# Federated Context-Aware Personalized Recommendation

**Zhihao Wang**[1*], **Xiaoying Liao**[23*], **Wenke Huang**[1*], **Bingqian Liu**[1], **Tian Chen**[1], **Jian Wang**[1†], **Bing Li**[1†]
[1]School of Computer Science, Wuhan University
[2]Changsha Bus Group
[3]Central South University of Forestry and Technology
{zhihao_wang, jianwang, bingli}@whu.edu.cn

## Abstract

Federated recommender system is emerging as a new paradigm for providing personalized services while preserving the privacy of user data. Most existing personalized federated recommender systems predict the user's next item by discretely training user and item embeddings. However, this training approach often overlooks the user's behavioral patterns, suffers from low interpretability, and requires a substantial amount of data and highly meticulous fine-tuning to achieve stable and accurate embeddings. To address these limitations, we propose **Fed**erated **C**ontext-**A**ware Personalized **R**ecommendation (*FedCAR*), a novel framework that effectively leverages users' recent interactions as behavioral context to guide prediction. Instead of static user embeddings, *FedCAR* dynamically constructs context representations by aggregating and weighting recently interacted item embeddings. Additionally, we incorporate a contrastive learning strategy that enables the model to capture shared behavioral structures across clients while maintaining personalized preferences, thereby enhancing both generalization and robustness in heterogeneous settings. Experiments on 5 benchmark datasets show that *FedCAR* outperforms state-of-the-art methods and provides interpretable recommendations by explicitly modeling context dependencies.

## 1 Introduction

Personalized recommendation systems are essential for delivering relevant content across online platforms. Traditional approaches (Kang and McAuley 2018; Li et al. 2023; Liu et al. 2024; Zhang et al. 2024c, 2025c) typically rely on centralized architectures where user data is collected and processed on a central server. While this enables effective modeling, it also raises serious privacy concerns and conflicts with regulations such as the General Data Protection Regulation (GDPR) (Voigt and Bussche 2017). Federated Recommendation (FR) (Lin et al. 2021; Liang, Pan, and Ming 2021; Zhang et al. 2023, 2024b; Wang et al. 2025a) has emerged as a promising solution, enabling decentralized model training while keeping user data local and thereby mitigating the privacy risks inherent in centralized methods.

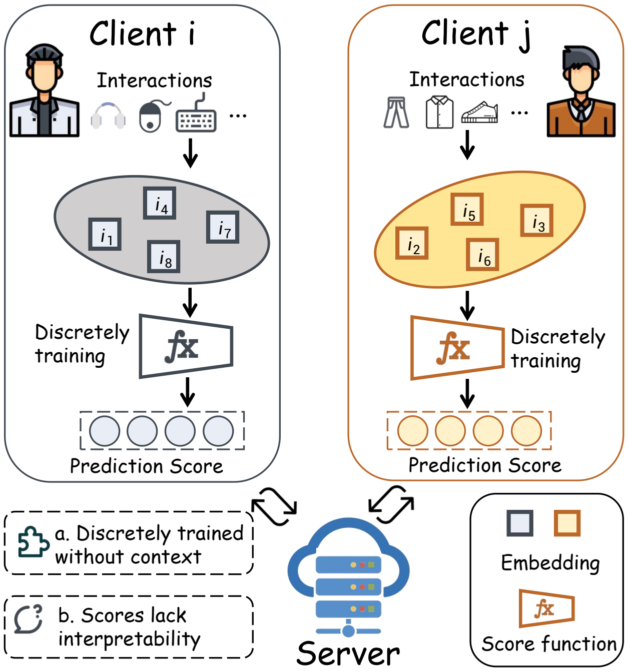


Figure 1: Illustration of personalized federated Recommendation. Local clients sample items from interaction history and make predictions, while the server aggregates parameters. This paradigm brings up two questions: (a) Clients discretely train on isolated samples without considering context. (b) The prediction scores lack interpretability and make it hard to understand user preferences.

While federated recommendation protects user data, achieving effective personalization in this setting remains challenging. As shown in Fig. 1, most existing methods rely on user-specific embeddings trained locally or through periodic aggregation. However, this paradigm often leads to suboptimal performance and provides limited insight into user behavior. In particular, these models struggle to adapt to dynamic user interests and typically lack interpretability, making it difficult to explain why certain items are recommended. Addressing these issues requires moving beyond static identity-based modeling toward more flexible and ex-

---

[*] Equal contribution.
[†] Corresponding author

plainable approaches.

Prior work has explored matrix factorization (Chai et al. 2021; Perifanis and Efraimidis 2022), graph aggregation(Wu et al. 2022; Zhang et al. 2024b), and reconstruction (Singhal et al. 2021) to enhance federated recommendation. Although these methods bring improvements in personalization, they generally continue to treat users as fixed entities and overlook behavioral dynamics. Furthermore, many approaches struggle to generalize across heterogeneous clients, especially when interaction histories are short.

To tackle these challenges, we propose Federated Context-Aware Personalized Recommendation (*FedCAR*), a novel federated framework that moves beyond the conventional paradigm of discretely training user (item) embeddings. Instead, it captures user preferences from recent behavioral context to enhance both personalization and interpretability. The framework consists of two key components. First, a local context encoder models recent user interaction sequences on-device, allowing the model to adapt to short-term behavior patterns while maintaining data locality. Instead of depending on persistent user identifiers, this encoder dynamically captures contextual preferences, enabling real-time personalization. Additionally, the attention-based context encoder provides interpretability by identifying which interactions most influence each prediction. Second, we introduce a contrastive learning objective that distinguishes the user's context-aware preferences from those of other clients. This helps the model focus on uniquely relevant patterns rather than simply relying on global trends. By jointly leveraging dynamic behavioral context and contrastive personalization, *FedCAR* provides a robust and interpretable solution for personalized federated recommendation.

Our contributions are summarized as follows:

- We propose *FedCAR*, a federated recommendation framework that captures recent user behavior through a local context encoder, providing a lightweight and privacy-preserving alternative to traditional approaches based on discrete personalization.
- We propose a contrastive learning strategy tailored to federated settings, which enhances model adaptability by distinguishing personalized behavior patterns across clients.
- We conduct extensive experiments on 5 public datasets, and the results show the superiority of our model over state-of-the-art methods.

# 2 Related Work

## 2.1 Federated Learning

Federated learning (FL) has emerged as a crucial machine learning paradigm aimed at enabling collaborative model training while preserving user data privacy, especially in scenarios marked by significant data heterogeneity (Konečný et al. 2016; Kairouz et al. 2019; Li et al. 2020a; Long et al. 2020; Huang, Ye, and Du 2022; Huang et al. 2023a,b, 2024; Wang et al. 2025b; Chen et al. 2025; Ma et al. 2025). The foundational work FedAvg (McMahan et al. 2017) proposed aggregating locally trained model parameters to construct a shared global model, laying the groundwork for many subsequent developments. Nevertheless, FedAvg struggles under non-i.i.d. data distributions—a common issue in real-world applications, which has inspired a range of improvements (Luo et al. 2021; Li et al. 2022; Ma et al. 2022; Dai et al. 2023a; Tan et al. 2023; Hong et al. 2023; Hu et al. 2024). To mitigate the effects of heterogeneity, many methods have introduced regularization mechanisms. SCAFFOLD (Karimireddy et al. 2020) uses control variates to correct local model updates, effectively reducing client drift and improving convergence speed. Similarly, FedProx (Li et al. 2020b) incorporates a proximal term to allow stable aggregation, especially when local updates are partial or imbalanced due to system-level constraints. FedStar (Tan et al. 2023), on the other hand, targets graph-based data and facilitates the transfer of shared latent structures across clients. Alternative strategies focus on controlling the divergence of client models. pFedME (T. Dinh, Tran, and Nguyen 2020) and FedDyn (Acar et al. 2021) both introduce optimization dynamics that encourage alignment of local models with a central objective, even under heterogeneous conditions. Meanwhile, some methods prioritize alignment in the representation space. Approaches like MOON (Li, He, and Song 2021), FedUFO (Zhang et al. 2021), FedProto (Tan et al. 2022), FedProc (Mu et al. 2023), and FedNH (Dai et al. 2023b) emphasize consistency between local and global embeddings to promote feature-level agreement and enhance model robustness. FedConcat (Diao, Li, and He 2024) cluster clients based on label distributions and encourage intra-group collaboration by aggregating local models accordingly. While prior FL works primarily focus on global model alignment or embedding consistency under heterogeneity, we mainly explore behavior-aware personalization.

## 2.2 Federated Recommendation

With the growing demand for data privacy and the increasing prevalence of decentralized data scenarios, federated recommendation has emerged as a viable solution for building personalized services without exposing raw user data (Sun et al. 2024; Zhang et al. 2024a; Wang et al. 2025a; Zhang et al. 2025a,b; Jiang et al. 2025). Early work, such as FCF (Ammad-ud-din et al. 2019), extended collaborative filtering to the federated setting by adopting FedAvg to aggregate local updates. Building on this foundation, FedRec (Lin et al. 2021) introduced strategies like user-level averaging and hybrid filling to mask sensitive rating information, while FedRec++ (Liang, Pan, and Ming 2021) further enhanced robustness through the involvement of dedicated denoising clients. FedMF (Chai et al. 2021) incorporated homomorphic encryption into matrix factorization, offering secure parameter updates, and P-NSMF (Hu et al. 2022) proposed group-concealing techniques to enable non-sampling factorization under secure aggregation. FedNCF (Perifanis and Efraimidis 2022) adapted neural collaborative filtering to federated environments, allowing deep embedding of users and items. More recent approaches, including FedPerGNN (Wu et al. 2022), PerFedRec (Luo, Xiao, and Song 2022), and P-GCN (Hu et al. 2023), enrich local representation learning while preserving user privacy by encoding neighborhood or item relations in a protected manner. Additionally, methods such as P2FCDR (Chen et al. 2023), FPPDM (Liu et al. 2023), and F2PGNN (Agrawal et al. 2024) enhance personalization through transformation or alignment techniques while embedding privacy-preserving mechanisms like local differential privacy. PFedRec (Zhang et al. 2023) proposes a communication-efficient framework that transmits only the item encoder during training, keeping user-specific modeling local to further reduce exposure risks. GPFedRec (Zhang et al. 2024b) further utilizes the user graph to guide the training. These advancements collectively advance the capabilities of federated recommendation systems in both accuracy and privacy-aware personalization. Different from prior methods that rely on static embeddings, our method captures dynamic user intent and enhances personalization by distinguishing individual behaviors.

# 3 Method

In this section, we define the research problem and introduce our proposed Federated Context-Aware Personalized Recommendation (*FedCAR*) in detail.

## 3.1 Problem Formulation

**Federated Learning.** According to the general setting (Huang, Ye, and Du 2022; Ma et al. 2022), the purpose of federated learning is to learn a global model with parameter $\theta$ based on private data $D_K$ of $K$ clients (indexed by $k$). The training process aims to minimize the accumulated loss of all $K$ clients,

$$\min_{\theta} \sum_{k=1}^{K} \alpha_k \mathcal{L}_k(\theta), \tag{1}$$

where $L_k(\theta)$ denotes the loss on the $k$-th client with data $D_k$, and all clients share the global parameter $\theta$. The $\alpha_k$ means the weight of loss in client $k$. After the local training epochs of clients $k$, it uploads back to the server for parameter aggregation in communication epoch $e$,

$$\theta_k^e \leftarrow \theta_k^e - \eta \nabla \sum_{i \in \mathcal{B}_k} \mathcal{L}(\theta_k^e, \xi_i), \quad \theta^{e+1} = \sum_{k} \alpha_k \theta_k^e. \tag{2}$$

The $\mathcal{B}$ means the mini-batch sampled from private data $D_k$, $\xi$ represents the query instance. The $\eta$ means the local learning rate. For example, FedAvg (McMahan et al. 2017) assigns the weight $\alpha_k$ according to the scale of clients data: $\alpha_k = D_k / \sum_{j=1}^{K} D_j$.

**Personalized Federated Learning.** From another perspective, personalized federated learning (PFL) focuses on tailoring models to individual clients while still leveraging shared knowledge across the federation. Unlike standard FL, where all clients optimize a shared global model $\theta$, PFL simultaneously learns a set of client-specific models $\{\theta_k\}_{k=1}^K$ together with the global representation. The typical optimization objective is formulated as:

$$\min_{\theta, \{\theta_k\}_{k=1}^K} \sum_{k=1}^{N} \alpha_k \mathcal{L}_k(\theta, \theta_k), \tag{3}$$

where $\theta_k$ denotes the personalized parameter for the $k$-th client, and $\theta$ represents the shared global knowledge. The loss function $\mathcal{L}_k(\theta, \theta_k)$ captures both the global and local perspectives in the training process.

**Personalized Federated Recommendation.** In the personalized federated recommendation (PFR) setting, each client $k$ corresponds to a unique user and holds private interaction data $D_k = \{(i, r)\}$, where $i$ is an interacted item and $r$ denotes the corresponding feedback, such as an explicit rating or implicit signal. The goal of PFR is to learn a recommendation model that captures user-specific preferences while benefiting from knowledge shared across clients.

Formally, the objective is to jointly optimize a global model parameter $\theta$ and client-specific personalization parameters $\{\theta_k\}_{k=1}^K$, with the following loss:

$$\min_{\theta, \{\theta_k\}_{k=1}^K} \sum_{k=1}^{K} \alpha_k \mathcal{L}_k(\theta, \theta_k; D_k), \tag{4}$$

where $\mathcal{L}_k(\theta, \theta_k; D_k)$ denotes the local loss for client $k$, incorporating both shared and personalized components. Specifically, the model predicts scores $\hat{r}_{k,i}$ for each item $i$ in $D_k$, and the local objective is computed as:

$$\mathcal{L}_k(\theta, \theta_k; D_k) = \frac{1}{|D_k|} \sum_{(i,r) \in D_k} \ell\left(f(k, i; \theta, \theta_k), r\right), \tag{5}$$

where $f(k, i; \theta, \theta_k)$ denotes the predicted score generated using both the global parameter $\theta$ and the personalized adaptation $\theta_k$, and $\ell(\cdot)$ is a suitable loss function such as mean squared error or cross-entropy.

## 3.2 Algorithm Design

We treat each user as a client under FL setting. Inspired by PFedRec(Zhang et al. 2023), we maintain the item embedding layer in clients and ignore the explicit user embedding layer in previous work like FedMF (Chai et al. 2021) and FedNCF (Perifanis and Efraimidis 2022). On the one hand, explicit user embedding may cause the privacy problem. On the other hand, the client score function can play the role of user preference and make predictions. The classic framework can be simplified as follows.

$$\mathbf{e}_{ki} = \mathbf{e}_k(i), \tag{6}$$

$$\hat{r}_{ki} = S_k(\mathbf{e}_{ki}), \tag{7}$$

where $\mathbf{e}_k$ denote the item embedding layer of client $k$, $\mathbf{e}_{ki}$ is the corresponding embedding of item $i$, $S_k$ is the score function of client $k$. However, such a training approach remains discrete, lacks interpretability, and is sensitive to user data.

**Context-aware Local Modeling.** In order to make more reasonable predictions, we additionally introduce a context encoder to comprehensively consider user behavior patterns. Naturally, we take the user's most recent $L$ interacted items as the user's context,

$$context_k = \{i_{current-L}, i_{current-L+1}, \ldots, i_{current-1}\}. \tag{8}$$

We first obtain the item embeddings of the contextual items via the client-side embedding layer $\mathbf{e}_k$:

$$\mathbf{E}_k = [\mathbf{e}_k(i)]_{i \in context_k} \in \mathbb{R}^{L \times d}, \tag{9}$$

where $d$ is the embedding dimension.

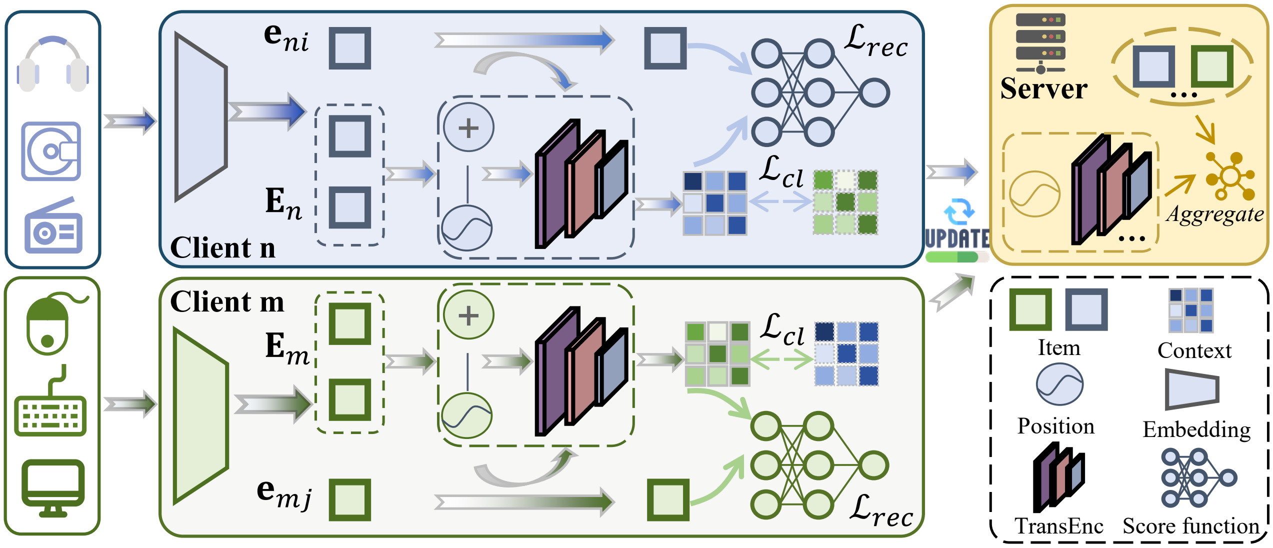


Figure 2: Architecture illustration of Federated Context-Aware Personalized Recommendation (*FedCAR*). To move beyond treating each candidate item in isolation, we introduce a context-aware modeling module. It fuses user context and position embedding through a TransEnc layer, then performs attention over the candidate item to generate a context representation. Additionally, to enhance personalization of the local context encoder, we employ a personalized contrastive learning strategy. Specifically, we use the same items and context, but apply the context encoder from the previous communication round to generate a positive sample, while a noise-injected context encoder from another random client provides the negative sample for contrastive learning. Please refer to Sec. 3.2. Best viewed in color. Zoom in for details.

To model temporal dependencies within the user's interaction sequence, we incorporate a learnable positional encoding matrix $\mathbf{P} \in \mathbb{R}^{L\times d}$ into the sequence embeddings:

$$\mathbf{Z}_k = \mathbf{E}_k + \mathbf{P}. \quad (10)$$

The enriched sequence $\mathbf{Z}_k$ is then passed through a single-layer Transformer encoder TransEnc (Vaswani et al. 2017) to extract the user's short-term preference representation. Specifically, we treat the current target item embedding as the query, and the transformed context as keys and values:

$$\mathbf{H}_k = \mathrm{TransEnc}(\mathbf{Z}_k) \in \mathbb{R}^{L\times d}, \quad (11)$$

$$\mathbf{q}_k = \mathbf{e}_k(i_{current}), \quad (12)$$

$$\mathbf{c}_k = \mathrm{Attn}(\mathbf{q}_k, \mathbf{H}_k, \mathbf{H}_k) \in \mathbb{R}^{d}, \quad (13)$$

where $\mathrm{Attn}(\cdot)$ denotes the scaled dot-product attention mechanism. Note that TransEnc is a shared and trainable Transformer encoder layer, and is updated through federated aggregation during training. It enables general modeling of behavioral patterns while preserving user-level data locality. Finally, the context representation is combined with the target item embedding for prediction:

$$\hat{r}_{ki} = S_k(\mathbf{c}_k, \mathbf{q}_k), \quad (14)$$

where $S_k$ is the client-specific score function that computes the interaction score based on both context and item representations. Given the prediction $\hat{r}_{ki}$, we can calculate the loss for recommendation:

$$\mathcal{L}_{rec_k} = -\sum_{i\in D_k} \log(\hat{r}_{ki}) - \sum_{i'\in D_k^-} \log(1-\hat{r}_{ki'}). \quad (15)$$

In this formulation:

- $D_k$ represents the set of positive samples, containing items that client $k$ has historically interacted with.
- $D_k^-$ is the set of negative samples used for training. Constructing this set efficiently is critical. We first define the pool of all unobserved items for client $k$ as $I_k^- = \mathcal{I} \setminus I_k$, where $\mathcal{I}$ is the universal set of all items and $I_k$ is the set of items in $D_k$. Then, to form $D_k^-$, we perform uniform random sampling from the candidate pool $I_k^-$.

This approach enhances the expressiveness and interpretability of the model while mitigating privacy concerns arising from explicit user embeddings.

**Personalized Contrastive Adaptation.** Through parameter aggregation, the context encoder can learn common user behavior patterns. In addition, we implement personalized adaptation to capture user-specific preferences. Concretely, we conduct contrastive learning based on the user's contextual representation to encourage local distinguishability.

For a given client $k$, we extract the current context embedding $\mathbf{c}_k^{\mathrm{curr}}$ as described above. Then, we define the **positive sample** as the context embedding generated by the same client's model from the previous communication round:

$$\mathbf{c}_k^{\mathrm{pos}} = C_k^{(t-1)}(\mathbf{E}_k^{t-1}), \quad (16)$$

where $C_k^{(t-1)}$ denotes client $k$'s context encoder (TransEnc, Attn() and $\mathbf{P}$) from the previous round $t-1$, and $\mathbf{E}_k^{t-1}$ is the context embedding in Eq. (9).

To construct the **negative sample**, we randomly select another client $k' \neq k$ and use its context encoder from the previous round, to which we apply Laplace noise to enhance

**Algorithm 1:** Model training in *FedCAR*

```
Input: Communication epochs $E$, local rounds $T$, number
       of participants $K$, the $k^{th}$ participant private data
       $D_k = \{(i, r)\}$, private model $\theta_k$ (contains score
       function $S_k$, item embedding $\mathbf{e}_k$ and context
       encoder $C_k$)
Output: The final client model set $\{\theta_k\}_{k=1}^{K}$

for $e = 1, 2, ..., E$ do
    $SC^e \leftarrow$ Sample clients of size $n$ randomly from $K$
    Participant Side;
    for client $k \in SC^e$ in parallel do
        $\mathbf{e}_k, C_k \leftarrow$ LocalUpdating($\mathbf{e}, f$)
    Server Side;
    $\mathbf{e} \leftarrow \frac{1}{n} \sum_{k=1}^{n} \mathbf{e}_k$, Update global item embedding
    $C \leftarrow \frac{1}{n} \sum_{k=1}^{n} C_k$, Update global context encoder

LocalUpdating($\mathbf{e}, C$):
$\mathbf{e}_k, C_k \leftarrow \theta$, Distribute global parameter
$S_k \leftarrow$ Initialize score function with last update
Count all uninteracted items set $\mathcal{I}_k^-$
Sample negative samples set $\mathcal{D}_k^-$ from $\mathcal{I}_k^-$
$\mathcal{B} \leftarrow$ Load data batch from $\mathcal{D}_k \cup \mathcal{D}_k^-$
for $t = 1, 2, ..., T$ do
    for $b \in \mathcal{B}$ do
        $\mathbf{e}_{ki}, \mathbf{E}_k \leftarrow$ get embedding in Eq. (6),Eq. (9)
        $\mathbf{c}_k \leftarrow$ Get context in Eq. (13)
        $\hat{r}_{ki} \leftarrow$ Make prediction in Eq. (14)
        $\mathcal{L}_{rec_k} \leftarrow$ Compute prediction loss in Eq. (15)
        $\theta_k \leftarrow \theta_k - \eta \nabla_{\theta_k} \mathcal{L}_{rec_k}$, Update client parameters
        $\mathbf{c}_k^{\text{pos}} \leftarrow$ Get positive sample in Eq. (16)
        $\mathbf{c}_k^{\text{neg}} \leftarrow$ Get negative sample in Eq. (19)
        $\mathcal{L}_{cl_k} \leftarrow$ Compute contrastive loss in Eq. (20)
        $C_k \leftarrow C_k - \eta' \nabla_{C_k} \mathcal{L}_{cl_k}$ Update context encoder
Return $\mathbf{e}_k, C_k$ to server
```

| **Dataset** | #Users | #Items | #Interactions |
|---|---|---|---|
| **MovieLens-100K** | 943 | 1,682 | 100,000 |
| **MovieLens-1M** | 6,040 | 3,952 | 100,000,209 |
| **Amazon-Vedio** | 11,830 | 8,072 | 63,836 |
| **Lastfm-2K** | 1,600 | 12,454 | 185,650 |
| **HetRec2011** | 2,113 | 10,109 | 855,598 |

Table 1: Statistics of datasets

privacy:

$$\mathbf{c}_{k'}^{\text{raw}} = C_{k'}^{(t-1)}(\mathbf{E}_{k'}^{t-1}), \tag{17}$$

$$\epsilon \sim \text{Laplace}(0, \lambda^{neg}), \tag{18}$$

$$\mathbf{c}_k^{\text{neg}} = \mathbf{c}_{k'}^{\text{raw}} + \epsilon, \tag{19}$$

where Laplace$(0, \lambda^{neg})$ denotes element-wise Laplace noise with scale $\lambda^{neg}$, and $\lambda^{neg}$ is the privacy parameter controlling the noise level.

We then compute a margin-based contrastive loss:

$$\mathcal{L}_{cl_k} = \max\left(0, \left\|\mathbf{c}_k^{\text{curr}} - \mathbf{c}_k^{\text{pos}}\right\|^2 - \left\|\mathbf{c}_k^{\text{curr}} - \mathbf{c}_k^{\text{neg}}\right\|^2 + \delta\right), \tag{20}$$

where $\delta$ is a fixed margin hyperparameter.

*Importantly*, the contrastive loss is optimized in a separate training phase, where only the context encoder components (i.e., the Transformer encoder, attention layer, and positional encoding) are updated. Inspired by PFedRec (Zhang et al. 2023), this modular optimization strategy avoids interference with the main recommendation objective and allows for better control of local personalization without the need for a weighting factor. The workflow of *FedCAR* is illustrated in Algorithm 1.

# 4 Experiment

## 4.1 Experimental Setup

**Dataset.** We conduct experiments on five widely used public datasets: MovieLens-100K, MovieLens-1M (Harper and Konstan 2015), Amazon-Video (Ni, Li, and McAuley 2019), Lastfm-2K and HetRec-2K (Cantador, Brusilovsky, and Kuflik 2011). The MovieLens-100K and MovieLens-1M datasets are widely used benchmarks for recommender systems, containing 100K and 1M user-movie ratings respectively. For the Amazon dataset, we select the Video category. Lastfm-2K is a music recommendation dataset that records each user's listening history. HetRec2011 is an extended version of the MovieLens-10M dataset. Detailed statistics of all datasets are summarized in Tab. 1.

**Comparison Methods.** We compare *FedCAR* with both centralized methods and personalized federated methods. For centralized methods, we compare with MF (Koren, Bell, and Volinsky 2009) and NCF (He et al. 2017). For personalized federated methods, we evaluate FedMF (Chai et al. 2021), FedNCF (Perifanis and Efraimidis 2022), FedRecon (Singhal et al. 2021), PFedRec (Zhang et al. 2023) and GPFedRec (Zhang et al. 2024b).

**Evaluation Metrics.** We adopt two widely used metrics: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) (He et al. 2015) to evaluate model performance. Both of them measure the model's performance on the next-item prediction task.

**Implementation Details.** All methods operate under the most fundamental setting, where recommendations are made purely based on user-item interactions without incorporating any auxiliary information. We filter the users who interact with fewer than 6 items in every dataset, and the length of context is 5. The $\lambda^{neg}$ in Eq. (18) is set as 0.1 and $\delta$ in Eq. (20) is set as 1. As for the data split, we follow the classic leave-one-out evaluation (He et al. 2017; Kang and McAuley 2018). For each positive item, we sample 4 negative items following (He et al. 2017). For a fair comparison, we conduct the communication epochs $E = 100$, and perform 1 local rounds ($T = 1$) for the federated setting. We adopt the one-layer MLP as the score function. Besides, the initial embedding size is fixed at 32 and the batch size is

| Methods | | MovieLens-100K | | MovieLens-1M | | Amazon-Video | | Lastfm-2K | | Hetrec2011 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HR@10↑ | NDCG@10↑ | HR@10↑ | NDCG@10↑ | HR@10↑ | NDCG@10↑ | HR@10↑ | NDCG@10↑ | HR@10↑ | NDCG@10↑ |
| Central | MF | 0.6617 | 0.3884 | 0.4879 | 0.2694 | 0.3017 | 0.1742 | 0.8117 | 0.6480 | 0.6432 | 0.3916 |
| | NCF | 0.6416 | 0.3682 | 0.6613 | 0.3870 | 0.5520 | 0.3496 | 0.8288 | 0.6528 | 0.6588 | 0.4040 |
| Federated | FedMF | 0.1782 | 0.0900 | 0.2692 | 0.1294 | 0.1136 | 0.0522 | 0.5866 | 0.5540 | 0.1107 | 0.0505 |
| | FedNCF | 0.1208 | 0.0531 | 0.2947 | 0.1659 | 0.3543 | 0.3314 | 0.4898 | 0.4528 | 0.1283 | 0.630 |
| | FedRecon | 0.3584 | 0.1948 | 0.3831 | 0.2118 | 0.5181 | 0.3723 | 0.5181 | 0.3723 | 0.5182 | 0.3054 |
| | PFedRec | 0.5101 | 0.2754 | 0.5075 | 0.2811 | 0.5487 | **0.3534** | 0.8168 | 0.7003 | 0.5953 | 0.3525 |
| | GPFedRec | 0.5027 | 0.2754 | 0.5348 | 0.3202 | 0.4759 | 0.3137 | 0.7834 | 0.6908 | 0.5660 | 0.3494 |
| | ***FedCAR*** | **0.5270** | **0.2837** | **0.5555** | **0.3068** | **0.5515** | 0.3529 | **0.8236** | **0.7222** | **0.5963** | **0.3571** |

Table 2: Comparison with the state-of-the-art methods on MovieLens-100K, MovieLens-1M, Amazon-Video, Lastfm-2K, and Hetrec2011. The best results in FL setting are bolded and the suboptimal results are underlined. Please refer to Sec. 4.2.

set as 256 for all methods. We use the SGD (Robbins and Monro 1951) optimizer and set the same learning rate for methods depending on the datasets. We fix the seed to ensure reproduction and conduct experiments on the NVIDIA 3090 with the PyTorch framework.

## 4.2 Results

**Performance Comparison.** We conduct a comprehensive comparison between our proposed method *FedCAR* and several representative baselines. Experimental results across five public datasets are reported in Tab. 2. *FedCAR* consistently achieves top performance across most federated benchmarks. On Lastfm-2K, it yields an HR@10 of 0.8236 and NDCG@10 of 0.7222, outperforming the previous best (PFedRec). Similar trends are observed on MovieLens, Amazon-Video, and Hetrec2011, where *FedCAR* demonstrates both high accuracy and robustness. The superior performance of our method can be attributed to two key design components: (a) Context-aware Local Modeling: Unlike prior works that rely solely on static user-item embeddings or item ID sequences, *FedCAR* incorporates a local context encoder that captures the fine-grained, recent interaction patterns of each user. This allows the model to better understand what aspects of an item are relevant in the current recommendation situation, rather than treating preferences as fixed. (b) Personalized Contrastive Adaptation: We further enhance local personalization by introducing a contrastive learning objective tailored to federated settings, which enables each client to differentiate its own interaction behaviors from those of others. Some baselines perform poorly due to limited robustness under unified hyperparameter settings like a fixed learning rate.

**Explainable Recommendation.** Our method enhances explainability by leveraging attention between the target item and user context, enabling the model to highlight which historical interactions are most influential. On the Lastfm-2K dataset, we select two users for case studies as shown in Fig. 3. For User A, the positive item (artist 173, Hoobastank) receives high attention to the third item in the context (Gotthard), both sharing tags like ”rock” and ”hard rock”, while the negative item (artist 11, Segue) with unrelated tags (”ambient”, ”dub”) shows evenly distributed attention. As a result, the positive item’s prediction score (0.6849) is significantly higher than the negative one (0.3723). Similarly, for User B, the positive item (artist 40, Infected Mushroom, with tags ”electronic”, ”psytrance”) gains strong attention from context-2 and context-3 items with related genres such as Box Car Racer. In contrast, the negative item (artist 1970, Alice Cooper) tagged as ”classic rock” receives less focused attention, confirming the model’s ability to align predictions with semantically coherent context.

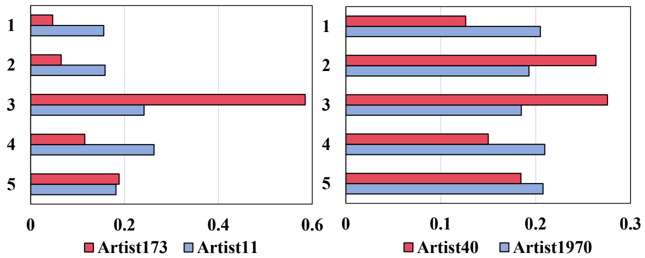
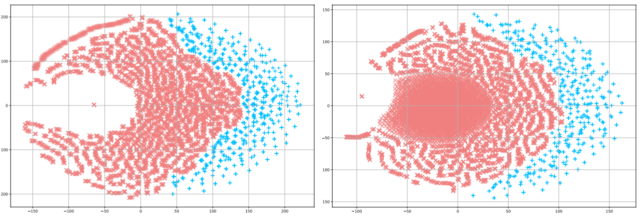


Figure 3: Attention weight of positive and negative samples in 2 clients (Lastfm-2K).



(a) User 178 (b) User 198

Figure 4: TSNE visualization of item embeddings in 2 different clients (Lastfm-2K) learned by *FedCAR*.× denotes negative items, while + denotes positve items.

To further validate the effectiveness of our model, we visualize the learned item embeddings using t-SNE (Maaten and Hinton 2008). As shown in Fig. 4, embeddings from two different clients exhibit clear separation between positive and negative items. This indicates that after training, our model captures meaningful representations that preserve preference signals across heterogeneous user distributions.

| Context | Contrast | MovieLens-100K | | Lastfm-2K | |
|---|---|---|---|---|---|
| | | HR@10 | NDCG@10 | HR@10 | NDCG@10 |
| CentralNCF | | 0.6416 | 0.3682 | 0.8288 | 0.6528 |
| ✘ | ✘ | 0.5027 | 0.2754 | 0.7834 | 0.6908 |
| ✔ | ✘ | 0.5175 | 0.2830 | 0.8196 | 0.7210 |
| ✔ | ✔ | 0.5270 | 0.2837 | 0.8236 | 0.7222 |

Table 3: Ablation Study on MovieLens-100K and Lastfm-2K datasets. ✘ denotes removing the corresponding module while ✔ means keeping it. Please refer to Sec. 4.2

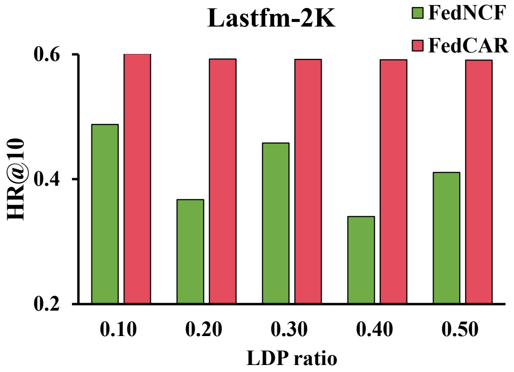


Figure 5: Influence of the LDP ratio in the training of Lastfm-2K. The higher LDP ratio indicates stronger noises.

**Ablation Study.** To better understand *FedCAR*, we conducted ablation experiments on MovieLens-100K and Lastfm-2K to investigate the roles of each component. For the case where both the context encoder and personalized contrastive adaptation are removed, we choose the most recent GPFedRec (Zhang et al. 2024b) for comparison. As can be seen in these two datasets, treating items in such a discrete manner leads to a noticeable performance degradation. This also shows that considering user context can effectively take into account users' consistent behavior patterns. Then, we removed personalized contrastive adaptation separately for comparison. These results highlight the effectiveness of personalized adaptation in helping local models capture fine-grained user preferences, surpassing models trained solely on general behavior patterns.

**Protection with Local Differential Privacy.** To enhance user privacy, we incorporate Local Differential Privacy (LDP) (Choi et al. 2018) into our framework. Specifically, each client perturbs its local model parameters with Laplacian noise, sampled from Laplace$(0, \lambda^{dp})$, before uploading to the server. We evaluate its effect using values $\lambda^{dp} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ on the Lastfm-2K dataset. Since our framework already introduces randomness through personalized contrastive adaptation during local training, it inherently improves robustness to noisy updates. As illustrated in Fig. 5, even under different levels of LDP noise, *FedCAR* consistently outperforms baseline methods, demonstrating its resilience to noise in the aggregation phase.

## 4.3 Discussion

**Concept Difference.** Traditional personalized recommendation often trains user-specific models or discrete embeddings independently for each user. In contrast, our approach employs a local user context encoder to capture short-term behavior without explicitly modeling user identity, reducing privacy risks while enabling adaptive personalization on the client. Centralized sequential recommendation typically uses a global encoder trained on aggregated user sequences, often ignoring user identity. Federated recommendation, by contrast, preserves privacy by keeping data and user-specific parameters local. Our context-aware encoder fits this setting by embedding recent user interactions locally, supporting lightweight, privacy-preserving personalization aligned with federated learning principles.

**Privacy in Federated Recommendation.** Our framework builds upon the decentralized nature of federated learning to preserve data locality and minimize exposure of user information, aligning with privacy regulations like GDPR (Voigt and Bussche 2017). By locally encoding user context and avoiding explicit user embeddings, we reduce the risk of identity inference. Furthermore, the structure of the model allows for seamless integration with privacy enhancement techniques (e.g., Differential Privacy), such as adding noise to the embeddings of the items and the context encoder before communication, further strengthening the protection of the privacy of the system.

**Limitation.** Although the proposed method performs well under federated settings, all evaluated approaches still face challenges in extremely sparse interaction scenarios, indicating room for improvement. In addition, while the framework is lightweight, its scalability across diverse client capacities and recommendation tasks has not been fully validated.

# 5 Conclusion

In this work, we propose *FedCAR*, a novel federated recommendation framework that integrates local context encoding and personalized contrastive adaptation to address key challenges in personalization and interpretability. By attending to recent user behavior patterns, our method offers a lightweight yet expressive alternative to traditional discrete personalization, enabling more accurate and explainable predictions without compromising user privacy. Extensive experiments across multiple datasets demonstrate the superiority of our approach in both recommendation performance and representation quality. Case studies and embedding visualizations further validate the model's ability to generate meaningful predictions and semantic alignment across clients. Our work highlights the potential of context-aware and contrastive learning designs in advancing personalized federated recommendation. Further exploration can be directed toward cross-domain recommendation tasks and improving robustness in sparse or low-resource scenarios.

## Acknowledgments

We thank the support of National Natural Science Foundation of China under Grant Nos. 62032016, 623B2080 and 62272412.

## References

Acar, D. A. E.; Zhao, Y.; Navarro, R. M.; Mattina, M.; Whatmough, P. N.; and Saligrama, V. 2021. Federated Learning Based on Dynamic Regularization. In *ICLR*.

Agrawal, N.; Sirohi, A. K.; Kumar, S.; and , J. 2024. No Prejudice! Fair Federated Graph Neural Networks for Personalized Recommendation. In *AAAI*.

Ammad-ud-din, M.; Ivannikova, E.; Khan, S. A.; Oyomno, W.; Fu, Q.; Tan, K. E.; and Flanagan, A. 2019. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System. *CoRR*, abs/1901.09888.

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

Chai, D.; Wang, L.; Chen, K.; and Yang, Q. 2021. Secure Federated Matrix Factorization. *IEEE Intell. Syst.*

Chen, G.; Zhang, X.; Su, Y.; Lai, Y.; Xiang, J.; Zhang, J.; and Zheng, Y. 2023. Win-Win: A Privacy-Preserving Federated Framework for Dual-Target Cross-Domain Recommendation. *AAAI*.

Chen, T.; Huang, W.; Wang, Z.; Shi, Z.; Li, H.; Dong, W.; Ye, M.; Du, B.; and Xu, Y. 2025. Pixel-wise Divide and Conquer for Federated Vessel Segmentation. In *IJCAI*.

Choi, W.-S.; Tomei, M.; Vicarte, J. R. S.; Hanumolu, P. K.; and Kumar, R. 2018. Guaranteeing Local Differential Privacy on Ultra-Low-Power Systems. In *ISCA*.

Dai, Y.; Chen, Z.; Li, J.; Heinecke, S.; Sun, L.; and Xu, R. 2023a. Tackling Data Heterogeneity in Federated Learning with Class Prototypes. *AAAI*.

Dai, Y.; Chen, Z.; Li, J.; Heinecke, S.; Sun, L.; and Xu, R. 2023b. Tackling Data Heterogeneity in Federated Learning with Class Prototypes. *AAAI*.

Diao, Y.; Li, Q.; and He, B. 2024. Exploiting Label Skews in Federated Learning with Model Concatenation. *AAAI*.

Harper, F. M.; and Konstan, J. A. 2015. The MovieLens Datasets: History and Context. *ACM TiiS*.

He, X.; Chen, T.; Kan, M.-Y.; and Chen, X. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM*.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural Collaborative Filtering. In *WWW*.

Hong, J.; Wang, H.; Wang, Z.; and Zhou, J. 2023. Federated Robustness Propagation: Sharing Adversarial Robustness in Heterogeneous Federated Learning. *AAAI*.

Hu, M.; Cao, Y.; Li, A.; Li, Z.; Liu, C.; Li, T.; Chen, M.; and Liu, Y. 2024. FedMut: Generalized Federated Learning via Stochastic Mutation. In *AAAI*.

Hu, P.; Lin, Z.; Pan, W.; Yang, Q.; Peng, X.; and Ming, Z. 2023. Privacy-preserving graph convolution network for federated item recommendation. *Artificial Intelligence*.

Hu, P.; Yang, E.; Pan, W.; Peng, X.; and Ming, Z. 2022. Federated one-class collaborative filtering via privacy-aware non-sampling matrix factorization. *KBS*.

Huang, W.; Ye, M.; and Du, B. 2022. Learn from Others and Be Yourself in Heterogeneous Federated Learning. In *CVPR*.

Huang, W.; Ye, M.; Shi, Z.; and Du, B. 2023a. Generalizable Heterogeneous Federated Cross-Correlation and Instance Similarity Learning. *IEEE PAMI*.

Huang, W.; Ye, M.; Shi, Z.; Li, H.; and Du, B. 2023b. Rethinking Federated Learning with Domain Shift: A Prototype View. In *CVPR*.

Huang, W.; Ye, M.; Shi, Z.; Wan, G.; Li, H.; Du, B.; and Yang, Q. 2024. A Federated Learning for Generalization, Robustness, Fairness: A Survey and Benchmark. *IEEE PAMI*.

Jiang, J.; Zhang, C.; Zhang, H.; Li, Z.; Li, Y.; and Yang, B. 2025. A Tutorial of Personalized Federated Recommender Systems: Recent Advances and Future Directions. In *Companion Proceedings of the ACM on Web Conference 2025*.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *ICDM*, 197–206.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. SCAFFOLD: stochastic controlled averaging for federated learning. In *ICML*.

Konečný, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR*, abs/1610.02527.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*.

Li, M.; Zhang, Z.; Zhao, X.; Wang, W.; Zhao, M.; Wu, R.; and Guo, R. 2023. Automlp: Automated mlp for sequential recommendations. In *WWW*, 1190–1198.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *ICDE*.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *CVPR*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020a. Federated Learning: Challenges, Methods, and Future Directions. *IEEE SPM*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020b. Federated Optimization in Heterogeneous Networks. In *MLSys*.

Liang, F.; Pan, W.; and Ming, Z. 2021. FedRec++: Lossless Federated Recommendation with Explicit Feedback. *AAAI*.

Lin, G.; Liang, F.; Pan, W.; and Ming, Z. 2021. FedRec: Federated Recommendation With Explicit Feedback. *IEEE Intell. Syst.*

Liu, B.; Li, D.; Wang, J.; Wang, Z.; Li, B.; and Zeng, C. 2024. Integrating user short-term intentions and long-term preferences in heterogeneous hypergraph networks for sequential recommendation. *IPM*.

Liu, W.; Chen, C.; Liao, X.; Hu, M.; Yin, J.; Tan, Y.; and Zheng, L. 2023. Federated probabilistic preference distribution modelling with compactness co-clustering for privacy-preserving multi-domain recommendation. In *IJCAI*.

Long, G.; Tan, Y.; Jiang, J.; and Zhang, C. 2020. *Federated Learning for Open Banking*, 240–254. Springer International Publishing.

Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; and Feng, J. 2021. No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data. In *NeurIPS*.

Luo, S.; Xiao, Y.; and Song, L. 2022. Personalized Federated Recommendation via Joint Representation Learning, User Clustering, and Model Adaptation. In *CIKM*.

Ma, X.; Zhu, J.; Lin, Z.; Chen, S.; and Qin, Y. 2022. A state-of-the-art survey on solving non-IID data in Federated Learning. *Future Gener Comput Syst*.

Ma, Y.; Dai, W.; Huang, W.; and Chen, J. 2025. Geometric Knowledge-Guided Localized Global Distribution Alignment for Federated Learning. In *CVPR*.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.

Mu, X.; Shen, Y.; Cheng, K.; Geng, X.; Fu, J.; Zhang, T.; and Zhang, Z. 2023. FedProc: Prototypical contrastive federated learning on non-IID data. *Future Gener Comput Syst*.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP*.

Perifanis, V.; and Efraimidis, P. S. 2022. Federated Neural Collaborative Filtering. *KBS*.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *AoMS*.

Singhal, K.; Sidahmed, H.; Garrett, Z.; Wu, S.; Rush, J.; and Prakash, S. 2021. Federated reconstruction: Partially local federated learning. *NeurIPS*.

Sun, Z.; Xu, Y.; Liu, Y.; He, W.; Kong, L.; Wu, F.; Jiang, Y.; and Cui, L. 2024. A Survey on Federated Recommendation Systems. *IEEE TNNLS*.

T. Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized Federated Learning with Moreau Envelopes. In *NeurIPS*.

Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023. Federated Learning on Non-IID Graphs via Structural Knowledge Sharing. In *AAAI*.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *AAAI*.

Vaswani, A.; Shazeer, N.; Parmar, N.; and Uszkoreit. 2017. Attention is all you need. *NeurIPS*, 30.

Voigt, P.; and Bussche, A. v. d. 2017. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated.

Wang, Z.; Bai, H.; Huang, W.; Li, D.; Wang, J.; and Li, B. 2025a. Federated Recommendation with Explicitly Encoding Item Bias. *AAAI*.

Wang, Z.; Huang, W.; Chen, T.; Shi, Z.; Wan, G.; Qiao, Y.; Yang, B.; Wang, J.; Li, B.; and Ye, M. 2025b. An Empirical Study of Federated Prompt Learning for Vision Language Model. In *IJCAI*.

Wu, C.; Wu, F.; Lyu, L.; Qi, T.; Huang, Y.; and Xie, X. 2022. A federated graph neural network framework for privacy-preserving personalization. *NC*.

Zhang, C.; Long, G.; Guo, H.; Fang, X.; Song, Y.; Liu, Z.; Zhou, G.; Zhang, Z.; Liu, Y.; and Yang, B. 2024a. Federated Adaptation for Foundation Model-based Recommendations. In *IJCAI*.

Zhang, C.; Long, G.; Guo, H.; Liu, Z.; Zhou, G.; Zhang, Z.; Liu, Y.; and Yang, B. 2025a. Multifaceted user modeling in recommendation: A federated foundation models approach. In *AAAI*.

Zhang, C.; Long, G.; Zhang, Z.; Li, Z.; Zhang, H.; Yang, Q.; and Yang, B. 2025b. Personalized recommendation models in federated settings: A survey. *IEEE TKDE*.

Zhang, C.; Long, G.; Zhou, T.; Yan, P.; Zhang, Z.; Zhang, C.; and Yang, B. 2023. Dual Personalization on Federated Recommendation. In *AAAI*.

Zhang, C.; Long, G.; Zhou, T.; Zhang, Z.; Yan, P.; and Yang, B. 2024b. GPFedRec: Graph-Guided Personalization for Federated Recommendation. In *ACM SIGKDD*.

Zhang, L.; Luo, Y.; Bai, Y.; Du, B.; and Duan, L.-Y. 2021. Federated Learning for Non-IID Data via Unified Feature Learning and Optimization Objective Alignment. In *ICCV*.

Zhang, Z.; Liu, S.; Liu, Z.; Zhong, R.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Q.; and Jiang, P. 2025c. Llm-powered user simulator for recommender system. In *AAAI*.

Zhang, Z.; Liu, S.; Yu, J.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Z.; Liu, Q.; Zhao, H.; Hu, L.; et al. 2024c. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *SIGIR*.