	FedMF w/ SR	FedSim w/ SR	FedSim w/ SM	FedSim w/ DM	FedSim w/ EM
Filmtrust	HR@10	0.6577	0.7206	0.7774	0.8433	**0.8932**
	NDCG@10	0.5290	0.5501	0.5959	0.7051	**0.7701**
ML-100K	HR@10	0.4889	0.6320	0.7190	0.9427	**0.9958**
	NDCG@10	0.2721	0.4457	0.5277	0.8175	**0.9427**
ML-1M	HR@10	0.4871	0.6166	0.7219	0.9078	**0.9507**
	NDCG@10	0.2733	0.4103	0.5278	0.7839	**0.8392**
LastFm-2K	HR@10	0.5934	0.7258	0.7455	0.7912	**0.7935**
	NDCG@10	0.3963	0.6498	0.6633	0.7131	**0.7151**

Table 2: Ablation study results. The best results are in bold.

Ablation Study

Component analysis. Based on the backbone FedMF, we apply similarity-based aggregation on the server side and denote this variant as FedSim. Moreover, to validate the rationality and effectiveness of the EM module, we introduce the following variants: (1) Static Replacement (SR): Replace the local model with the global model for local training. (2) Static Merging (SM): Set a fixed scalar ρ shared across all clients to merge the global and local models. (3) Dynamic Merging (DM): Each client adopts a local adapter to generate a personalized scalar ρ for model merging. (4) Elastic Merging (EM): The adapter generates a vector $\boldsymbol{\rho}$ to perform fine-grained model merging at the item level for each client.

Figure 5: Model visualization of item embeddings on ML-100K. Items interacted with by the user are marked with red $+$, while unobserved items are marked with blue $\times$.

From the results in Table 2, we observe that incorporating similarity into global aggregation helps mitigate the negative impact of client heterogeneity on local personalization. Focusing on local strategies, the SR scheme adopted by existing methods directly discards the local model, leading to suboptimal performance. By merging the global and local models, SM retains some personalized information and achieves better results. However, SM applies the same merging weight to all clients, neglecting client heterogeneity and diverse personalization needs. DM improves upon SM by introducing the adapter to generate client-specific weights, enabling more tailored merging. Building on DM, EM further leverages the embedding-based nature of recommendation models to perform fine-grained merging at the item level. This design allows for a more balanced fusion of collaborative information from global aggregation and the off-the-shelf personalized information from local training, offering a more efficient and effective solution.

Datasets	Metrics	FedMF			FedNCF			PFedRec			GPFedRec			FedRAP		
		w/o EM	w/ EM	Improv.	w/o EM	w/ EM	Improv.	w/o EM	w/ EM	Improv.	w/o EM	w/ EM	Improv.	w/o EM	w/ EM	Improv.
Filmtrust	HR@10	0.6577	0.7804	↑ 18.66%	0.6597	0.7315	↑ 10.88%	0.6756	0.9691	↑ 43.44%	0.6836	0.8723	↑ 27.60%	0.8024	0.8044	↑ 0.25%
	NDCG@10	0.5290	0.6426	↑ 21.47%	0.5337	0.6852	↑ 28.39%	0.5398	0.8544	↑ 58.28%	0.5425	0.6706	↑ 23.61%	0.5455	0.5498	↑ 0.79%
ML-100K	HR@10	0.4889	0.8759	↑ 79.16%	0.4252	0.7529	↑ 77.07%	0.6882	0.9905	↑ 43.93%	0.7010	0.9247	↑ 31.91%	0.8897	0.9735	↑ 9.42%
	NDCG@10	0.2721	0.7533	↑ 176.85%	0.2290	0.7162	↑ 212.75%	0.3913	0.9238	↑ 136.08%	0.4069	0.6938	↑ 70.51%	0.7950	0.8964	↑ 12.75%
ML-1M	HR@10	0.4871	0.8199	↑ 68.32%	0.4230	0.7606	↑ 79.81%	0.6730	0.9656	↑ 43.48%	0.6776	0.8823	↑ 30.21%	0.8619	0.9538	↑ 10.66%
	NDCG@10	0.2733	0.6606	↑ 141.71%	0.2285	0.6979	↑ 205.43%	0.3898	0.7978	↑ 104.67%	0.3973	0.6777	↑ 70.58%	0.7661	0.8648	↑ 12.88%
LastFM-2K	HR@10	0.5934	0.7013	↑ 18.18%	0.4925	0.6564	↑ 33.28%	0.7833	0.8810	↑ 12.47%	0.7975	0.8385	↑ 5.14%	0.6210	0.6320	↑ 1.77%
	NDCG@10	0.3963	0.6061	↑ 52.94%	0.3215	0.5445	↑ 69.36%	0.6822	0.8052	↑ 18.03%	0.6690	0.7475	↑ 11.73%	0.5923	0.5869	–

Table 3: Performance improvement by integrating the Elastic Merging (EM) module into existing FR/pFR baselines. "Improv." indicates the performance gain over the original baselines.

Model Visualization. We incorporate the EM module into FedMF and perform t-SNE visualization of the local, global, and merged models, as shown in Figure 5. The locally trained model contains rich personalized information, with the preferred items being relatively concentrated. Although global aggregation introduces more diverse collaborative information into the model, it also disrupts user preferences. Using such a global model for local training can harm personalization. In contrast, the EM module performs model merging at the item level, effectively leveraging the off-the-shelf personalized information from the local model while selectively incorporating collaborative information from the global model, leading to better performance.

Compatibility Study

Our proposed EM module can be seamlessly integrated as a plug-in into existing FR/pFR methods, aiming to overcome the performance bottleneck caused by global aggregation. Specifically, we take FedMF, FedNCF, PFedRec, GPFedRec, and FedRAP as examples, where elastic merging is performed after the client downloads the global model, instead of following static replacement. As shown in Table 3, all methods exhibit significant performance improvements after incorporating the EM module, validating both the limitations of the static replacement and the effectiveness of elastic merging. The EM module is lightweight and easily integrable, incurring little to no overhead on local clients and demonstrating good compatibility with existing models.

Hyperparameter Analysis

Adapter architecture. In our experiments, we set the embedding dimension to $d = 16$, and by default, the adapter is implemented as an MLP with three hidden layers, following the structure $[32, 16, 8, 1]$. Additionally, we explore MLPs with varying numbers of hidden layers, and the corresponding results are illustrated in Figure 6. When the number of MLP layers L is small, the adapter struggles to effectively balance the local and global models, leading to suboptimal results. As L increases, the model performance does not improve significantly, but the adapter becomes more complex. The default structure achieves a good trade-off between performance and architectural simplicity.

Similarity Coefficient. During global aggregation, we find that setting the similarity coefficient α around 1 can effectively alleviate the negative impact of heterogeneity discussed in Lemma 1 and improve overall model performance.

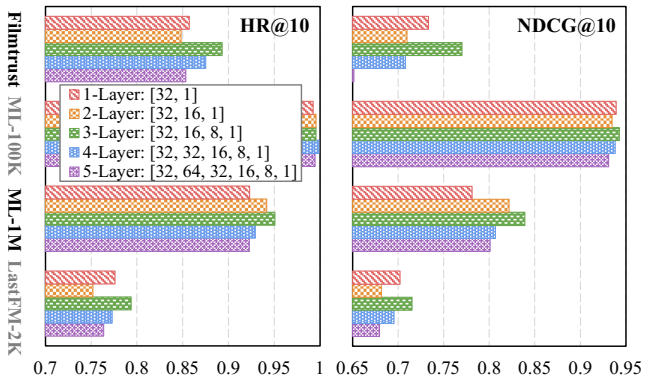

Figure 6: Performance under different adapter architectures.

Privacy Protection

We evaluate the performance of privacy-enhanced FedEM by incorporating the LDP strategy. As shown in Table 4, the performance on all datasets slightly degrades with LDP, but remains within an acceptable range and still outperforms other baselines, demonstrating the robustness of our method.

Datasets	Filmtrust		ML-100K		ML-1M		LastFM-2K	
	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
w/o LDP	0.8932	0.7701	0.9958	0.9427	0.9507	0.8392	0.7935	0.7151
w/ LDP	0.8802	0.7353	0.9936	0.9399	0.9392	0.8193	0.7707	0.6930
Degrade	↓ 1.46%	↓ 4.52%	↓ 0.22%	↓ 0.30%	↓ 1.21%	↓ 2.37%	↓ 2.87%	↓ 3.09%

Table 4: Performance of applying LDP to our method. H and N are short for HR and NDCG, respectively.

Conclusion

In this work, we experimentally and theoretically identify a performance bottleneck in FR caused by global aggregation. Unlike existing methods that heuristically design personalization mechanisms, we address this bottleneck directly with a simple yet effective method called FedEM. Grounded in model merging theory, FedEM exploits the off-the-shelf local models to compensate for the aggregated global model. It elastically strikes a balance between global collaboration and local personalization, achieving excellent performance and compatibility compared to other SOTA methods.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities 2023JBZY031, in part by the National Science Foundation of China under Grant U2268203, 62402031, 62203040, and 62372436, in part by the Beijing Nova Program 20240484620. It is also supported by Beijing Postdoctoral Research Foundation, Q6059000202501.

References

Ammad-Ud-Din, M.; Ivannikova, E.; Khan, S. A.; Oyomno, W.; Fu, Q.; Tan, K. E.; and Flanagan, A. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*.

Bayer, I.; He, X.; Kanagal, B.; and Rendle, S. 2017. A generic coordinate descent framework for learning from implicit feedback. In *WWW*, 1341–1350.

Bottou, L.; Curtis, F. E.; and Nocedal, J. 2018. Optimization methods for large-scale machine learning. *SIAM review*, 60(2): 223–311.

Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In *RecSys*, 387–388.

Chai, D.; Wang, L.; Chen, K.; and Yang, Q. 2020. Secure federated matrix factorization. *IEEE Intelligent Systems*, 36(5): 11–20.

Chen, G.; Zhang, X.; Su, Y.; Lai, Y.; Xiang, J.; Zhang, J.; and Zheng, Y. 2023. Win-win: a privacy-preserving federated framework for dual-target cross-domain recommendation. In *AAAI*, 4149–4156.

Chen, J.; Zhang, H.; Li, H.; Zhang, C.; Li, Z.; and Li, Y. 2025. Beyond Personalization: Federated Recommendation with Calibration via Low-rank Decomposition. *arXiv preprint arXiv:2506.09525*.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In *ICML*, 2089–2099.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*, 3557–3568.

Feng, J.; Lai, Y.; Sun, H.; and Ren, B. 2025. SADBA: Self-Adaptive Distributed Backdoor Attack Against Federated Learning. In *AAAI*, 16568–16576.

Feng, J.; Yang, L. T.; Zhu, Q.; and Choo, K.-K. R. 2018. Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment. *IEEE TDSC*, 17(4): 857–868.

Guo, G.; Zhang, J.; and Yorke-Smith, N. 2013. A Novel Bayesian Similarity Measure for Recommender Systems. In *IJCAI*, 2619–2625.

Han, M.; Li, D.; Xia, J.; Liu, J.; Gu, H.; Zhang, P.; Gu, N.; and Lu, T. 2025. FedCIA: Federated Collaborative Information Aggregation for Privacy-Preserving Recommendation. In *SIGIR*, 1687–1696.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *ACM TIIS*, 5(4): 1–19.

He, X.; Chen, T.; Kan, M.-Y.; and Chen, X. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *CIKM*, 1661–1670.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, 639–648.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *WWW*, 173–182.

Huang, W.; Ye, M.; Shi, Z.; Wan, G.; Li, H.; Du, B.; and Yang, Q. 2024. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE TPAMI*, 46(12): 9387–9406.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized cross-silo federated learning on non-iid data. In *AAAI*, 7865–7873.

Ji, S.; Pan, S.; Long, G.; Li, X.; Jiang, J.; and Huang, Z. 2019. Learning private neural language modeling with attentive aggregation. In *IJCNN*, 1–8.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 30–37.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021. Ditto: Fair and robust federated learning through personalization. In *ICML*, 6357–6368.

Li, Z.; Long, G.; and Zhou, T. 2024. Federated recommendation with additive personalization. In *ICLR*.

Luo, S.; Xiao, Y.; and Song, L. 2022. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *CIKM*, 4289–4293.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 1273–1282.

Muhammad, K.; Wang, Q.; O'Reilly-Morgan, D.; Tragos, E.; Smyth, B.; Hurley, N.; Geraci, J.; and Lawlor, A. 2020. Fedfast: Going beyond average for faster training of federated recommender systems. In *SIGKDD*, 1234–1242.

Nguyen, N.-H.; Nguyen, T.-A.; Nguyen, T.; Hoang, V. T.; Le, D. D.; and Wong, K.-S. 2024. Towards efficient communication and secure federated recommendation system via low-rank training. In *WWW*, 3940–3951.

Perifanis, V.; and Efraimidis, P. S. 2022. Federated neural collaborative filtering. *Knowledge-Based Systems*, 242: 108441.

Qi, T.; Wang, H.; and Huang, Y. 2024. Towards the Robustness of Differentially Private Federated Learning. In *AAAI*, 19911–19919.

Sun, Z.; Xu, Y.; Liu, Y.; He, W.; Kong, L.; Wu, F.; Jiang, Y.; and Cui, L. 2024. A survey on federated recommendation systems. *IEEE TNNLS*, 36(1): 6–20.

T Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized federated learning with moreau envelopes. In *NeurIPS*, 21394–21405.

Yang, E.; Shen, L.; Guo, G.; Wang, X.; Cao, X.; Zhang, J.; and Tao, D. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

Ye, R.; Ni, Z.; Wu, F.; Chen, S.; and Wang, Y. 2023. Personalized federated learning with inferred collaboration graphs. In *ICML*, 39801–39817.

Yin, H.; Qu, L.; Chen, T.; Yuan, W.; Zheng, R.; Long, J.; Xia, X.; Shi, Y.; and Zhang, C. 2024. On-Device Recommender Systems: A Comprehensive Survey. *arXiv preprint arXiv:2401.11441*.

Zhang, C.; Long, G.; Zhou, T.; Yan, P.; Zhang, Z.; Zhang, C.; and Yang, B. 2023a. Dual personalization on federated recommendation. In *IJCAI*, 4558–4566.

Zhang, C.; Long, G.; Zhou, T.; Zhang, Z.; Yan, P.; and Yang, B. 2024a. GPFedRec: Graph-guided personalization for federated recommendation. In *SIGKDD*, 4131–4142.

Zhang, H.; Li, H.; Chen, J.; Cui, S.; Yan, K.; Wuerkaixi, A.; Zhou, X.; Shen, Z.; and Li, Y. 2024b. Beyond Similarity: Personalized Federated Recommendation with Composite Aggregation. *arXiv preprint arXiv:2406.03933*.

Zhang, H.; Li, Z.; Li, H.; Zhou, X.; Zhang, J.; and Li, Y. 2026. Transfr: Transferable federated recommendation with adapter tuning on pre-trained language models. In *AAAI*.

Zhang, H.; Luo, F.; Wu, J.; He, X.; and Li, Y. 2023b. LightFR: Lightweight Federated Recommendation with Privacy-preserving Matrix Factorization. *ACM TOIS*, 41(4): 1–28.

Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; and Guan, H. 2023c. Fedala: Adaptive local aggregation for personalized federated learning. In *AAAI*, 11237–11244.

Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; and Alvarez, J. M. 2021. Personalized federated learning with first order model optimization. In *ICLR*.

Zheng, H.; Hu, Z.; Yang, L.; Zheng, M.; Xu, A.; and Wang, B. 2025. ConFREE: Conflict-free Client Update Aggregation for Personalized Federated Learning. In *AAAI*, 22875–22883.

Zhou, Z.; Chen, Z.; Chen, Y.; Zhang, B.; and Yan, J. 2024. On the Emergence of Cross-Task Linearity in Pretraining-Finetuning Paradigm. In *ICML*.