Beyond Similarity: Personalized Federated Recommendation with Composite Aggregation

HONGLEI ZHANG, Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China
HAOXUAN LI, Center for Data Science, Peking University, Beijing, China
JUNDONG CHEN, School of Cyberspace Science and Technology, Beijing Jiaotong University, Beijing, China
SEN CUI, KUNDA YAN, and ABUDUKELIMU WUERKAIXI, Institute for Artificial Intelligence, Tsinghua University, Beijing, China
XIN ZHOU and ZHIQI SHEN, College of Computing and Data Science, Nanyang Technological University, Singapore, Singapore
YIDONG LI, Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China

Federated recommendation aims to collect global knowledge by aggregating local models from massive devices, to provide recommendations while ensuring privacy. Current methods mainly leverage aggregation functions invented by federated vision community to aggregate parameters from similar clients, e.g., clustering aggregation. Despite considerable performance, we argue that it is suboptimal to apply them to federated recommendation directly. This is mainly reflected in the disparate model structures. Different from structured parameters like convolutional neural networks in federated vision, federated recommender models usually distinguish itself by employing one-to-one item embedding table. Such a discrepancy induces the challenging *embedding skew* issue, which continually updates the trained embeddings but ignores the non-trained ones during aggregation, thus failing to predict future items accurately. To this end, we propose a personalized Federated recommendation model with Composite Aggregation (FedCA), which not only aggregates *similar* clients to enhance trained embeddings but also aggregates *complementary* clients to update non-trained

This work was supported by the National Natural Science Foundation of China under Grants U2268203, 62402031, 62203040, and 62372436, in part by the Beijing Nova Program (20240484620), and in part by the Fundamental Research Funds for the Central Universities (2024XKRC086). H. Li was supported by the National Natural Science Foundation of China (623B2002).

Authors' Contact Information: Honglei Zhang, Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, School of Computer Science and Technology, Beijing Jiaotong University, **Beijing,** China; e-mail: honglei.zhang@bjtu.edu.cn; Haoxuan Li, Center for Data Science, Peking University, **Beijing,** China; e-mail: hxli@stu.pku.edu.cn; Jundong Chen, School of Cyberspace Science and Technology, Beijing Jiaotong University, **Beijing,** China; e-mail: jundongchen@bjtu.edu.cn; Sen Cui, Institute for Artificial Intelligence, Tsinghua University, **Beijing,** China; e-mail: cuis@mail.tsinghua.edu.cn; Kunda Yan, Institute for Artificial Intelligence, Tsinghua University, **Beijing,** China; e-mail: ykd22@mails.tsinghua.edu.cn; Abudukelimu Wuerkaixi, Institute for Artificial Intelligence, Tsinghua University, **Beijing,** China; e-mail: wekxabdk21@mails.tsinghua.edu.cn; Xin Zhou, College of Computing and Data Science, Nanyang Technological University, Singapore, Singapore; e-mail: xin.zhou@ntu.edu.sg; Zhiqi Shen, College of Computing and Data Science, Nanyang Technological University, Singapore, Singapore; e-mail: zqshen@ntu.edu.sg; Yidong Li(corresponding author), Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, School of Computer Science and Technology, Beijing Jiaotong University, **Beijing,** China; e-mail: ydli@bjtu.edu.cn.

ACM 1558-2868/2026/1-ART46
https://doi.org/10.1145/3779442

embeddings. Besides, we formulate the overall learning process into a unified optimization algorithm to jointly learn the similarity and complementarity. Extensive experiments on several real-world datasets substantiate the effectiveness of our proposed model. Our code is available at https://github.com/hongleizhang/FedCA.

CCS Concepts: • **Information systems → Collaborative filtering**; • **Security and privacy → Privacy protections**;

Additional Key Words and Phrases: Federated Recommendation, Composite Aggregation, Embedding Skew, Model Similarity, Data Complementary

ACM Reference format:
Honglei Zhang, Haoxuan Li, Jundong Chen, Sen Cui, Kunda Yan, Abudukelimu Wuerkaixi, Xin Zhou, Zhiqi Shen, and Yidong Li. 2026. Beyond Similarity: Personalized Federated Recommendation with Composite Aggregation. *ACM Trans. Inf. Syst.* 44, 2, Article 46 (January 2026), 28 pages.
https://doi.org/10.1145/3779442

1 Introduction

Federated Recommendation (FR), as an emerging distributed learning paradigm [5, 57, 59], has attracted significant interest from both academia [2, 58] and industry [17, 43]. Existing FRs typically employ different collaborative filtering backbones as their local models [19, 27, 47] and perform various aggregation functions to obtain a global recommender, following basic **Federated Learning (FL)** principles [38]. For instance, one pioneering work is FCF [1], which is an adaptation of centralized matrix factorization by performing local updates and global aggregation with federated optimization. FedNCF [44] integrates the linearity of matrix factorization with the non-linearity of deep embedding techniques, building upon the foundations of FCF. Besides, FedIAR integrates personalized item embedding exploration, enhancing global representation while achieving personalized collaboration for FR [62]. These embedding-based FR models effectively balance model accuracy and data privacy [21, 32, 54].

Generally, the success of FRs stems from their capability to embody data locality while achieving knowledge globality across multiple clients through aggregation functions [15, 31, 50]. These functions play a crucial role in federated optimization, determining which knowledge from each client and to what extent it is integrated into the global model [53]. Among them, the most well-known method is FedAvg, which allocates larger weights to clients with more data samples to perform weighted aggregation, thus achieving better knowledge collection [38]. Subsequent works aim to improve aggregation strategies to address the data heterogeneity challenge in federated settings [12, 22, 34]. For instance, PerFedRec first exploits clustering mechanisms to identify clients with similar data distributions and then conducts group-wise aggregation to accomplish the adaptation process [35]. FedEM introduces an elastic model merging mechanism that blends global aggregated parameters with preserved local models, mitigating the aggregation bottleneck [8]. Besides, PFedCLR introduces a dual-function module that calibrates user embedding skew caused by aggregation process while personalizing item embeddings via low-rank buffer decomposition, enabling more accurate global aggregation [7]. The above aggregation methods effectively mitigate the heterogeneity challenge by considering fine-grained model similarity.

Notably, such aggregation functions utilized in FRs are primarily inspired by those in federated vision community, such as weighted aggregation [38], clustering aggregation [34], and attentive aggregation [22]. Here, "federated vision community" in this work refers to traditional FL methods such as FedAvg [38] and FedProx [29], which primarily utilize **Convolutional Neural Networks (CNNs)** to perform vision tasks in federated settings. All of these aggregation methods are essentially rely on model similarity, where similar clients with consistent parameter distributions

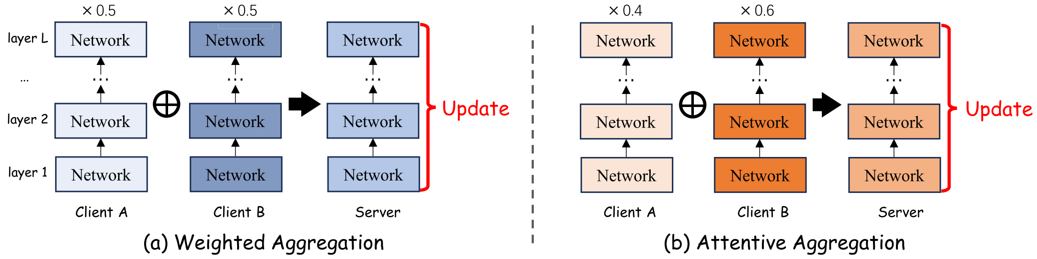


Fig. 1. Taking two clients as an ideal example in federated vision tasks, ⊕ denotes aggregation operators. We illustrate similarity-based aggregation using weighted aggregation (a) and attentive aggregation (b) with multi-layer networks, where 0.5 serves as the average coefficient, and 0.4 and 0.6 represent the attention coefficients. Similarity aggregation can continuously update model parameters in federated vision tasks.

are assigned more weights, while dissimilar clients are given relatively smaller weights. Hence, similarity-based aggregation primarily distinguishes among clients by their model parameter distributions and performs weighted aggregation based on model similarity, thereby alleviating heterogeneity. Despite achieving satisfactory performance, we argue that directly adopting off-the-shelf aggregation functions from federated vision domain may not be well-adapted to FR tasks, which naturally exhibit significant heterogeneity and are highly required personalization preference for each client.

The reason for this research gap is mainly reflected in the disparate model structures between federated vision and federated recommender tasks [53, 60]. Specifically, federated vision models, e.g., CNNs [51], typically with a deep structure involving multiple networks (a.k.a., structured parameters), as shown in Figure 1. Owing to the nonlinear mapping of structured parameters, local features are encoded into the parameters of each layer. As long as local data are available, the parameters of all layers in this client are updated, leaving no network layer untouched. As depicted in Figure 1(a), by aggregating the parameters of clients A and B with similar distributions (light blue and dark blue, respectively), an aggregated model (medium blue) can be obtained. Consequently, similarity-based aggregation can achieve a more optimal parameter space for all network layers by aggregating parameters from similar clients. Unlike federated vision models, federated recommender models usually distinguish itself by employing one-to-one item embedding table, as shown in Figure 2. Since different clients may involve distinct subsets of interacted items, leading to different rows trained in the embedding table for each client. When only relying on model similarity aggregation, it leads to the unique *embedding skew* issue in FRs, where trained embeddings (blue) continually improve while non-trained embeddings (gray) keep intact or even deteriorate during aggregation, as depicted in Figure 2(a). Hence, it poses a great challenge to predict uninteracted items in local device solely by similarity aggregation.

In this work, we take the first step in exploring aggregation mechanisms for FR models and identify the unique embedding skew issue in FR tasks. In light of this, we propose a composite aggregation mechanism tailored to embedding tables in FR scenarios, which considers not only model similarity but also data complementarity. We provide a theoretical guarantee that fine-grained heterogeneity modeling requires the mutual reinforcement of model similarity and data complementarity. It is important to note that similarity and complementarity are not mutually exclusive. High data complementarity does not imply small model similarity. Such a mechanism can aggregate not only similar clients but also complementary ones, thus updating the already trained embeddings, and enhancing those that were not trained. Hence, it can improve the ability to predict future items on edge devices in FR tasks, as shown in Figure 2(b). Building on model

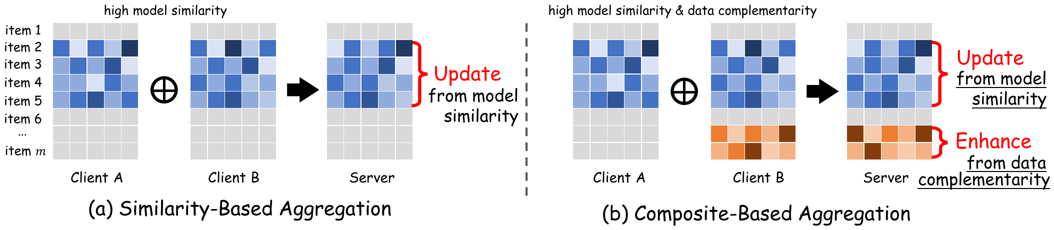

high model similarity
item 1
item 2
item 3
item 4
item 5
item 6
...
item m
⊕
Client A
Client B
Server
Update from model similarity
(a) Similarity-Based Aggregation
high model similarity & data complementarity
Client A
Client B
Server
Update from model similarity
Enhance from data complementarity
(b) Composite-Based Aggregation

Fig. 2. Taking two clients as an ideal example in FR tasks, ⊕ denotes aggregation operators. Previous work can only update trained items repeatedly within embedding tables via similarity aggregation (a), while our composite aggregation (b) can both update trained items and enhance non-trained items.

similarity, we introduce data complementarity as an additional source of information, thereby expanding the scope of item updates. This enables not only the updating of items already captured by model similarity but also the enhancement of new items derived from data complementarity. Besides, we formulate the aggregation process into a unified optimization framework to jointly learn the similarity and complementarity metrics, encompassing several classical aggregation methods. Extensive experiments on several datasets show that our proposed model consistently outperforms several state-of-the-art methods.

In summary, our main contributions are listed as follows:

- We identify the embedding skew issue caused by aggregating embedding tables in FR tasks from empirical analysis. From theoretical perspectives, we rethink the heterogeneity in FRs to account for embedding skew. Building on traditional model similarity, we introduce a fine-grained heterogeneity modeling mechanism for the introduction of data complementarity.
- We propose a composite aggregation mechanism tailored for FR tasks, which flexibly accounts for both model similarity and data complementarity during the embedding table aggregation process to address the identified embedding skew issue.
- We introduce a unified aggregation optimization framework that encompasses various classic aggregation mechanisms and enables efficient model optimization.
- Extensive experiments on several real-world datasets show that our model consistently outperforms several state-of-the-art aggregation methods.

2 Related Work

In this section, we summarize the prior work on traditional FRs as well as recent advancements in personalized FRs. For a comprehensive review, please refer to the recent survey papers [5, 55].

2.1 Traditional FR

Traditional FRs aim to learn a shared item encoder for all clients and a private user encoder for each client [23, 30, 46, 55]. It mainly comprises three modules: a private user encoder, a shared item encoder, and a fusion module [49, 56], following basic FL principle [9, 29, 38]. Some attempts are launched to follow these three lines [44, 49, 58]. Specifically, HPFL introduced a hierarchical user encoder to differentiate between private and public user information [49]. Zhang et al. proposed federated discrete optimization model to learn binary codes of item encoder [58]. FedNCF attempted to use a multi-layer perceptron fusion module to learn non-linearities between users and items [44]. Subsequent research has built upon this foundation to explore more comprehensive FRs, such as privacy-enhanced FRs [6, 33, 60, 61], multimodal FRs [37], resource-efficient FRs [26, 41, 58], and robustness-enhanced FRs [46]. For instance, DuAda combines a dummy user simulator with an adaptive distribution attacker to craft realistic malicious clients and perform targeted poisoning in FR, while a merged adaptive defense is designed to counter such attacks. These efforts are directed toward developing a more comprehensive framework for FR.

2.2 Personalized FR

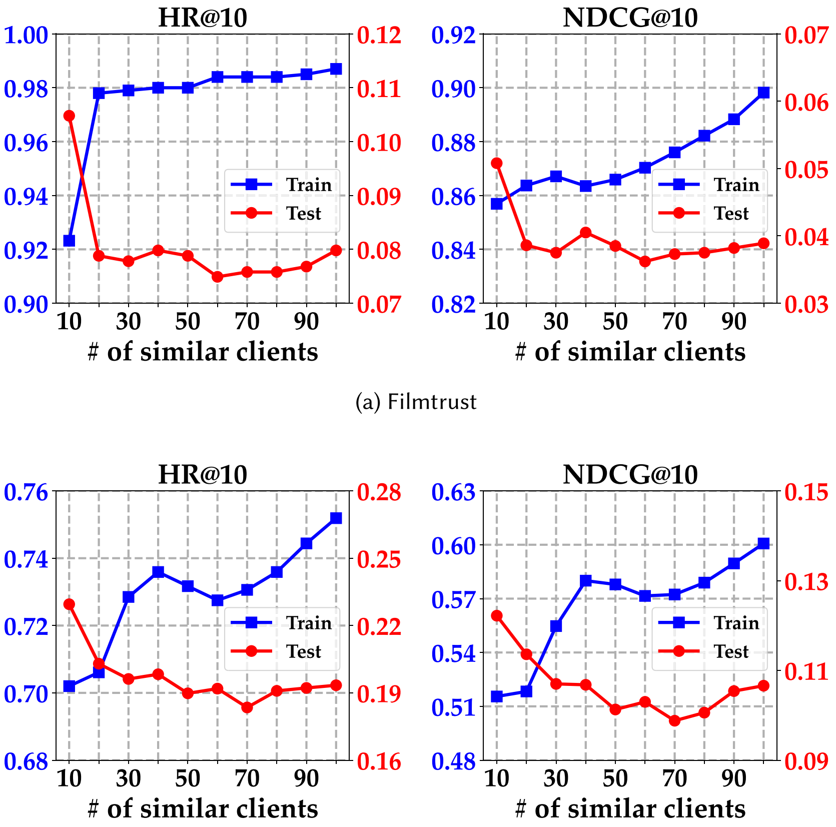
To achieve personalized FRs, some pioneer works aim to learn personalized item encoders [16, 31, 48], such as dual personalization [56] and additive personalization [31]. Specifically, PerFedRec++ leverages self-supervised graph pre-training with privacy-preserving augmentation to enhance personalization [36]. Besides, FedCF jointly encodes shared and personalized knowledge with dual encoders and a gating network, leveraging a VAE-based CF formulation to balance personalization and generalization. Note that the above methods employ the classic FedAvg for aggregation [38]. Subsequent methods attempt to improve the effectiveness of FedAvg, such as clustering aggregation [12, 20, 35, 40], attention aggregation [22], and graph aggregation [52]. For instance, FedFast [40] used clustering aggregation to enhance training efficiency, while FedAtt [22] utilized attention to learn coefficients between global and local models. pFedGraph [52] introduced a collaborative graph to learn the similarity between individuals. By modeling the relationships among clients as a graph, it can better understand the underlying patterns and dependencies, thereby improving the personalization for each client. FedCIA introduces a collaborative aggregation mechanism that aggregates item similarity matrices instead of model parameters, enabling parameter-free aggregation [15]. Despite achieving considerable results, all aggregation methods are with similarity assumption, which is more suitable for structured parameters in federated vision tasks. Different from previous work, we are the first work to design composite aggregation mechanism tailored for FR tasks, which simultaneously considers similarity and complementarity metrics to more effectively aggregate embedding tables.

3 Empirical Analysis

By analyzing disparate model structures with federated vision models, we intuitively explored the embedding skew issue that uniquely occurred during aggregation process in FR tasks. To experimentally validate our findings, as illustrated in Figure 2(a), this section conducts verification analysis on two commonly used datasets (Filmtrust [14] and Movielens [18]) in FR tasks, aiming to show the unique embedding skew issue from an empirical perspective.

Specifically, we conduct exploratory experiments with improved FedAvg model by aggregating parameters with $s \in \{10, 20, \cdots, 100\}$ most similar clients. Concretely, we first use the L2 distance to compute the uploaded model parameters to identify the 100 most similar clients for each client. These similar clients are then randomly shuffled to form a new list. We then examine the performance changes as we aggregate s clients. This procedure aims to eliminate the disturbance of performance degradation caused by sorting clients in descending order of similarity. We use the classic FedAvg algorithm to implement similarity-based aggregation. As depicted in Figure 3, it is evident that as the number of aggregated similar clients increases, the accuracy (HR@10 and NDCG@10) of the train set continues to rise, while that of the test set generally declines on both datasets. Typically, the accuracy trends of the training set and the test set should be consistent. This ultimately results in a widening gap between the train and test sets, indicative of performance degradation caused by embedding skew, i.e., trained embeddings of interacted items greatly enhanced while untrained embeddings of non-interacted items deteriorated during aggregation.

Here, we aim to clarify the distinction between identified embedding skew issue and the easily confused concept of overfitting from following aspects. (1) *Performance Trends on Test Sets.* Overfitting causes the test performance to initially improve and then decline, whereas embedding skew leads to a gradual decrease in test performance. (2) *Horizontal Axis in Plots.* Overfitting is observed as



(a) Filmtrust

(b) Movielens

Fig. 3. Empirical results regarding HR@10 and NDCG@10 on train and test sets, respectively.

the number of training iterations increases, whereas embedding skew is observed as the number of similar clients increases during aggregation processes. (3) *Implementation Mechanisms*. Overfitting occurs during the continuous learning process from training data, whereas embedding skew arises during the well-trained embedding table aggregation. Overall, the above empirical observation indicates that solely utilizing similarity to aggregate embedding tables is suboptimal. This also aligns with the motivation behind our proposed model, which exploits composite aggregation considering both similarity and complementarity. Hence, it can delicately improve generalization on test sets, thus enabling accurate modeling of future items on edge devices.

4 Problem Formulation

Here, we introduce the basic notations, general FR framework, rethinking heterogeneity in FRs, and theoretical guarantees for complementarity in FRs.

Notations. Assume there are n users/clients $\mathcal{U} = \{u\}$, and m items $\mathcal{I} = \{i\}$ stored in the server. Each user u keeps a local dataset $\mathcal{D}_u$, which comprises tuples $(u, i, r_{ui} | i \in \mathcal{I}_u)$, where $\mathcal{I}_u$ denotes the observed items for client u, and each entry $r_{ui} \in \{0, 1\}$ indicates the label for user u on item i. The goal of FRs is to predict $\hat{r}_{ui}$ of user u for each future item $i \in \mathcal{I} \setminus \mathcal{I}_u$ on local devices.

General FR Framework. Formally, the global objective of general FR over n clients is

$$\min_{(\mathbf{p}_1,\mathbf{p}_2,\cdots,\mathbf{p}_n;\Theta_1,\Theta_2,\cdots,\Theta_n)} \frac{1}{n}\sum_{u=1}^{n} p_u \mathcal{L}_u\left(\mathbf{p}_u,\Theta_u;\mathcal{D}_u\right), \tag{1}$$

where $\mathbf{p}_u$ and Θ_u denote local user embedding and local parameters of item encoder stored in clients, respectively. The server aggregates $\Theta_u^g = \sum_{u=1}^{n} p_u\Theta_u$ with aggregation weight p_u, e.g., $p_u = |\mathcal{D}_u|/|\mathcal{D}|$ in FedAvg [38] to facilitate global update. $\mathcal{L}_u$ is the task-specific objective (e.g., log loss [19]) to facilitate local training. Traditional FRs attempt to learn a global model Θ^g across n clients, where $\Theta^g = \Theta_1^g = \cdots = \Theta_n^g$ [1, 44], while personalized FRs keep different global models Θ_u^g to achieve high efficacy on their local clients [31, 56]. Note that unless otherwise specified, we use item embedding table $\mathbf{Q}_u$ to instantiate general Θ_u in following sections, since $\mathbf{Q}_u$ is the standard configuration in embedding-based FRs [1, 40].

Rethinking Heterogeneity in FRs. The FR tasks naturally exhibit great heterogeneity in each client, since the item sets interacted by each are vastly different, radically causing the embedding skew issue. Formally, we assume $p(x,y)$ to denote the joint distribution of features $x \in \{(u,i)|i \in \mathcal{I}_u\}$ and labels $y \in \{r_{ui}|i \in \mathcal{I}_u\}$. For two heterogeneous clients u and v, it is evident that $p(x^u,y^u) \neq p(x^v,y^v)$. Currently, classical methods follow the simple assumption $p(x,y) = p(y|x)$, treating the conditional distribution as equivalent to the joint distribution, to utilize model similarity to mitigate the *concept shift* problem [24], by aggregating similar conditional distributions to ensure that $p\left(y^u|x^u\right) \approx p\left(y^v|x^v\right)$ [12, 34]. However, we argue that there is significant natural heterogeneity in FRs, which necessitates fine-grained modeling of $p(x,y)$, such that the joint distribution can be expressed as $p(x,y) = p(x)p(y|x)$, the combination of the marginal and conditional distributions. Hence, heterogeneity can be modeled as $p\left(x^u\right)p\left(y^u|x^u\right) \neq p\left(x^v\right)p\left(y^v|x^v\right)$. Therefore, if we aim to ensure $p(x^u,y^u) \neq p(x^v,y^v)$, then under the premise of maintaining model similarity $p\left(y^u|x^u\right) \approx p\left(y^v|x^v\right)$, it is necessary to explicitly satisfy $p(x^u) \neq p(x^v)$. Thus, fine-grained heterogeneity modeling in FR tasks becomes essential.

Effectiveness of Complementarity in FRs. We explain the rationale of introducing data complementarity from theoretical aspects. Specifically, we assume $p(x,y)$ to denote the joint distribution of features x and labels y. For two heterogeneous clients u and v, we have $p(x^u,y^u) \neq p(x^v,y^v)$. Traditional similarity aggregation methods roughly model data distribution as $p(x,y) = p(y|x)$ and only use model similarity to alleviate heterogeneity issues [24], like that

$$p(y^u|x^u) \approx p(y^v|x^v). \tag{2}$$

In our context, we allow the local data following the joint distribution as $p(x,y) = p(x)p(y|x)$. Hence, the fine-grained modeling of heterogeneity can be expressed as

$$p\left(x^u\right)p\left(y^u|x^u\right) \neq p\left(y^v|x^v\right)p\left(x^v\right). \tag{3}$$

Compared to Equation (2), it is clear that to ensure $p(x^u,y^u) \neq p(x^v,y^v)$, then under the premise of maintaining model similarity $p\left(y^u|x^u\right) \approx p\left(y^v|x^v\right)$, it is necessary to explicitly satisfy $p(x^u) \neq p(x^v)$. This newly introduced marginal distribution $p\left(x^u\right)$ and $p\left(x^v\right)$ precisely reflects the complementarity of local data, which is where the rationale of introducing complementarity lies. Thus, we propose a composite aggregation mechanism to ensure both model similarity and data complementarity, aiming to more accurately model the fine-grained heterogeneity. Overall, we model both data complementarity $p(x^u) \neq p(x^v)$ and model similarity $p(y^u|x^u) \approx p(y^v|x^v)$ to expand the scope of item updates, so as to alleviate the identified embedding skew issue.

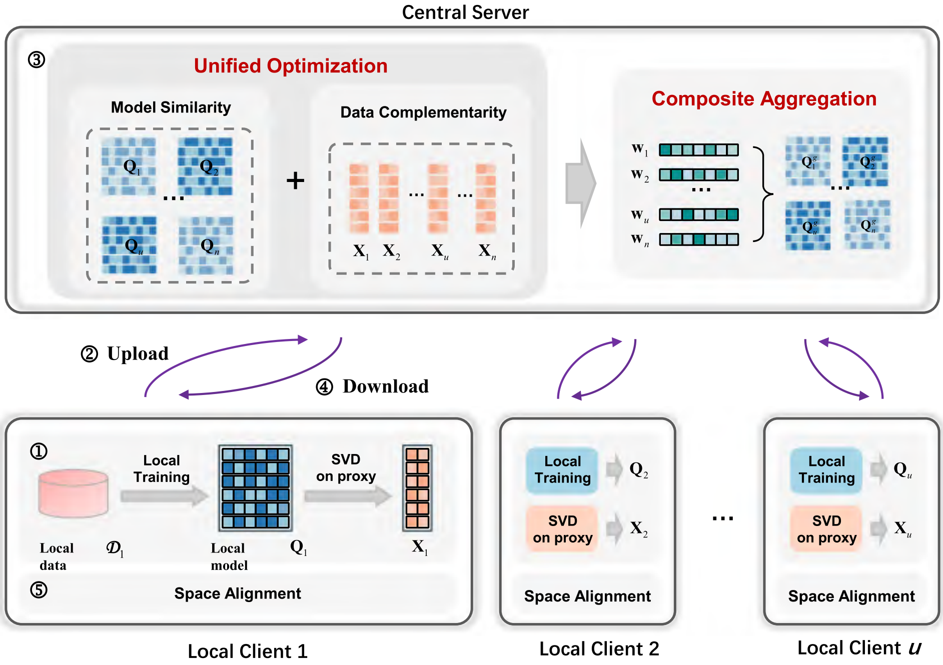


Fig. 4. The overall FedCA framework.

5 The Proposed Federated Recommendation with Composite Aggregation (FedCA) Model

In this section, we elaborate on our proposed framework personalized FedCA, which considers both model similarity and data complementarity. The goal is to alleviate the embedding skew issue inherit in FR tasks. Concretely, we first formulate a unified learning framework to optimize similarity and complementarity. Then, we provide a detailed optimization for server aggregation, followed by the local training and inference on the client side. Finally, we provide more details about FedCA and discuss the relationship between FedCA and other aggregation mechanisms.

5.1 The Overall Learning Framework

From a global perspective, we integrate the server aggregation and local training into a unified optimization framework tailored for FR tasks, as depicted in Figure 4. It aims to optimize the personalized local parameters $\{\mathbf{p}_u, \mathbf{Q}_u\}$ and aggregation weight vector $\{\mathbf{w}_u\}$ for each client, as shown in Equation (4), which is influenced by the joint constraints of similarity and complementarity.

$$\begin{aligned}\min_{\{\mathbf{p}_u,\mathbf{Q}_u,\mathbf{w}_u\}} \sum_{u=1}^{n} \Bigg(&\mathcal{L}_u\left(\mathbf{p}_u, \mathbf{Q}_u; \mathcal{D}_u\right) + \alpha \sum_{v=1}^{n} \mathcal{F}_s\left(w_{uv}; \mathbf{Q}_u; \mathbf{Q}_v\right) \\ &+ \beta \sum_{v=1}^{n} \mathcal{F}_c\left(w_{uv}; \mathcal{D}_u; \mathcal{D}_v\right) \Bigg) \\ \text{s.t.} \quad & \mathbf{1}^T \mathbf{w}_u = 1, \quad \mathbf{w}_u \geq \mathbf{0}.\end{aligned} \tag{4}$$

where the term $\mathcal{L}_u$ denotes the local empirical risk toward model parameters $\mathbf{p}_u$ and $\mathbf{Q}_u$, following the weighted aggregation of model parameters $\mathbf{Q}_u = \sum_{v=1}^{n} w_{nv}\mathbf{Q}_v^g$ downloaded from the server. The term $\mathcal{F}_s$ represents the model similarity between $\mathbf{Q}_u$ and $\mathbf{Q}_v$, while the term $\mathcal{F}_c$ quantifies the data complementarity between $\mathcal{D}_u$ and $\mathcal{D}_v$. The two constraints ensure $\mathbf{w}_u$ satisfy normalization and non-negativity. Besides, α and β are tuning coefficients. Through the unified learning framework, we jointly optimize $\mathbf{w}_u$ to a balance point to suitably aggregate item embeddings, thereby considering both similarity and complementarity during the server aggregation and local training procedures.

5.1.1 Server Aggregation. The server's responsibility is to optimize the aggregation weight $\mathbf{w}_u$ for each client u, thus achieving personalized aggregation for each client. Ideally, we aim for $\mathbf{w}_u$ to be perfectly optimized under the loss function in Equation (4). However, this is impractical due to constraints imposed by federated settings. The server can only access the local models $\mathbf{Q}_u$ uploaded by each client, without detailed knowledge of each client's user embedding $\mathbf{p}_u$ and local data $\mathcal{D}_u$, thus making it challenging to directly compute $\mathcal{L}_u$ at the server. To reasonably perceive contribution of each client, we utilize the mean squared loss between $\mathbf{w}_u$ and relative quantity of local data $\mathbf{p}$ as a proxy for $\mathcal{L}_u$, measuring the optimization level of each client, inspired by recent work [52]. Hence, the loss function to optimize $\mathbf{w}_u$ at the server side is rewritten as

$$\begin{aligned} \min_{\mathbf{w}_u} \sum_{v=1}^{n} ((w_{uv} - p_v)^2 + \alpha \mathcal{F}_s(w_{uv}; \mathbf{Q}_u; \mathbf{Q}_v) \\ + \ \beta \mathcal{F}_c(w_{uv}; \mathcal{D}_u; \mathcal{D}_v)) \\ \text{s.t.} \quad \mathbf{1}^T \mathbf{w}_u = 1, \quad \mathbf{w}_u > \mathbf{0}. \end{aligned} \tag{5}$$

The above formulation minimizes the pre-defined supervised loss while balancing the similarity and complementarity. To measure the similarity between conditional distributions of clients $p(y|x)$, we adopt common practices [24], i.e., local model parameters to capture the mapping from the marginal distribution $p(x)$ to the label distribution $p(y)$. Hence the term $\mathcal{F}_s$ can be represented as

$$\mathcal{F}_s(w_{uv}; \mathbf{Q}_u; \mathbf{Q}_v) = (w_{uv} - \sigma(\mathbf{Q}_u, \mathbf{Q}_v))^2, \tag{6}$$

where $\sigma(\cdot)$ denotes the similarity function and here $\sigma(\mathbf{Q}_u, \mathbf{Q}_v) = 1/(1+ \parallel \mathbf{Q}_u - \mathbf{Q}_v \parallel^2)$. It can be switched to any similarity function, e.g., cosine similarity. The function $\mathcal{F}_s$ ensures that the aggregation weight w_{uv} increases when the models of two clients are highly similar. To assess the complementarity of client data about marginal distributions $p(x)$ at the server side, we utilize the intermediate features as proxies for local data $\mathcal{D}_u$, i.e., the subset of item embeddings $\mathbf{Q}_u^s$ corresponding to the local interacted item sets $\mathcal{I}_u$. To further guard against input reconstruction attacks in FRs [58], we perform **Singular Value Decomposition (SVD)** on $\mathbf{Q}_u^s$ and then retain the left singular matrix with first k columns. This yields a privacy-enhanced representation $\mathbf{X}_u$ of the local data. Inspired by mutual information theory [28], the term $\mathcal{F}_c$ can be represented as

$$\mathcal{F}_c(w_{uv}; \mathcal{D}_u; \mathcal{D}_v) = -w_{uv} \cdot \cos(\phi(\mathbf{X}_u, \mathbf{X}_v)), \tag{7}$$

where $\phi(\mathbf{X}_u, \mathbf{X}_v) = \frac{1}{k} \sum_{l=1}^{k} \arccos({\mathbf{x}_u^l}^T \mathbf{x}_v^l)$. $\phi(\cdot)$ is used to measure the angle between the lth singular vectors corresponding to the data of two clients. This adapted approach relies on limited, non-sensitive information to determine the degree of complementarity between clients [45]. The principle angle method offers a geometric perspective for measuring the distance between subspaces. Further ablation experiments explaining the rationale for using this metric can be found in the experimental section. When the lengths of two vectors are unequal, we apply a padding operation to keep consistency in lengths. The function $\mathcal{F}_c$ ensures that when the angle between two clients is orthogonal, the smaller the mutual information between $\mathbf{X}_u$ and $\mathbf{X}_v$, implying a great complementarity between the two clients. We denote the similarity vector computed by $\sigma(\cdot)$ for each user u as $\mathbf{s}_u$ and the complementarity vector computed by $\phi(\cdot)$ as $\mathbf{c}_u$. We can intuitively see that Equation (5) can be easily rewritten as a standard quadratic program problem regarding the aggregation weight $\mathbf{w}_u$. Hence, it can be efficiently solved by classic convex optimization solvers [10].

Optimizing $\mathbf{w}_u$ *for Composite Aggregation.* To solve for $\mathbf{w}_u$ in Equation (4), we first vectorize the following variables. The relative dataset sizes for all clients can be represented as a vector $\mathbf{p} = [p_1, p_2, \cdots, p_n]^T$. We denote the model similarity of user u to other clients as a vector $\mathbf{s}_u$ and the data complementarity of user u to other clients as a vector $\mathbf{c}_u$. Thus, we can transform Equation (4) through derivation into a standard quadratic form:

$$\begin{aligned} \mathbf{w}_u = \arg\min_{\mathbf{x}} \ & \mathbf{x}^T\mathbf{x} + (-2\mathbf{p} - 2\alpha\mathbf{s}_u - \beta\mathbf{c}_u)^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{1}^T\mathbf{x} = 1, \quad -\mathbf{x} \leq \mathbf{0}. \end{aligned} \tag{8}$$

We can solve for the optimal personalized aggregation weight $\mathbf{w}_u$ for each client with Equation (8). Existing convex optimization solvers, such as the cvxpy package in PyTorch can efficiently solve this quadratic program problem. By alternately solving for $\mathbf{w}_u$ in Equation (8) on the server and $\mathbf{p}_u$ and $\mathbf{Q}_u$ on the local clients, we can ultimately achieve model convergence in federated optimization procedure. To improve the efficiency of optimizing the aggregation coefficient $\mathbf{w}_u$, we explore possible approximations and efficient solvers for the underlying **Quadratic Programming (QP)** problem, aiming to enable large-scale optimization in future applications. As for possible approximations, first-order methods can be employed to trade a small loss in accuracy for significant gains in scalability. In particular, gradient descent or projected gradient methods can be used in place of exact second-order solutions, with further acceleration achieved through Nesterov's momentum or coordinate descent. As for possible efficient solvers, alternating direction method of multipliers provides an efficient framework well-suited for distributed and federated settings [4], as it decomposes the original large-scale problem into smaller subproblems, each of which typically reduces to a simple projection or a small-scale QP, thereby substantially improving optimization efficiency.

Effectiveness Analysis About $\mathbf{w}_u$. In a standard FR framework, it needs to learn a unique user embedding $\mathbf{p}_u$ for each client u and a shared item embedding table $\mathbf{Q}$ for all clients. However, this approach may yield suboptimal performance in heterogeneous settings, where data distributions vary significantly across clients. We assume that according to the local client's data distribution, each client has a ground-truth user embedding $\mathbf{p}_u^*$ and a ground-truth item embedding table $\mathbf{Q}_u^*$. Therefore, the loss between these actual distributions and the predictions is as follows:

$$\min_{\{\mathbf{p}_u\},\mathbf{Q}} = \frac{1}{n}\sum_{u=1}^{n} \left\| \mathbf{p}_u^T\mathbf{Q} - \mathbf{p}_u^{*T}\mathbf{Q}_u^* \right\|_2^2, \tag{9}$$

where $\mathbf{Q}$ is shared across all clients. Our method can utilize the composite aggregation to learn personalized item embedding $\mathbf{Q}_u$ for each client. Hence, the expected loss takes the following form:

$$\min_{\{\mathbf{p}_u,\mathbf{Q}_u\}} = \frac{1}{n}\sum_{u=1}^{n} \left\| \mathbf{p}_u^T\mathbf{Q}_u - \mathbf{p}_u^{*T}\mathbf{Q}_u^* \right\|_2^2, \tag{10}$$

where $\mathbf{Q}_u = \sum_{v=1}^{n} w_{uv}\mathbf{Q}_v$ is the personalized item embeddings by applying composite aggregation for each client. Clearly, we can deduce that $(\tilde{\mathbf{p}}_u, \tilde{\mathbf{Q}})$ is a global optimum of Equation (9) if and only if $\tilde{\mathbf{p}}_\mathbf{u}^T\tilde{\mathbf{Q}} = \mathbf{p}_u^{*T}(\frac{1}{n}\sum_{u=1}^{n}\mathbf{Q}_u^*)$. Besides, if we denote $(\hat{\mathbf{p}}_u, \{\hat{\mathbf{Q}}_u\}_n)$ as a global optimum of Equation (10), then it follows that $\hat{\mathbf{p}_u}^T\hat{\mathbf{Q}}_u = \mathbf{p}_u^{*T}\mathbf{Q}_u^*$ for each client u. Thus, our model finds an exact solution with zero global loss, whereas Equation (9) has a global loss Δ:

$$\Delta = \frac{1}{n}\sum_{u=1}^{n} \left\| \frac{1}{n}\mathbf{p}_u^{*T}\sum_{v=1}^{n}\left(\mathbf{Q}_v^* - \mathbf{Q}_u^*\right) \right\|_2^2, \tag{11}$$

where Δ increases with the heterogeneity of $\mathbf{Q}_u^*$. Moreover, since our formulation provides n matrix equations, we can fully recover the column space of $\mathbf{p}_u^*$ as long as $\mathbf{Q}_u^*$'s span $\mathbb{R}^f$. Conversely, solving Equation (9) yields only one matrix equation, so it fails to recover $\mathbf{p}_u^*$ for any $f > 1$. Due to the unique nature of FR tasks, combining similarity and complementarity allows for optimizing embedding tables that better reflect the true local distribution. Thus, by introducing the composite aggregation weight $\mathbf{w}_u$, it can achieve lossless personalized federated optimization.

By introducing both similarity and complementarity metrics into the overall aggregation framework, we can not only aggregate similar item embeddings but also aggregate complementary ones, thereby alleviating the specific embedding skew issue when aggregating item embedding tables in FR tasks. Notably, it can ensure consistency in conditional distributions $p(y|x)$ while preserving complementarity in marginal distributions $p(x)$ to better model heterogeneity by the joint distributions $p(x, y)$.

5.1.2 Local Training. The mission of each client u is to utilize local data to optimize the local empirical loss $\mathcal{L}_u$ regarding private user embedding $\mathbf{p}_u$ and personalized item embedding $\mathbf{Q}_u$. The private user embedding $\mathbf{p}_u$ is kept locally, while the computed item embedding $\mathbf{Q}_u$ is uploaded to the server for global aggregation. To mine the information from interactions during training, we specify $\mathcal{L}_u$ as the **Binary Cross-Entropy (BCE)** loss, which is a well-designed objective function for recommender systems. Formally, the objective function of BCE loss is defined as

$$\mathcal{L}_u = -\sum_{(u,i)\in\mathcal{D}_u} r_{ui}\log\hat{r}_{ui} + (1 - r_{ui})\log(1 - \hat{r}_{ui}), \tag{12}$$

where $\mathcal{D}_u = \mathcal{D}_u^+ \cup \mathcal{D}_u^-$ and $\mathcal{D}_u^+$ represents observed interactions, i.e., $r_{ui} = 1$, and $\mathcal{D}_u^-$ represents uniformly sampled negative instances, i.e., $r_{ui} = 0$. Note that unlike federated vision tasks, which require the proximal term to restrict personalized models to be closer to the global model, i.e., $\| \mathbf{Q}_u - \mathbf{Q}_u^g \|^2$ in FedProx [29], pFedGraph [52], and so on, where $\mathbf{Q}_u^g$ denotes the global model, FR tasks inherently involve great heterogeneity and strong requirements for personalization. Hence, we solely use task-driven loss $\mathcal{L}_u$ without additional terms to keep the localization properties of item embedding. By optimizing the BCE loss in the local client, we can update the user embedding $\mathbf{p}_u$ and $\mathbf{Q}_u$ by stochastic gradient descent as follows

$$\mathbf{p}_u = \mathbf{p}_u - \eta \cdot \frac{\partial\mathcal{L}_u}{\partial\mathbf{p}_u}, \quad \mathbf{Q}_u = \mathbf{Q}_u - \eta \cdot \frac{\partial\mathcal{L}_u}{\partial\mathbf{Q}_u}, \tag{13}$$

where η is the learning rate. At the end of local training in each round, clients upload $\mathbf{Q}_u$ to the server for global aggregation.

5.1.3 Local Inference. During the local inference stage, client u first downloads the aggregated item embeddings $\mathbf{Q}_u^g = \sum_{v=1}^n w_{uv}\mathbf{Q}_v$ from the server. Notably, in federated vision domains, it can directly perform local inference using shared global parameters $\mathbf{Q}^g$. However, in FR tasks, the existence of client-specific user embedding $\mathbf{p}_u$ introduces a spatial misalignment issue between the user embedding $\mathbf{p}_u^{t-1}$ at previous round $t-1$ and the aggregated item embedding $\mathbf{Q}_u^{g(t)}$ at this round t. To achieve space alignment, we employ a simple yet effective interpolation method to narrow the gap between local-specific parameters $\mathbf{p}_u$ and global parameters $\mathbf{Q}_u^g$, i.e., $\mathbf{Q}_u^{g(t)} = \rho\mathbf{Q}_u^{g(t-1)} + (1-\rho)\mathbf{Q}_u^{g(t)}$ where ρ controls the weight of the local parameters in the current round. By introducing ρ, we balance the weight of local parameters $\mathbf{Q}_u$ and global aggregated parameters $\mathbf{Q}_u^g$, thereby aligning items with users in the embedding space. After obtaining the item embedding $\mathbf{q}_u^i \in \mathbf{Q}_u$ for each item i, we can perform local inference with $\mathbf{p}_u$ at the local client u, which is $\hat{r}_{ui} = f\left(\mathbf{p}_u, \mathbf{q}_u^i\right)$, where $f(\cdot)$ denotes the inner product or neural match function [19] to compute similarities between user

Algorithm 1: FedCA Executive Process

```
Input: local models: $\mathbf{p}_u, \mathrm{Q}_u, \mathbf{w}_u$; global rounds: $T$; local epochs: $E$;
learning rate: $\eta$; selected clients: $\mathcal{U}_s$;
Output: local models: $\mathbf{p}_u, \mathrm{Q}_u$ at clients and $\mathbf{w}_u$ at the server;
Server executes:
Initialize global item embeddings $\{\mathrm{Q}_u^g\}_{u=1}^n$;
for each round $t = 1, 2, \cdots, T$ do
    Sends $\mathrm{Q}_u^{g(t)} = \sum_{v=1}^n w_{uv}\mathrm{Q}_v^t$ to each client $u$;
    for each client $u \in \mathcal{U}_s$ in parallel do
        $\mathrm{Q}_u^{t+1} \longleftarrow$ LocalTraining($\mathrm{Q}_u^{g(t)}$, $u$);
    end for
    $\mathbf{s}_u \longleftarrow$ compute similarity with Eq.(6);
    $\mathbf{c}_u \longleftarrow$ compute complementarity with Eq.(7);
    $\mathbf{w}_u \longleftarrow$ optimize with Eq.(5) for each client $u$;
end for
LocalTraining($\mathrm{Q}_u$,u):
for each local epoch $e = 1, 2, \cdots, E$ do
    for each batch in $\mathcal{D}_u$ do
        compute local loss $\mathcal{L}_u$ by following Eq.(12);
        update $\mathbf{p}_u$ and $\mathrm{Q}_u$ with Eq.(13);
    end for
end for
return $\mathrm{Q}_u$
```

u and item i. By aggregating both similar and complementary clients, our model can enhance the prediction accuracy for future items. We present the FedCA algorithm in detail in Algorithm 1.

5.2 More Discussion About FedCA

Considering the unique characteristics of FR tasks, we propose a unified composite aggregation framework to finely aggregate embedding tables from an optimization perspective. This framework flexibly integrates constraint terms that measure model similarity $\mathcal{F}_s$ and data complementarity $\mathcal{F}_c$, building on the task-specific loss $\mathcal{L}_u$ for client u. This approach alleviates the embedding skew issue caused by solely using similarity aggregation. Specifically, our proposed composite aggregation framework is defined as follows:

$$
\begin{aligned}
\min_{\{\mathbf{p}_u, \mathrm{Q}_u, \mathbf{w}_u\}} \sum_{u=1}^{n} \Bigg(& \mathcal{L}_u\left(\mathbf{p}_u, \mathrm{Q}_u; \mathcal{D}_u\right) + \alpha \sum_{v=1}^{n} \mathcal{F}_s\left(w_{uv}; \mathrm{Q}_u; \mathrm{Q}_v\right) \\
& + \beta \sum_{v=1}^{n} \mathcal{F}_c\left(w_{uv}; \mathcal{D}_u; \mathcal{D}_v\right) \Bigg) \\
\text{s.t.} \quad & \mathbf{1}^T \mathbf{w}_u = 1, \quad \mathbf{w}_u \geq \mathbf{0}.
\end{aligned}
\tag{14}
$$

Note that our method is a model-agnostic, plug-and-play aggregation mechanism that can be seamlessly integrated into the server aggregation process of mainstream FR tasks. Regarding *flexibility*, we can customize the specific loss for model similarity $\mathcal{F}_s$ and data complementarity $\mathcal{F}_c$, although we use squared loss in this work. As for model similarity $\mathcal{F}_s$, further research could explore more potential model representations, such as dimensionality reduction techniques. For data complementarity $\mathcal{F}_c$, future work could consider advanced complementarity metrics that better align with the local data distribution. Regarding *versatility*, we can adaptively adjust the values of α and β to implement mainstream aggregation mechanisms, such as weighted aggregation ($\alpha = 0, \beta = 0$), cluster-based aggregation ($\alpha \neq 0, \beta = 0$), and active aggregation ($\alpha = 0, \beta \neq 0$). Hence, our method specifically summarizes a unified aggregation mechanism tailored for FR tasks.

Relations with Classic Aggregation Mechanisms. In this section, we will analyze the compatibility of our proposed FedCA model. We introduce a unified optimization framework for aggregating item embeddings in FR tasks. This framework can transform into several classical aggregation methods by flexibly adjusting hyperparameters α and β, as well as the proxy coefficient p_u. Specifically, when $\alpha = 0$ and $\beta = 0$, and the proxy coefficient p_u is set to the mean, our method degrades to the average aggregation method used in FCF [1]. When $\alpha = 0$ and $\beta = 0$, and p_u is set to the relative dataset size, our method achieves the weighted aggregation used in FedAvg [38]. When $\alpha = 0$ and $\beta = 0$, and p_u is set to the degree of difference between local and global models, our method can degrade to the FedAtt method [22]. When $\alpha \neq 0$ and $\beta = 0$, it can become the similarity-based aggregation method pFedGraph [52]. Specifically, if only the most similar clients are selected for each client, it is equivalent to the clustering aggregation in PerFedRec [35]. When $\alpha = 0$ and $\beta \neq 0$, it implies aggregating only dissimilar parameters for each client, which is equivalent to the FedFast [40], where clients are first clustered, and then clients from each cluster are aggregated proportionally. We conclude that our method can flexibly implement several aggregation methods.

Computation Complexity Analysis. The computation complexity of FR models primarily consists of client-side and server-side components. Specifically, the complexity on the client side mainly includes the local training process and the SVD process for privacy enhancement. The local training process has a time complexity of $O(mf)$, while the privacy enhancement process involves the SVD of $\mathbf{Q}_u^s \in \mathbb{R}^{|\mathcal{I}_u| \times f}$, with a complexity of $O(|\mathcal{I}_u| f^2)$, where $|\mathcal{I}_u|$ is the number of items interacted with by client u, and f is the embedding dimensions. It is worth noting that $|\mathcal{I}_u| \ll m$ and $f = 16$ in this work. Therefore, the computational complexity on the client side is comparable to that of classical FR models. The complexity on the server side mainly includes the aggregation process and the optimization process. The aggregation process is a necessary step for any FR algorithm and has a complexity of $O(nmf)$. The optimization process primarily involves computing pairwise similarity and complementarity, with a complexity of $O(n^2)$. This complexity is consistent with that of mainstream similarity-based aggregation methods. We can leverage approximate nearest neighbor search techniques to significantly improve the computation efficiency of similarity and complementarity matrix calculations. Besides, since this process can independently solve for each client, advanced parallel algorithms can reduce the complexity to $O(n \log n)$, or even as low as $O(n)$. This is feasible for computationally resource-rich servers, making it suitable for large-scale deployment scenarios.

Privacy Discussion. Our FedCA approach maintains the same privacy protection standards as the baseline models, e.g., FCF [1], PerFedRec [35], and PFedRec [56] since it protects user privacy by keeping users' original interaction records local and ensuring that private user embeddings do not interact with unauthorized third parties. Besides, since our composite aggregation mechanism is model-agnostic, it can seamlessly integrate with other privacy-enhanced FR models like FedRec [33], and can easily incorporate various privacy protection strategies, such as differential privacy [11], to further enhance user privacy guarantees.

Furthermore, we analyze the privacy protection capabilities of the proposed method from the dimensions of user data, user embeddings, and item embeddings. (1) Our method follows the standard setup of FRs and thus maintains the same privacy protection standards as baseline models, e.g., FCF and PerFedRec. Specifically, it protects user privacy by keeping users' original interaction

Table 1. Statistics for the Datasets Used in the Evaluation

Datasets	# Clients	# Items	# Ratings	# Avg.	Density
ML-100K	943	1,682	1,00,000	106	6.3%
Filmtrust	1,508	2,071	35,497	24	1.14%
ML-1M	6,040	3,952	10,00,209	166	4.19%
MC-100K	1,00,000	19,738	7,19,405	30	0.04%

records local and ensuring that private user embeddings do not interact with third parties. (2) Our composite aggregation mechanism can enhance privacy protection of item embeddings by introducing complementarity. Specifically, by incorporating complementary items from other clients into item embeddings of the current client, it serves equivalent roles to pseudo-items, thereby enhancing privacy protection of item embeddings, which is similar to FedRec and FedRec++. (3) We introduced differential privacy into transmitted item embeddings of our method, with theoretically guaranteed privacy of transmitted parameters, which is consistent with PFedRec and FedRAP. In summary, our method considers comprehensive privacy protection regarding user local data, user embeddings, and item embeddings.

6 Experiments

In this section, we provide detailed experimental settings and comprehensive experimental results.

6.1 Experimental Settings

Datasets. We evaluate our model on four benchmark datasets with varying client scales: **Movielens-100K (ML-100K)** [18], Filmtrust [14], **Movielens-1M (ML-1M)** [18], and **Microlens-100K (MC-100K)** [42]. The first three datasets are for movie recommendation with explicit feedback, where ratings greater than 0 are converted to 1. The last dataset is for short video recommendation with implicit feedback. Each user is treated as an independent client, and each client's data inherently exhibits great heterogeneity. The details about the used datasets are listed in Table 1, where # Avg. represents the averaged number of interacted items by each client.

Baselines. To thoroughly explore the effectiveness of various aggregation mechanisms, we compared eight classic federated models: (1) *Local:* local training without federated aggregation. (2) *FCF:* [1] averaged aggregation by allocating equal weights to each client. (3) *FedAvg:* [38] weighted aggregation by the relative size of local client data. (4) *PerFedRec:* [35] clustering aggregation by grouping clients into several clusters with model similarities. (5) *FedAtt:* [22] attentive aggregation by minimizing the weighted distance between global and local models. (6) *FedFast:* [40] active aggregation by identifying representatives from different clusters. (7) *pFedGraph:* [52] graph aggregation by learning the similarities between individuals. (8) *PFedRec:* [56] recent personalized FR model, achieved through dual personalization of score function and item embedding.

Implementations. Following previous works [56], we randomly sample $N = 4$ negative instances for each positive sample and utilize the leave-one-out strategy for efficient validation. Besides, we filter out the users with fewer than 10 interactions. For our model, we set $k = 4$ and choose the optimal values for hyperparameters on four datasets. Besides, we utilize two common evaluation metrics for item ranking tasks: HR@K and NDCG@K where $K = 10$. We conduct hyperparameter tuning for all compared models and report the results as the average of five repeated experiments. To validate the model agnostics of our method, we verify its effectiveness on three canonical backbones, PMF [39], NCF [19], and SASRec [25].

As for the settings of main experiments, we set the global rounds $T = 100$, local epochs $E = 10$, batch size $B = 256$, and learning rate $\eta = 0.01$ for all methods for fair comparison. We set the

Table 2. Comparison Results of FedCA and Other Baselines Evaluated on Four Commonly Used Datasets

Backbones	Models	ML-100K		Filmtrust		ML-1M		MC-100K	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
PMF	Local	0.4128	0.2203	0.4760	0.2410	0.4264	0.2314	0.1246	0.0567
	FCF	0.4327	0.2497	0.6407	0.4914	0.4454	0.2484	0.1294	0.0594
	FedAvg	0.4878	0.2786	0.6517	0.5126	0.4912	0.2751	0.1295	0.0601
	PerFedRec	0.4973	0.2797	0.6577	0.5247	0.4623	0.2622	0.1255	0.0599
	FedAtt	0.4645	0.2558	0.6088	0.3359	0.4310	0.2168	0.0982	0.0445
	FedFast	0.4984	0.2747	0.6527	0.5191	0.5061	0.2898	0.1278	0.0600
	pFedGraph	0.5928	0.4025	0.6961	0.5430	0.7904	0.6347	0.1324	0.0605
	PFedRec	0.7254	0.4648	0.7096	0.5629	0.8032	0.6519	0.1334	0.0621
	FedCA	**0.8738**	**0.7597**	**0.7725**	**0.5945**	**0.8348**	**0.7118**	**0.1351**	**0.0678**
NCF	Local	0.4077	0.2145	0.4312	0.2485	0.3881	0.1839	0.1004	0.0459
	FCF	0.4115	0.2390	0.6477	0.4968	0.4269	0.2232	0.1290	0.0668
	FedAvg	0.4478	0.2731	0.6507	0.4969	0.4899	0.2703	0.1397	0.0674
	PerFedRec	0.4135	0.2253	0.3752	0.1418	0.4219	0.2093	0.1128	0.0591
	FedAtt	0.4910	0.2626	0.6547	0.4801	0.4136	0.2177	0.1375	0.0669
	FedFast	0.4436	0.2708	0.6632	0.5007	0.4040	0.2008	0.1402	0.0774
	pFedGraph	0.5822	0.3587	0.6718	0.5021	0.5113	0.2992	0.1416	0.0669
	PFedRec	0.6931	0.5031	0.6732	0.5031	0.6826	0.4041	0.1422	0.0687
	FedCA	**0.8452**	**0.7444**	**0.6836**	**0.5099**	**0.7815**	**0.6662**	**0.1465**	**0.0782**

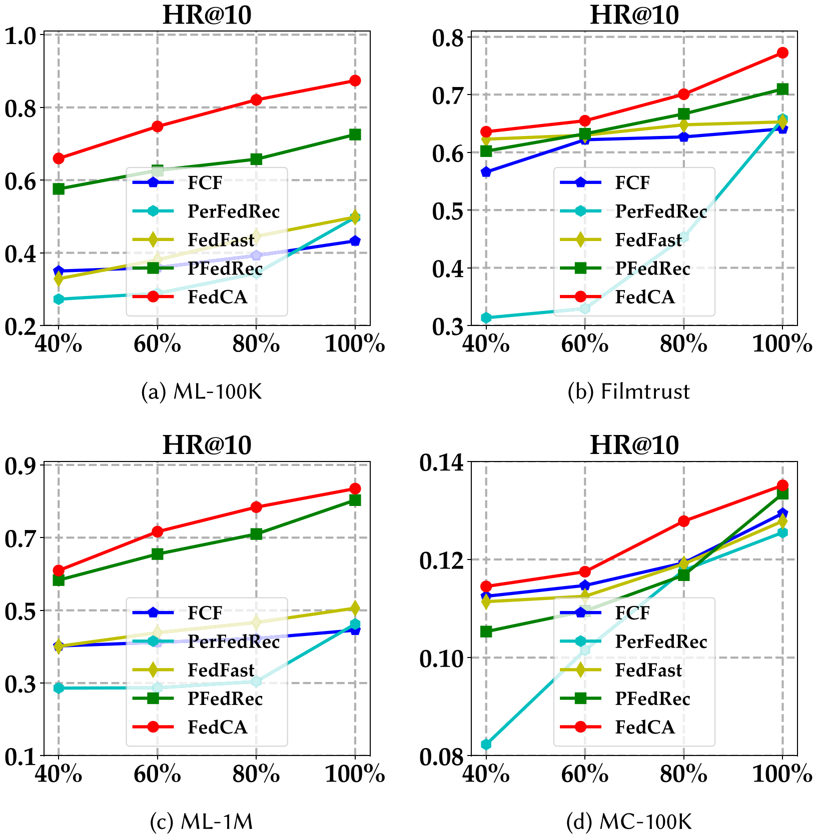
Higher values indicate better performance. The best results are in bold.

embedding dimensions $f = 16$ for users and items. We set the client aggregation ratio $r = 60\%$ for all compared models. We use Adam optimizer with default parameters for local training. In this work, we use uniform distribution to initialize $\mathbf{Q}_u$ and $\mathbf{p}_u$ at the beginning of the server aggregation and local training. For clustering methods such as PerFedRec and FedFast, we use the default parameters of k-means and specify the number of clusters as 10. In NCF, the specific structure of the **Multi-Layer Perception (MLP)** layers is $[32, 16, 8]$. During the optimization of MLP parameters, we applied the weight decay technique with $\lambda = 0.001$. All experiments are conducted on a machine with four RTX A5000 GPUs. For reproducibility, we release our source code and utilized data at https://github.com/hongleizhang/FedCA.

6.2 Experimental Results

This section introduces the effectiveness of our method through various experiments, including overall performance, analyses with different ratios of training data, and visualization results.

Overall Performance. Table 2 presents the results of our model compared to baselines using two backbones, evaluated in terms of HR@10 and NDCG@10 across four datasets. From the experimental results, we can observe that: (1) compared to local training, general FL methods (FedAvg, FedAtt, and pFedGraph) and FR models (FCF, PerFedRec, FedFast, and PFedRec) demonstrate better predictive performance, indicating the effectiveness of various aggregation mechanisms in federated settings. (2) By comparing different aggregation mechanisms, it can be noticed that both similarity-based aggregation (PerFedRec, FedAtt, and pFedGraph), and dissimilarity-based aggregation (FedFast) can achieve effective knowledge aggregation in federated settings. This suggests the motivation of our model to combine similarity and complementarity. (3) Our method outperforms other baseline models, indicating that our composite aggregation, compared to solely using similarity for aggregation as borrowed from federated vision, is more suitable for aggregating embedding tables in FR tasks.



(a) ML-100K

(b) Filmtrust

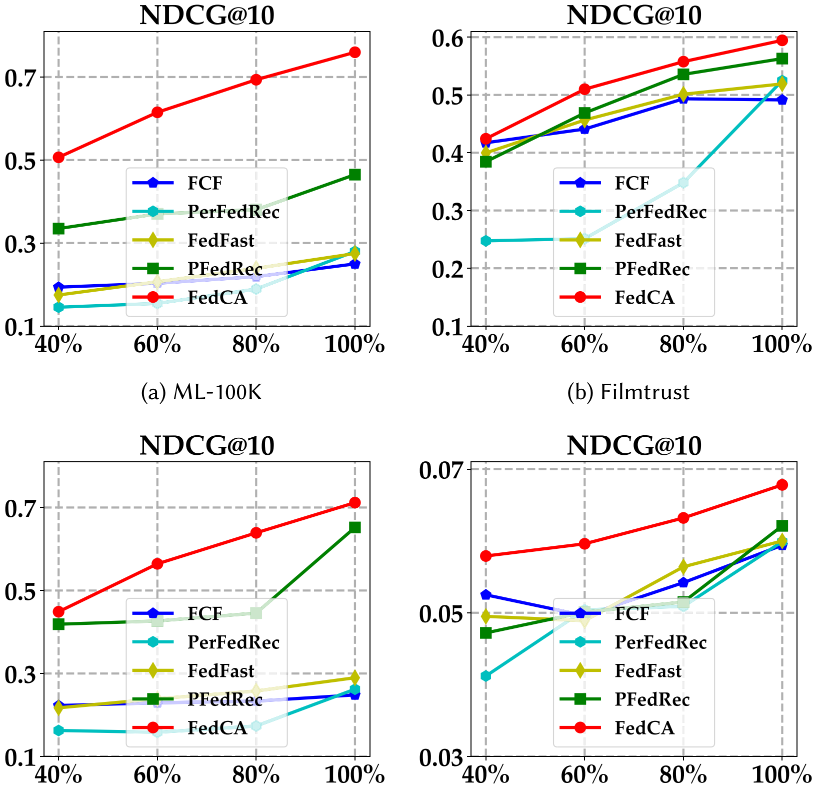
(c) ML-1M

(d) MC-100K

Fig. 5. HR@10 results comparing our FedCA with FR baselines under varying train data ratios.

Robustness to Varying Sparsity of Train Data. Recall from the empirical analysis in Section 3 that solely using similarity for aggregating embedding tables in FRs can lead to embedding skew issue. This means that as the aggregation process, those already trained embeddings improve while untrained ones remain random or degrade, ultimately failing to make predictions on future items. Our composite aggregation aims to alleviate this problem in FRs by combining similarity and complementarity to enhance untrained embeddings. Hence, theoretically, even with limited training data, our method can still achieve good generalization on test sets. Note that in federated vision domains, each client can flexibly partition local data based on different label distributions to reflect varying heterogeneity. This area has been extensively studied [29, 52]. However, in FR tasks, each user naturally represents a single client, so each client's local data are fixed. Thus, the data partitioning methods mentioned above cannot be directly used in FRs. Hence, we use the sparsity of local data to reflect heterogeneity in FRs, with the assumption that the sparser the local data, the lower the probability of item overlap between clients, resulting in greater heterogeneity.

To explore the efficacy of our model, we evaluate the robustness of four FR methods instantiated on PMF backbone under different train data sparsity (40%, 60%, 80%, and 100%). The experimental results on HR@10 and NDCG@10 are shown in Figures 5 and 6. The results suggest that our method consistently outperforms other baselines under different levels of training data sparsity, directly demonstrating the effectiveness of our composite aggregation mechanism for aggregating embedding tables in FRs. It also verifies the superiority of our approach in mitigating heterogeneity.



(a) ML-100K

(b) Filmtrust

(c) ML-1M

(d) MC-100K

Fig. 6. NDCG@10 results comparing FedCA with FR baselines under varying train data ratios.

Specifically, when the sparsity of train data is at 40%, our model greatly outperforms the baselines on the ML-100K and MC-100K datasets. This indicates that combining similarity and complementarity for aggregating embedding tables is highly effective for FR tasks, especially when training data are very limited. Besides, we observed that the cluster-based aggregation method (PerFedRec) performs the worst under sparse data conditions (40%) and exhibits very unstable learning process during the training iterations. This is primarily because existing clustering methods (such as k-means [3]) require careful selection of the number of clusters, as different datasets have varying client scales. Moreover, with limited data, it is challenging to accurately measure the similarity of each client, ultimately leading to the failure of cluster-based aggregation methods. This finding is consistent with the very recent work [20]. In contrast, our method formulates the process into a unified optimization loss to smoothly select more similar clients, effectively achieving the benefits of cluster-based aggregation without the need for manual parameter tuning.

In FR tasks, each user naturally represents a single client, so each client's local data are inherently fixed. Moreover, the items interacted with by each client differ substantially, giving rise to inherent heterogeneity. Thus, the data partitioning methods with artificial manner in federated vison tasks [29, 52] cannot be directly used in FRs. Hence, we use the sparsity of local data to reflect heterogeneity in FRs, with the assumption that the sparser the local data, the lower the probability of item overlap between clients, resulting in greater heterogeneity. Different forms of heterogeneity also reflect distinct user behavior patterns. We have shown the robustness of FedCA

Table 3. Results of FedCA and Baselines with Varying Heterogeneity across Four Datasets

$\tau = 0.3$	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Local	0.2014	0.1006	0.2166	0.1058	0.1933	0.1046	0.0915	0.0412
FCF	0.3075	0.1709	0.2232	0.1163	0.3163	0.1849	0.1032	0.0448
PerFedRec	0.2813	0.1613	0.2327	0.1244	0.3596	0.2588	0.1094	0.0489
FedFast	0.3107	0.1833	0.2314	0.1236	0.3309	0.2401	0.1046	0.0462
PFedRec	0.5034	0.3894	0.2592	0.1283	0.3811	0.2902	0.1096	0.0498
FedCA	**0.5695**	**0.4281**	**0.2914**	**0.1397**	**0.4883**	**0.3529**	**0.1125**	**0.0505**
$\tau = 0.2$	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Local	0.1987	0.0934	0.1577	0.0688	0.1831	0.1019	0.0872	0.0325
FCF	0.2704	0.1433	0.2064	0.1019	0.2219	0.1452	0.0962	0.0395
PerFedRec	0.2124	0.1254	**0.2176**	**0.1120**	0.2837	0.2013	0.1075	0.0483
FedFast	0.2969	0.1547	0.2063	0.1025	0.2778	0.1977	0.0978	0.0402
PFedRec	0.3538	0.2192	0.2088	0.1056	0.2933	0.2182	0.0988	0.0445
FedCA	**0.4625**	**0.3189**	0.2095	0.1074	**0.3511**	**0.2557**	**0.1086**	**0.0497**
$\tau = 0.1$	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Local	0.1801	0.0898	0.1297	0.0564	0.1738	0.0947	0.0638	0.0287
FCF	0.1495	0.0685	0.1457	0.0642	0.1593	0.0781	0.0583	0.0202
PerFedRec	0.1507	0.0710	0.1566	0.0794	0.1684	0.0832	0.0598	0.0225
FedFast	0.2206	0.1073	0.1547	0.0766	0.1836	0.1049	**0.0814**	**0.0399**
PFedRec	0.1738	0.1002	0.1552	0.0781	0.1930	0.1182	0.0694	0.0377
FedCA	**0.2948**	**0.1605**	**0.1682**	**0.0801**	**0.2453**	**0.1407**	0.0763	0.0382

The best results are in bold, and the second results are underlined.

under heterogeneity levels 1.0, 0.8, 0.6, and 0.4 in Figures 5 and 6. To further validate performance under varying heterogeneity, we added the experiments when heterogeneity levels are 0.3, 0.2, and 0.1 in Table 3. Our model achieves superior results even in extremely heterogeneous conditions, thus further verifying the effectiveness of our composite aggregation.

Compatibility on Sequential Recommendation Task. Our approach is applicable to any recommendation task equipped with embedding tables, making it suitable for a wider range of complex scenarios. To evaluate the classical aggregation methods compared in this work, we adopt SASRec as new backbone, a widely used baseline in sequential recommendation, aiming to validate the effectiveness of our proposed composite aggregation mechanism in more advanced and challenging recommendation settings. As shown in Table 4, aggregation-based methods (FCF, PerFedRec, FedFast, PFedRec, and FedCA) consistently outperform local training, demonstrating the effectiveness of aggregation in sequential recommendation tasks. Furthermore, our method surpasses similarity-based aggregation approaches (FCF, PerFedRec, FedFast, and PFedRec), indicating that incorporating data complementarity enables more effective aggregation of embedding tables in recommendation tasks. Notably, the NDCG metric in sequential recommendation tasks exceeds that in traditional tasks, highlighting the advantage of modeling sequential patterns.

Visualizing Composite Aggregation. Figure 7 presents the visualization results of the similarity matrix $\mathbf{S}$, the complementarity matrix $\mathbf{C}$, and the composite aggregation weights $\mathbf{W}$, reflecting their mutual influence in the overall loss function in Equation (4). We randomly select 10

Table 4. Comparison Results of FedCA with SASRec Backbone Evaluated on Four Commonly Used Datasets

Backbones	Models	ML-100K		Filmtrust		ML-1M		MC-100K	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
SASRec	Local	0.1743	0.1076	0.4581	0.2359	0.1284	0.0818	0.0431	0.0238
	FCF	0.2047	0.1337	0.6387	0.5430	0.1495	0.1022	0.0843	0.0487
	PerFedRec	0.4821	0.2698	0.6381	0.5255	0.3026	0.1894	0.0973	0.0528
	FedFast	0.5032	0.2894	0.6483	0.5297	0.4421	0.3274	0.0946	0.0521
	PFedRec	0.6582	0.5519	0.7287	0.5782	0.5182	0.4371	0.1028	0.0629
	FedCA	**0.6783**	**0.5893**	**0.7539**	**0.6021**	**0.5633**	**0.4576**	**0.1156**	**0.0652**

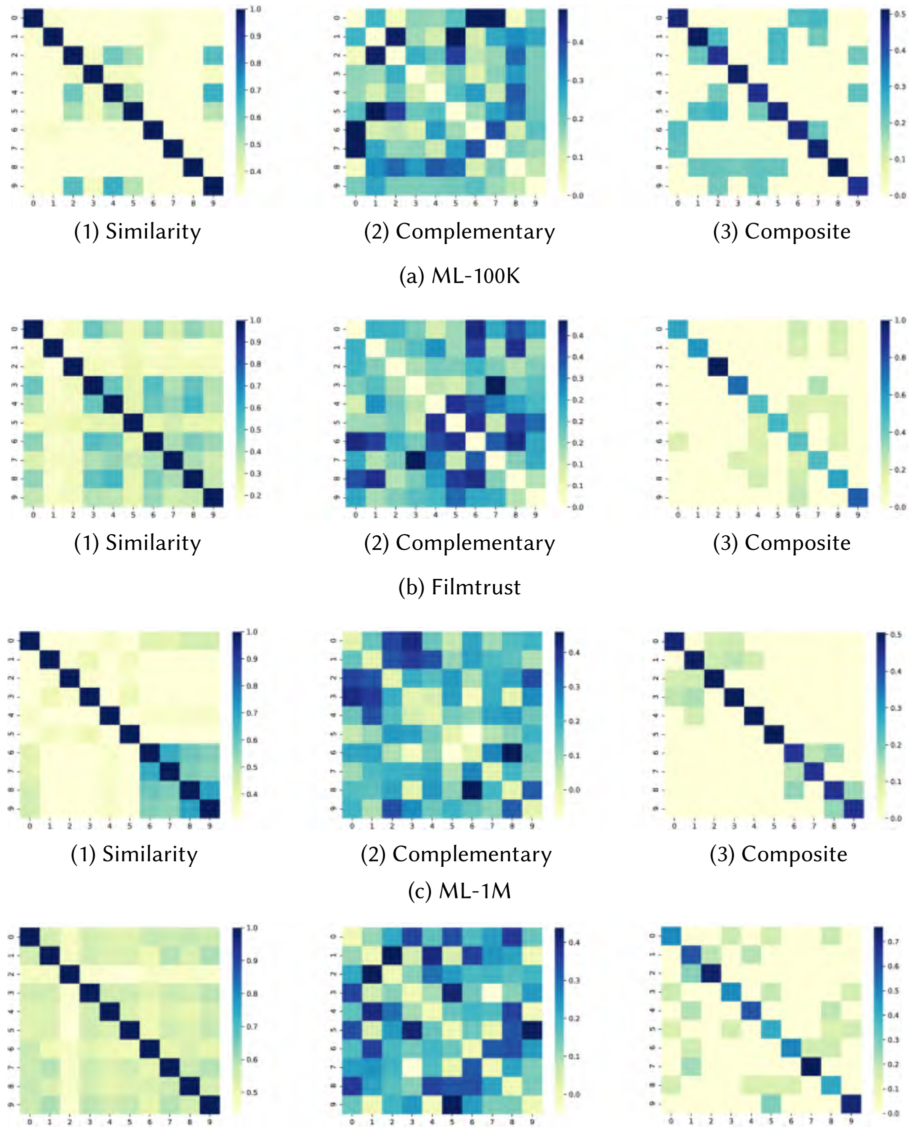
The best results are in bold.

clients for this demonstration. From the results, it can be observed that the composite aggregation weight effectively balances similarity and complementarity, accommodating both the model similarity and data complementarity among clients. It tends to favor clients with both high similarity and complementarity, where similarity ensures the consistency of the embedding distribution for interacted items among clients, and complementarity enhances the embeddings of non-interacted items from other clients, which is similar to the classical user-based collaborative filtering [13].

Privacy Enhancement. To further improve the privacy capability of our proposed model, we explored enhancing our method with **Local Differential Privacy (LDP)** [11]. Specifically, we apply Laplace noise to the item embeddings $\mathbf{Q}_u$ and set the noise strength from 0.1 to 0.4 with an interval of 0.1. As shown in Table 5, performance degrades as the noise strength δ increases, while the performance drop is slight when δ is small. Hence, a moderate noise strength, e.g., $\delta = 0.1$, is desirable to achieve a good tradeoff between model performance and privacy protection.

To further enhance the privacy of our model, we apply LDP not only to the original item embedding parameters but also to the SVD-derived item embeddings, in order to examine its impact on the final recommendation performance. As shown in Table 6, adding LDP to the SVD-derived item embeddings, with the noise intensity controlled by δ', makes the parameters more sensitive to perturbations compared to the original item embeddings. With increasing noise intensity, the privacy protection becomes stronger, but the recommendation performance degrades more severely. Therefore, to balance privacy preservation and recommendation performance, we set $\delta' = 0.05$ for enhancing the SVD-derived item embeddings, thereby improving both the privacy and effectiveness of the proposed method.

Communication Overhead Analysis. We first analyzed the communication overhead theoretically. For the FCF, the download and upload communication overhead is $O(mf)$, where m is the number of items and f is the embedding dimension. PerFedRec requires downloading the extra clustered model resulting in an overhead of $O(2mf)$. The upload overhead is $O(mf + |\mathcal{I}_u|f)$ since it requires uploading user profiles. FedFast has a download overhead of $O(mf)$ and an upload overhead of $O(mf + |\mathcal{I}_u|f)$. The overhead of PFedRec matches that of FCF. Our FedCA has a download overhead of $O(mf)$ and an upload overhead of $O(mf + |\mathcal{I}_u|k)$, where k is the number of singular values, typically is 4. Note that the parameters transmitted in each round are consistent, resulting in the same overhead across different communication rounds. As suggested, we compared the communication overheads (downloads and uploads) across four varying-scale datasets, on a machine with 64-bit floating point precision. The results in Table 7 show that our download overhead is equal to the FCF, and our upload overhead is comparable to that of the baselines.



(1) Similarity (2) Complementary (3) Composite

(a) ML-100K

(1) Similarity (2) Complementary (3) Composite

(b) Filmtrust

(1) Similarity (2) Complementary (3) Composite

(c) ML-1M

(1) Similarity (2) Complementary (3) Composite

(d) MC-100K

Fig. 7. Visualization results regarding similarity, complementarity, and composite aggregation weights.

Time Complexity Analysis. The time overhead of our proposed method primarily involves local training, SVD decomposition, similarity computation, complementarity computation, and optimization solving w_u. As shown in Equation (7), our method indeed introduces additional steps to enhance privacy for $\mathbf{Q}_u^s$ on the client side, yet the time complexity is acceptable. Additionally, while our method introduces extra complementarity computation and optimization on the server side, these operations can be executed efficiently on computationally resource-rich servers. When a large number of clients are involved in server-side optimization, the server's substantial computational

Table 5. Results with Added Noises for LDP in Item Embeddings, Where δ Represents Noise Intensity

δ	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
0	0.8738	0.7597	0.7725	0.5945	0.8348	0.7118	0.1351	0.0678
0.1	0.8730	0.7597	0.7701	0.5912	0.8301	0.7054	0.1348	0.0665
0.2	0.8643	0.7467	0.7678	0.5884	0.8274	0.6976	0.1337	0.0656
0.3	0.8433	0.7220	0.7619	0.5832	0.8234	0.6834	0.1322	0.0643
0.4	0.8293	0.7115	0.7523	0.5741	0.8176	0.6739	0.1302	0.0631

Table 6. Results with Added Noises for LDP in SVD-Derived Item Embeddings, Where δ' Denotes Noise Intensity

δ'	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
0	0.8730	0.7597	0.7701	0.5912	0.8301	0.7054	0.1348	0.0665
0.05	0.8705	0.7546	0.7689	0.5903	0.8295	0.7027	0.1323	0.0648
0.1	0.8621	0.7474	0.7622	0.5834	0.8201	0.6933	0.1304	0.0633
0.15	0.8382	0.7201	0.7549	0.5798	0.8167	0.6798	0.1287	0.0927
0.2	0.8143	0.7045	0.7412	0.5648	0.8084	0.6593	0.1245	0.0895

Table 7. Comparison Results of Communication Overhead between FedCA and Baselines (Unit: KB)

Models	ML-100K		Filmtrust		ML-1M		MC-100K	
	Download ↓	Upload ↑	Download ↓	Upload ↑	Download ↓	Upload ↑	Download ↓	Upload ↑
FCF	**210.25**	**210.25**	**258.88**	**258.88**	**494.20**	**494.20**	**2,467.25**	**2,467.25**
PerFedRec	420.50	223.50	517.75	261.88	988.00	514.75	4,934.50	2,471.00
FedFast	210.25	223.50	258.88	261.88	494.20	514.75	2,467.25	2,471.00
PFedRec	210.25	210.25	258.88	258.88	494.20	494.20	2,467.25	2,467.25
FedCA	**210.25**	213.56	**258.88**	259.63	**494.20**	499.19	**2,467.25**	2,468.19

The best results are in bold, and the second results are underlined.

resources and efficient parallelization techniques can be leveraged to significantly reduce runtime. We present the specific run-time consumption at a single client, averaged over five runs on four datasets in Table 8, demonstrating that the overhead of the SVD step (0.02 s) is greatly less than that of local training (1.94 s) on ML-1M dataset. Moreover, the runtime of the optimization step (0.59 s) is considerably smaller than that of computing the similarity (0.86 s) and complementarity (0.88 s) matrices. Hence, in practice, this time complexity is acceptable. Besides, since local training and server-side optimization can be performed asynchronously, where the server computes the similarity and complementarity matrix from the previous round while clients conduct local training and SVD decomposition. This process can effectively improve time efficiency.

Ablation Study. In this section, we investigate the contributions of different components of the proposed method, including the proximal terms, the combination of different loss functions, and the use of various complementarity metrics, to demonstrate the validity of each component.

First, we explore the proximal term during local training. We explored the effectiveness of the proximal term in FR tasks, which is widely used in federated vision domains. Table 9 presents the experimental results with (w) and without (w/o) the proximal term based on the local task-specific

Table 8. Empirical Run-Time Results during Different Training Stages

Datasets	Local Training	SVD	Similarity	Complementarity	Optimization
ML-100K	0.53 s	0.005 s	0.24 s	0.26 s	0.13 s
Filmtrust	0.12 s	0.002 s	0.09 s	0.11 s	0.08 s
ML-1M	1.94 s	0.018 s	0.86 s	0.88 s	0.59 s
MC-100K	0.18 s	0.003 s	0.11 s	0.12 s	0.09 s

Table 9. Ablation Study Results of the Proximal Term in FR Tasks

Proximal	ML-100K		Filmtrust		ML-1M		MC-100K	
Term	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
w	0.8469	0.7274	0.7605	0.5744	0.8168	0.6801	0.1301	0.0577
w/o	**0.8738**	**0.7597**	**0.7725**	**0.5945**	**0.8348**	**0.7118**	**0.1351**	**0.0678**

The best results are in bold.

Table 10. Ablation Study Results for Using Different Loss Combinations

Loss Type	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
$\mathcal{L}_u$	0.4878	0.2786	0.6517	0.5126	0.4912	0.2751	0.1259	0.0534
$\mathcal{L}_u + \mathcal{F}_c$	0.8431	0.6983	0.7625	0.5752	0.8049	0.6585	0.1289	0.0606
$\mathcal{L}_u + \mathcal{F}_s$	0.8653	0.7457	0.7645	0.5791	0.8103	0.6987	0.1268	0.0565
$\mathcal{L}_u + \mathcal{F}_c + \mathcal{F}_s$	**0.8738**	**0.7597**	**0.7725**	**0.5945**	**0.8348**	**0.7118**	**0.1351**	**0.0678**

The best results are in bold.

loss in Equation (12). It can be observed that not utilizing the proximal term constraint which is effective in federated vision domains yields higher predictive performance in FR tasks. This indicates that recommendation tasks require stronger personalization at the local client level. Hence, it is not necessary to enforce the local model to be as similar as to the global model. Instead, the local model should be given the flexibility to develop a highly personalized representation that is tailored to the specific needs and preferences of each individual client. This approach not only enhances the recommendation accuracy but also aligns better with the inherent nature of recommendation tasks, which prioritize user-specific personalization over global consistency.

Second, we explore different combinations of loss functions to validate the effectiveness of the proposed composite aggregation mechanism. To validate the contributions of model similarity and data complementarity modules, we decompose the overall optimization loss in Equation (4) into three basic components: client-specific task loss $\mathcal{L}_u$, model similarity loss $\mathcal{F}_s$, and data complementarity loss $\mathcal{F}_c$. Since the server cannot access local data of clients, the client-specific loss is represented by the first squared loss term in Equation (5). From the experimental results shown in Table 10, we can verify the contribution of each proposed component and observe the following important conclusions.

Optimizing the aggregation process using only the $\mathcal{L}_u$ component is analogous to the weighted aggregation in FedAvg. This variant helps demonstrate the basic aggregation without additional constraints. Based on that, considering complementarity $\mathcal{F}_c$ and similarity $\mathcal{F}_s$ modules separately both improve model performance, indicating that both factors play important roles in the aggregation process. Given the importance of both complementarity and similarity, our proposed composite

Table 11. Ablation Study Results for Using Different Data Complementarity Metrics

Metrics	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
raw(·)	0.8739	0.7597	0.7729	0.5946	0.8350	0.7121	0.1353	0.0678
sin(·)	0.8536	0.7395	0.7496	0.5573	0.8135	0.6814	0.1239	0.0520
cos(·)	0.8738	0.7597	0.7725	0.5945	0.8348	0.7118	0.1351	0.0678

aggregation method takes a holistic approach by integrating these two factors within a unified optimization framework. By combining the similarity $\mathcal{F}_s$ and complementarity $\mathcal{F}_c$ modules, our method is able to leverage the strengths of both aspects simultaneously. This integration allows for a more nuanced and effective aggregation of embedding tables, which is particularly beneficial for FR tasks. As a result, the composite aggregation method significantly improves the effectiveness of aggregating embedding tables in FR tasks, leading to better recommendation accuracy.

Finally, we examine the impact of different data proxies for complementarity metrics on model performance. Specifically, we analyze the original operation raw(·), the sine operation sin(·), and the cosine operation cos(·) to identify the practical solution that balances data utility and privacy preservation. As shown in Figure 11, the original operation without any data proxy yields the best performance but directly exposes raw user representations, posing privacy risks. The sine operation protects privacy through nonlinear transformation of raw data but incurs significant performance degradation. In contrast, the cosine-based SVD operation achieves a favorable tradeoff between performance retention and privacy protection.

Sensitivity Analysis. In this section, we provide the sensitivity analysis of the proposed method to various hyperparameters, including the interpolation coefficient ρ, the proportion of participating clients r, and the coefficients controlling data similarity α and model complementarity β, in order to evaluate performance under different parameter settings.

First, we explore the effect of the interpolation method with varying interpolation coefficients ρ during local inference. Different interpolation coefficients reflect varying proportions of combining the local and aggregated models, thereby balancing personalization in local clients with collaborative information in the global server. From the results in Table 12, it can be seen that the local model achieved optimal performance when $\rho = 0.8$ and $\rho = 0.9$ on the ML-100K and ML-1M datasets, respectively. This suggests that during local inference, a balance should be struck between the global model at current round and the local model at last round. This balance helps mitigate the spatial misalignment issue caused by the client-specific user embedding $\mathbf{p}_u$ in FR tasks, which is the main difference compared to federated vision domain.

Second, we investigate the impact of varying client participation ratios on overall performance in FR, analyzing how the proposed method scales with the number of clients in FL systems, thereby providing insights for large-scale deployment. Specifically, Figure 8 illustrates the impact of varying client participation ratios r on recommendation performance. As the ratio increases, overall performance improves steadily. However, beyond a certain point, the gains become marginal. Considering the tradeoff between system communication efficiency and computational overhead, we set the client participation ratio to $r = 0.6$ to balance model performance and communication efficiency.

Finally, we examine the impact of the model similarity coefficient α and the data complementarity coefficient β on overall performance. The unified composite aggregation framework proposed in this work requires coordination between α and β to balance the importance of model similarity

Table 12. Results for the Interpolation Method on Two Utilized Datasets during Local Inference

ρ	ML-100K		Filmtrust		ML-1M		MC-100K	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
0.5	0.6681	0.4930	0.7056	0.5564	0.6833	0.4892	0.1251	0.0565
0.6	0.7031	0.5295	0.7156	0.5644	0.7028	0.5158	0.1262	0.0571
0.7	0.7434	0.5675	0.7335	0.5642	0.8025	0.6711	0.1262	0.0552
0.8	**0.8738**	**0.7595**	0.7515	0.5712	0.8278	0.6949	0.1284	0.0560
0.9	0.8611	0.7432	**0.7725**	**0.5945**	**0.8348**	**0.7118**	**0.1351**	**0.0678**
1.0	0.7922	0.6329	0.6457	0.4556	0.8008	0.6918	0.1301	0.0578

The best results are in bold.

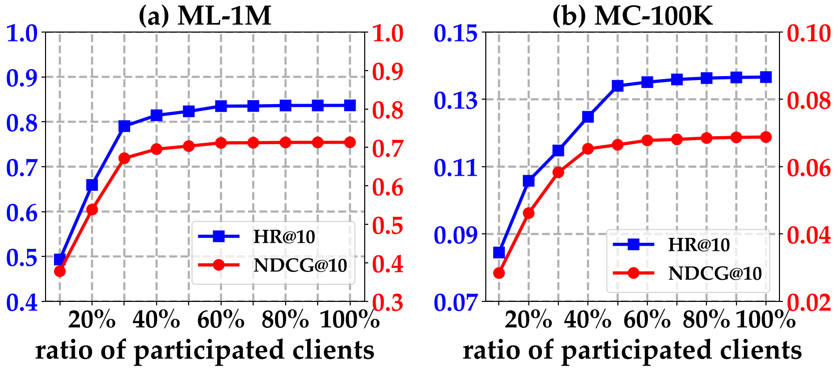
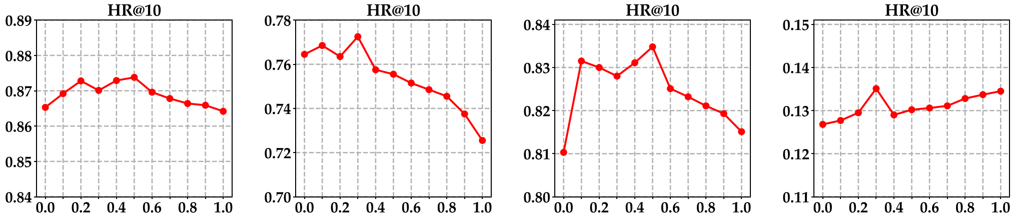


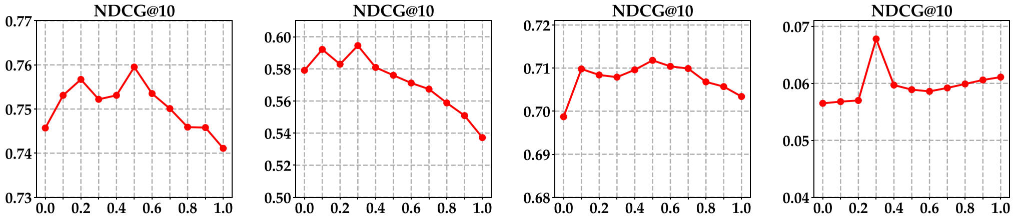
Fig. 8. Results of our FedCA with varying ratios of participated clients.



(a) ML-100K (b) Filmtrust (c) ML-1M (d) MC-100K

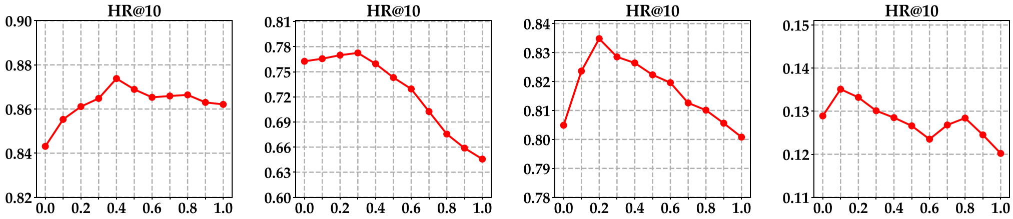
Fig. 9. Impact of α on HR@10 across four datasets.

and data complementarity throughout the overall optimization process. Figures 9 and 11, respectively, demonstrate the performance impact of α and β on HR@10 across four datasets. Figures 10 and 12, respectively, demonstrate the performance impact of α and β on NDCG@10 across four datasets. Overall, the variations of these hyperparameters have relatively minor effects on model performance. Thus, our method exhibits insensitivity to hyperparameters, although manual adjustment of these parameters is still necessary to achieve optimal performance. Through experimental analysis, it was found that the optimal prediction accuracy is achieved with $\alpha = 0.5$, $\beta = 0.4$ for ML-100K, $\alpha = 0.3$, $\beta = 0.3$ for Filmtrust, $\alpha = 0.5$, $\beta = 0.2$ for ML-1M, and $\alpha = 0.3$, $\beta = 0.1$ for MC-100K datasets. Furthermore, we observed that overall, the weight of α is larger than that of β. This indicates that throughout the aggregation process, greater emphasis should be placed on



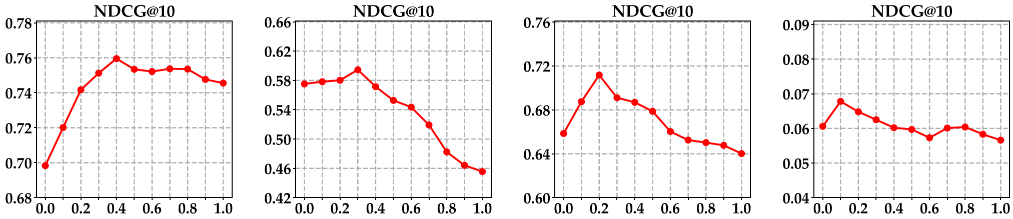
(a) ML-100K (b) Filmtrust (c) ML-1M (d) MC-100K

Fig. 10. Impact of α on NDCG@10 across four datasets.



(a) ML-100K (b) Filmtrust (c) ML-1M (d) MC-100K

Fig. 11. Impact of β on HR@10 across four datasets.



(a) ML-100K (b) Filmtrust (c) ML-1M (d) MC-100K

Fig. 12. Impact of β on NDCG@10 across four datasets.

model similarity to ensure that the parameter distribution of the clients to be aggregated remains consistent with their own parameters. Subsequently, it is essential to take into account the concept of complementary clients in order to enhance the generalization capability of the model on the test set.

Regarding the optimal parameter selection, we would like to provide the following additional details. (1) During the tuning of α and β, we identified some general guidelines that may help reduce the complexity. For instance, the similarity parameter α generally is larger than the complementarity parameter β, indicating that the model should prioritize similarity before considering complementarity. Hence, in most scenarios, $\alpha = 0.5$ and $\beta = 0.2$ can effectively balance similarity and complementarity. (2) Certainly, to achieve optimal results, partitioning a small portion of the dataset as a validation set for hyperparameter search around our recommended parameters is advisable. This approach is practical for large-scale requirements and does not incur significant additional costs before deployment. (3) The versatility of our model necessitates addressing the parameter tuning issue, which we have discussed as a potential limitation in the conclusion. Besides, developing an automatic parameter search technique to reduce the complexity presents a promising direction for future research.

7 Conclusion

This work first rethinks the fundamental differences between federated vision and FR tasks. Specifically, the federated vision community primarily utilizes structured parameters (e.g., CNNs) for federated optimization, whereas FR tasks mainly employ one-to-one item embedding tables for personalized recommendations. This key difference renders similarity-based aggregation borrowed from federated vision domain ineffective for aggregating embedding tables, leading to embedding skew issues. To address the above challenge, we introduce a composite aggregation mechanism tailored for FR tasks. Specifically, by combining model similarity and data complementarity within a unified optimization framework, our approach enhances the trained embeddings of items that a client has already interacted with and optimizes the non-trained embeddings of items the client has not interacted with. This enables effective prediction of future items. Besides, we explore the ineffectiveness of the proximal term on personalized preferences in FR tasks and propose an interpolation method to alleviate the spatial misalignment issue in FRs.

This research specifically proposes a promising composite aggregation framework for FR tasks. It is a model-agnostic, plug-and-play module that can be seamlessly integrated into mainstream FR models. However, we need to manually adjust the weight allocation for similarity and complementarity in this work. These limitations can be alleviated by using automated machine learning techniques to learn the weight allocation adaptively in future studies. Besides, exploring more suitable model similarity and data complementarity mechanisms for FR tasks is also a promising research direction.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. arXiv:1901.09888. Retrieved from https://arxiv.org/abs/1901.09888

[2] Shilong Bao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2022. Rethinking collaborative metric learning: Toward an efficient alternative without negative sampling. *IEEE Trans. Pattern Anal. Mach. Intell*. 45, 1 (2022), 1017–1035.

[3] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. 2010. Random projections for k-means clustering. In *NeurIPS*.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends in Mach. Learn.* 3, 1 (2011), 1–122.

[5] Qiqi Cai, Jian Cao, Guandong Xu, and Nengjun Zhu. 2024. Distributed recommendation systems: Survey and research directions. *ACM Trans. Inf. Syst*. 43, 1 (2024), 1–38.

[6] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. Secure federated matrix factorization. *IEEE Intell. Syst*. 36, 5 (2020), 11–20.

[7] Jundong Chen, Honglei Zhang, Haoxuan Li, Chunxu Zhang, Zhiwei Li, and Yidong Li. 2025. Beyond personalization: Federated recommendation with calibration via low-rank decomposition. arXiv:2506.09525. Retrieved from https://arxiv.org/abs/2506.09525

[8] Jundong Chen, Honglei Zhang, Chunxu Zhang, Fangyuan Luo, and Yidong Li. 2026. Breaking the aggregation bottleneck in federated recommendation: A personalized model merging approach. In *AAAI*.

[9] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *ICML*, 2089–2099.

[10] Steven Diamond and Stephen Boyd. 2016. CVXPY: A python-embedded modeling language for convex optimization. *J. Mach. Learn. Res* 17, 83 (2016), 1–5.

[11] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. 2021. Deep learning with label differential privacy. In *NeurIPS*, 27131–27145.

[12] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. In *NeurIPS*, 19586–19597.

[13] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

[14] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2013. A novel Bayesian similarity measure for recommender systems. In *IJCAI*, 2619–2625.

[15] Mingzhe Han, Dongsheng Li, Jiafeng Xia, Jiahao Liu, Hansu Gu, Peng Zhang, Ning Gu, and Tun Lu. 2025. FedCIA: Federated collaborative information aggregation for privacy-preserving recommendation. In *SIGIR*, 1687–1696.

[16] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. 2020. Lower bounds and optimal algorithms for personalized federated learning. In *NeurIPS*, 2304–2315.

[17] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. arXiv:1811.03604. Retrieved from https://arxiv.org/abs/1811.03604

[18] F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst*. 5, 4 (2015), 1–19.

[19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *The Web Conference*, 173–182.

[20] Xinrui He, Shuo Liu, Jacky Keung, and Jingrui He. 2024. Co-clustering for federated recommender system. In *The Web Conference*, 3821–3832.

[21] Mubashir Imran, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Alexander Zhou, and Kai Zheng. 2023. ReFRS: Resource-efficient federated recommender system for dynamic and diversified user preferences. *ACM Trans. Inf. Syst* 41, 3 (2023), 1–30.

[22] Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. 2019. Learning private neural language modeling with attentive aggregation. In *IJCNN*, 1–8.

[23] Jing Jiang, Chunxu Zhang, Honglei Zhang, Zhiwei Li, Yidong Li, and Bo Yang. 2025. A tutorial of personalized federated recommender systems: Recent advances and future directions. In *The Web Conference*, 21–24.

[24] Ellango Jothimurugesan, Kevin Hsieh, Jianyu Wang, Gauri Joshi, and Phillip B. Gibbons. 2023. Federated learning under distributed concept drift. In *IJCAI*, 5834–5853.

[25] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*, 197–206.

[26] Farwa K. Khan, Adrian Flanagan, Kuan Eeik Tan, Zareen Alamgir, and Muhammad Ammad-Ud-Din. 2021. A payload optimization method for federated recommender systems. In *RecSys*, 432–442.

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

[28] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Phys. Rev. E* 69, 6, (2004), 1–16.

[29] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *MLSys*, 429–450.

[30] Zhiwei Li, Guodong Long, Chunxu Zhang, Honglei Zhang, Jing Jiang, and Chengqi Zhang. 2024. Navigating the future of federated recommendation systems with foundation models. arXiv:2406.00004. Retrieved from https://arxiv.org/abs/2406.00004

[31] Zhiwei Li, Guodong Long, and Tianyi Zhou. 2024. Federated recommendation with additive personalization. In *ICLR*.

[32] Zhitao Li, Xueyang Wu, Weike Pan, Youlong Ding, Zeheng Wu, Shengqi Tan, Qian Xu, Qiang Yang, and Zhong Ming. 2024. FedCORE: Federated learning for cross-organization recommendation ecosystem. *IEEE Trans. Knowl. Data Eng* 36, 8 (2024), 3817–3831.

[33] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. 2020. Fedrec: Federated recommendation with explicit feedback. *IEEE Intell. Syst* 36, 5 (2020), 21–30.

[34] Bingyan Liu, Yao Guo, and Xiangqun Chen. 2021. PFA: Privacy-preserving federated adaptation for effective model personalization. In *The Web Conference*, 923–934.

[35] Sichun Luo, Yuanzhang Xiao, and Linqi Song. 2022. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *CIKM*, 4289–4293.

[36] Sichun Luo, Yuanzhang Xiao, Xinyi Zhang, Yang Liu, Wenbo Ding, and Linqi Song. 2024. Perfedrec++: Enhancing personalized federated recommendation with self-supervised pre-training. *ACM Trans. Intell. Syst. Technol* 15, 5 (2024), 1–24.

[37] Xingyuan Mao, Yuwen Liu, Lianyong Qi, Li Duan, Xiaolong Xu, Xuyun Zhang, Wanchun Dou, Amin Beheshti, and Xiaokang Zhou. 2024. Cluster-driven personalized federated recommendation with interest-aware graph convolution network for multimedia. In *ACM MM*, 5614–5622.

[38] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTAT*, 1273–1282.

[39] Andriy Mnih and Russ R. Salakhutdinov. 2007. Probabilistic matrix factorization. In *NeurIPS*.

[40] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. Fedfast: Going beyond average for faster training of federated recommender systems. In *SIGKDD*, 1234–1242.

[41] Ngoc-Hieu Nguyen, Tuan-Anh Nguyen, Tuan Nguyen, Vu Tien Hoang, Dung D. Le, and Kok-Seng Wong. 2024. Towards efficient communication and secure federated recommendation system via low-rank training. In *The Web Conference*, 3940–3951.

[42] Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. 2023. A content-driven micro-video recommendation dataset at scale. arXiv:2309.15379. Retrieved from https://arxiv.org/abs/2309.15379

[43] Lin Ning, Karan Singhal, Ellie X. Zhou, and Sushant Prakash. 2021. Learning federated representations and recommendations with limited negatives. arXiv:2108.07931. Retrieved from https://arxiv.org/abs/2108.07931

[44] Vasileios Perifanis and Pavlos S. Efraimidis. 2022. Federated neural collaborative filtering. *Knowl.-Based Syst*. 242 (2022), 1–16.

[45] Gilbert Strang. 2012. *Linear Algebra and Its Applications*. Brooks/Cole.

[46] Jiajie Su, Chaochao Chen, Yihao Wang, Weiming Liu, Yuyuan Li, Tao Wang, Zhigang Li, Xiaolin Zheng, and Jianwei Yin. 2025. DuAda: Adaptive targeted model poisoning attack framework via dummy user simulation on federated recommendation. *ACM Trans. Inf. Syst*. 43, 6 (2025), 1–37.

[47] Hao Wang, Zhichao Chen, Honglei Zhang, Zhengnan Li, Licheng Pan, Haoxuan Li, and Mingming Gong. 2025. Debiased recommendation via wasserstein causal balancing. *ACM Trans. Inf. Syst*. 43, 6 (2025), 1–24.

[48] Shanfeng Wang, Yuxi Zhou, Xiaolong Fan, Jianzhao Li, Zexuan Lei, and Maoguo Gong. 2025. Personalized federated contrastive learning for recommendation. *IEEE Trans. Comput. Soc. Syst*. 12, 5 (2025), 2986–2998.

[49] Jinze Wu, Qi Liu, Zhenya Huang, Yuting Ning, Hao Wang, Enhong Chen, Jinfeng Yi, and Bowen Zhou. 2021. Hierarchical personalized federated learning for user modeling. In *The Web Conference*, 957–968.

[50] Enyue Yang, Weike Pan, Qiang Yang, and Zhong Ming. 2024. Discrete federated multi-behavior recommendation for privacy-preserving heterogeneous one-class collaborative filtering. *ACM Trans. Inf. Syst*. 42, 5 (2024), 1–50.

[51] Yun Yang, Yuanyuan Hu, Xingyi Zhang, and Song Wang. 2021. Two-stage selective ensemble of CNN via deep tree training for medical image classification. *IEEE Trans. Cybern*. 52, 9 (2021), 9194–9207.

[52] Rui Ye, Zhenyang Ni, Fangzhao Wu, Siheng Chen, and Yanfeng Wang. 2023. Personalized federated learning with inferred collaboration graphs. In *ICML*, 39801–39817.

[53] Wei Yuan, Liang Qu, Lizhen Cui, Yongxin Tong, Xiaofang Zhou, and Hongzhi Yin. 2024. HeteFedRec: Federated recommender systems with model heterogeneity. In *ICDE*.

[54] Wei Yuan, Chaoqun Yang, Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2024. FELLAS: Enhancing federated sequential recommendation with llm as external services. *ACM Trans. Inf. Syst*. 43, 6 (2024), 1–25.

[55] Chunxu Zhang, Guodong Long, Zijian Zhang, Zhiwei Li, Honglei Zhang, Qiang Yang, and Bo Yang. 2025. Personalized recommendation models in federated settings: A survey. *IEEE Trans. Knowl. Data Eng* 37, 11 (2025), 6562–6581.

[56] Chunxu Zhang, Guodong Long, Tianyi Zhou, Peng Yan, Zijian Zhang, Chengqi Zhang, and Bo Yang. 2024. Dual personalization on federated recommendation. In *IJCAI*.

[57] Honglei Zhang, Zhiwei Li, Haoxuan Li, Xin Zhou, Jie Zhang, and Yidong Li. 2026. Transfr: Transferable federated recommendation with adapter tuning on pre-trained language models. In *AAAI*.

[58] Honglei Zhang, Fangyuan Luo, Jun Wu, Xiangnan He, and Yidong Li. 2023. LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization. *ACM Trans. Inf. Syst* 41, 4 (2023), 1–28.

[59] Honglei Zhang, Shuyi Wang, Haoxuan Li, Chunyuan Zheng, Xu Chen, Li Liu, Shanshan Luo, and Peng Wu. 2024. Uncovering the propensity identification problem in debiased recommendations. In *ICDE*, 653–666.

[60] Honglei Zhang, Xin Zhou, Zhiqi Shen, and Yidong Li. 2025. PrivFR: Privacy-enhanced federated recommendation with shared hash embedding. *IEEE Trans. Neural Networks Learn. Syst*. 36, 1 (2025), 32–46.

[61] Zhongjian Zhang, Mengmei Zhang, Xiao Wang, Lingjuan Lyu, Bo Yan, Junping Du, and Chuan Shi. 2025. Rethinking byzantine robustness in federated recommendation from sparse aggregation perspective. In *AAAI*.

[62] Pengyang Zhou, Chaochao Chen, Weiming Liu, Wenkai Shen, Xinting Liao, Huarong Deng, Zhihui Fu, Jun Wang, Wu Wen, and Xiaolin Zheng. 2025. Joint item embedding dual-view exploration and adaptive local-global fusion for federated recommendation. In *SIGIR*, 424–434.

Received 7 May 2025; revised 10 October 2025; accepted 28 November 2025