Federated User Modeling from Hierarchical Information

QI LIU, JINZE WU, ZHENYA HUANG, HAO WANG, YUTING NING, MING CHEN, and ENHONG CHEN, Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China and State Key Laboratory of Cognitive Intelligence, China
JINFENG YI and BOWEN ZHOU, JD AI Research, China

The generation of large amounts of personal data provides data centers with sufficient resources to mine idiosyncrasy from private records. User modeling has long been a fundamental task with the goal of capturing the latent characteristics of users from their behaviors. However, centralized user modeling on collected data has raised concerns about the risk of data misuse and privacy leakage. As a result, federated user modeling has come into favor, since it expects to provide secure multi-client collaboration for user modeling through federated learning. Unfortunately, to the best of our knowledge, existing federated learning methods that ignore the inconsistency among clients cannot be applied directly to practical user modeling scenarios, and moreover, they meet the following critical challenges: (1) *Statistical heterogeneity*. The distributions of user data in different clients are not always **independently identically distributed (IID)**, which leads to unique clients with needful personalized information; (2) *Privacy heterogeneity*. User data contains both public and private information, which have different levels of privacy, indicating that we should balance different information shared and protected; (3) *Model heterogeneity*. The local user models trained with client records are heterogeneous, and thus require a flexible aggregation in the server; (4) *Quality heterogeneity*. Low-quality information from inconsistent clients poisons the reliability of user models and offsets the benefit from high-quality ones, meaning that we should augment the high-quality information during the process. To address the challenges, in this paper, we first propose a novel client-server architecture framework, namely **Hierarchical Personalized Federated Learning (HPFL)**, with a primary goal of serving federated learning for user modeling in inconsistent clients. More specifically, the client trains and delivers the local user model via the hierarchical components containing hierarchical information from privacy heterogeneity to join collaboration in federated learning. Moreover, the client updates the personalized user model with a fine-grained personalized update strategy for statistical heterogeneity. Correspondingly, the server flexibly aggregates hierarchical components from heterogeneous user models in the case of privacy and model heterogeneity with a differentiated component aggregation strategy. In order to augment high-quality information and generate high-quality user models, we expand HPFL to the **Augmented-HPFL (AHPFL)** framework by incorporating the augmented mechanisms, which filters out low-quality information such as noise, sparse information

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2021YFF0901003), and the National Natural Science Foundation of China (Grants No. 61922073, U20A20229, and 62106244).
Authors' addresses: Q. Liu, J. Wu, Z. Huang (corresponding author), H. Wang, Y. Ning, M. Chen, and E. Chen, Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China and State Key Laboratory of Cognitive Intelligence, No. 443, Huangshan Road Shushan District, Hefei, China, 230027; emails: qiliuql@ustc.edu.cn, hxwjz@mail.ustc.edu.cn, huangzhy@ustc.edu.cn, wanghao3@ustc.edu.cn, ningyt@mail.ustc.edu.cn, chenming166613@mail.ustc.edu.cn, cheneh@ustc.edu.cn; J. Yi and B. Zhou, JD AI Research, No. 18, Kechuang 12th Street, Yizhuang District, Beijing, China, 100176; emails: yijinfeng@jd.com, bowen.zhou@jd.com.

1046-8188/2023/03-ART46 $15.00
https://doi.org/10.1145/3560485

and redundant information. Specially, we construct two implementations of AHPFL, i.e., AHPFL-SVD and AHPFL-AE, where the augmented mechanisms follow **SVD (singular value decomposition)** and **AE (autoencoder)**, respectively. Finally, we conduct extensive experiments on real-world datasets, which demonstrate the effectiveness of both HPFL and AHPFL frameworks.

CCS Concepts: • **Security and privacy** → *Privacy protections* • **Information systems** → *Data mining*;

Additional Key Words and Phrases: User modeling, federated learning, privacy heterogeneity, model personalization, augmented mechanism

ACM Reference format:
Qi Liu, Jinze Wu, Zhenya Huang, Hao Wang, Yuting Ning, Ming Chen, Enhong Chen, Jinfeng Yi, and Bowen Zhou. 2023. Federated User Modeling from Hierarchical Information. *ACM Trans. Inf. Syst.* 41, 2, Article 46 (March 2023), 33 pages.
https://doi.org/10.1145/3560485

1 INTRODUCTION

Massive personalized apps have gained popularity over the past decades, benefiting from the development of personal devices. Simultaneously, an unprecedented amount of data is being generated on individual devices at an astonishing speed [68]. Naturally, this vast amount of data is effectively analyzed to model the users, leading user modeling to become fundamental to capture useful potential characteristics that rely on personal data [19, 110]. For example, in intelligent education systems, user modeling assists cognitive diagnosis for modeling student capacities [88]; moreover, in e-commerce recommender systems, user modeling helps to model customer preferences [30, 75]. Generally speaking, user modeling processes centralized training with collected data, much of which is private, which results in privacy leakage. Considering the privacy and sensitivity of personal data, regulations such as the **General Data Protection Regulation (GDPR)** have been enacted to restrict the centralized use of private data [7, 84, 87]. In this case, data that is forced to remain local (e.g., personal devices) becomes isolated [54, 90]. By focusing on this dilemma, **federated learning (FL)** has attracted widespread attention owing to its ability to achieve secure distributed user modeling [65, 69]. This approach involves building and aggregating user models while leaving private data isolated, thereby preserving the data privacy [60].

Broadly speaking, standard federated learning is a centralized system for client collaboration, following a client-server architecture, as shown in Figure 1(a). There are two participants in federated learning, specifically the client and server. In federated learning, the local data in the clients are not required, while only the models are shared among participants. In particular, the client trains and delivers models only with the local data, while the server aggregates the homogeneous local models to a global one and distributes it back. In the case of isolated data, federated learning guarantees user privacy protection on the client-level by silos. In the literature, many efforts have been made to improve federated learning from a technical perspective, such as FedSGD, FedAvg [60], FedAtt [38], and FedProx [47]. Though some great performances have been achieved, previous works are "dance in shackles", that is, current federated learning frameworks are designed for consistent clients [28]. Concretely speaking, on one hand, researchers assume that the data in the clients are consistent, i.e., **independent and identically distributed (IID)**. On the other hand, researchers simply initialize the local models of various clients with consistent structures. These assumptions limit the ability of federated learning to adapt to inconsistent clients in user modeling scenarios, where there are clients with variant information and models based on different usage habits [43]. It is therefore necessary to design a superior federated learning framework that better fits in federated user modeling tasks for isolated data from inconsistent clients [39].

Nevertheless, the particularity of user modeling with inconsistent clients leads to the following multilevel challenges: (1) *Statistical heterogeneity*: Unlike the traditional scenarios in which the

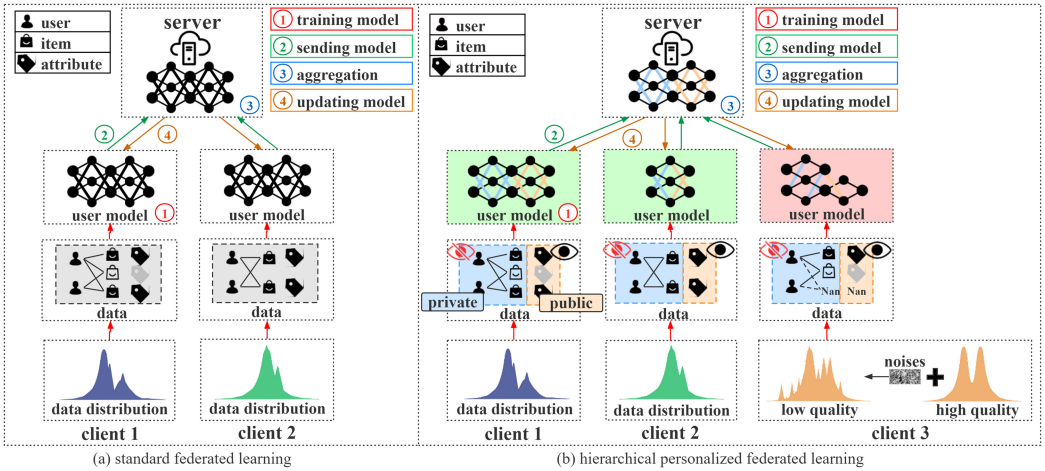


Fig. 1. Differences between standard federated learning (left), and our hierarchical personalized federated learning (right) for user modeling.
Standard FL simply aggregates and updates the consistent entire user models indiscriminately, while HPFL partitions and processes the different components of the heterogeneous models independently. The top part illustrates a server with a global model. The bottom presents clients with Non-IID data and local user models. Each round consists of four steps: training a model locally, sending the model to the server, model aggregation in the server, and updating models for clients.

data is assumed to be IID [5], personal records for user modeling are usually **non-independently identically distributed (Non-IID)**, which is the source of personalization among population [48]. For example, as shown in Figure 1, the preferences of users in the client 1 are focused on the items belonging to two different regions, while the users in the client 2 prefer other items from a certain region. The methods training local models to cover all clients based on the consistent global model, inevitably eliminate the personalization of clients and reduce the ability of user models to depict user characteristics [70]. Accordingly, it is necessary to integrate the personalized information of user models to adapt statistical heterogeneity; (2) *Privacy heterogeneity*: As suggested by [2, 21], different types of information have different levels of privacy. For example, as shown in Figure 1, the attribute information of items (e.g., labels and categories) in the clients are relatively public, as they are summarized from prior domain knowledge and rules and are consistent for all clients [49]. As we know, for any customer, the labels of a T-shirt (like color and style) are consistent. While information such as representations of users in user models are strictly private since they reflect the user idiosyncrasy [10]. On one hand, in order to expand more information, rashly sharing representations will bring the risk of exposing privacy [31]. On the other hand, discarding the sensitive information to protect privacy will lead to a loss of available information. Therefore, we should securely apply specialized federated learning settings to the hierarchical information with privacy heterogeneity so that we can balance the information to be protected or shared across user models. Based on the partition of clients for privacy, federated learning can be divided into two scenarios: one is cross-device federated learning, that is, data is only stored in personal devices with strict privacy requirements; the other is the cross-silo federated learning with general privacy requirements, that is, data can be centralized in some organizations or geo-distributed datacenters and each client could control the data from a number of their respective users [43]. In our paper, we focus on the cross-silo federated learning scenario with the different granularity of private information; (3) *Model heterogeneity*: The mainstream federated learning methods use consistent

user models to model all locals [43]. However, in practical user modeling applications, due to the different properties of the private data, the user model structures are often different among different clients [46]. As shown in Figure 1, the different item spaces browsed by users leads to differences in the structures of user models. Therefore, the strategy for flexibly processing heterogeneous user models in federated learning also requires a careful design.

To tackle these challenges, first in our preliminary work [97], we proposed a novel personalized federated learning framework for user modeling, called **Hierarchical Personalized Federated Learning (HPFL)** framework. Generally speaking, HPFL followed the client-server architecture as shown in Figure 1(b). While in the client stage of HPFL, a two-stage process was applied to balance the protecting and sharing of information. In the first phase, the client defined the hierarchical information as public information and private information from privacy heterogeneity in personal data. Accordingly, a general user model with hierarchical components, consisting of both public component and private component was initialized and trained in each client. Afterward, the client uploaded the local user model by components; more specifically, it directly uploaded the public component, while delivering the drafts of the private component instead of the original information to safeguard the data privacy. In the second phase in the client, the client updated the local user model by a proposed fine-grained personalized update strategy. The corresponding components in the client were updated from a fusion of both the local user model and global model from the server. As for the server stage, the server flexibly aggregated inconsistent local user models by components using a differentiated component aggregation strategy. For the public components, the server executed a weighted aggregation on the same attribute. Correspondingly, for the private component, the server aggregated the universe of local drafts by clustering on representations to widely expand the available information. In client, we took both the expansion of the user model knowledge at the global perspective and the inheritance of user model personalization at the local perspective into consideration, which accommodated statistical heterogeneity. Moreover, in the server, we safely aggregated inconsistent user models by different components, enabling us to address both privacy heterogeneity and model heterogeneity.

In HPFL, we carried out federated learning on inconsistent clients to directly aggregate and update the extensive information by components. As a result, it ignored another possible quality heterogeneity challenge among the clients: (4) *Quality Heterogeneity*: The success of deep learning-based user modeling method is highly dependent on data quality [55, 96]. In real scenarios, the isolated data from the clients is not always sufficient and of high quality. Instead, there is partial low-quality information, such as noise, sparse information and redundant information, which may result in low-quality user models. For example, as shown in Figure 1(b), the data quality varies from client to client. Data quality in Client 1 and 2 may be higher due to the purer data with few noises and guide to higher quality user models, while the data in Client 3 may be noisier, which eventually leads to low-quality user models. Once the low-quality local models participate in the aggregation process in federated learning, all user models are later damaged, ultimately reducing the effectiveness of user modeling. Obviously, there is quality heterogeneity in federated user modeling. Therefore, we should pay more attention to the high-quality information in the user model rather than harnessing all information where the quality is not even.

In this paper, based on our previous HPFL, we further propose an extended framework, **Augmented-HPFL (AHPFL)** to augment the high-quality information of different clients in federated learning, considering the quality heterogeneity caused by inconsistent clients. More specifically, during the client updating and server aggregation process, rather than using the original heuristic method, we reprovingly augment the information from clients with the augmented mechanisms so as to refine the high-quality information. In particular, we implement the augmented mechanisms in the AHPFL framework, following some existing typical augmented mechanisms.

The first one is a matrix decomposition-based method, i.e., AHPFL-SVD with singular value decomposition, which decomposes and truncates all information to lossily compress the original information into a higher-quality one. The second is a neural network-based method, i.e., AHPHL-AE with autoencoder, which extracts valuable high-quality information from the original information.

Finally, we conduct extensive experiments on some typical user modeling scenarios, including intelligent education and recommendation systems. More specifically, the experiments are specialized on the student capacities modeling and user preferences modeling task. Experimental results on real-world datasets clearly demonstrate the superiority of our HPFL and AHPFL frameworks in user modeling tasks in terms of quantitative performances and modeling rationality as well as the augmented extent of AHPFL. To the best of our knowledge, HPFL is the first comprehensive attempt to serve federated learning for user modeling by components in user models. Furthermore, AHPFL is designed to augment high-quality information from the inconsistent clients in federated user modeling.

2 RELATED WORK

In this section, we review the related works from three aspects in terms of user modeling, federated learning, and augmented mechanism.

2.1 User Modeling

User modeling is a fundamental task, with the goal of analyzing behavioral information and inferring the unobservable characteristics such as capability, preference, habit, and tendency [110]. User modeling has been widely used to model rich user characteristics in various applications. For example, based on user capability fitting, researchers have employed user modeling to model user vision level [16], lawyer expertise [71], and gamer competitiveness [104]; based on user preference mining, researchers have also applied user modeling to tasks such as personalized search [59, 74, 100], restaurant recommendation [107], news recommendation [17], social network [98, 105], and other broad recommendation tasks [50, 76, 80, 108]. To effectively model users based on large amounts of data, some artificial neural network-based user modeling methods have been developed and applied into some personalized user modeling tasks.**Neural Cognitive Diagnosis (NeuralCD)** [88] provides a general cognitive diagnosis paradigm for fitting student cognitive abilities in intelligent education system. **Neural Collaborative Filtering (NCF)** [30] generalizes traditional matrix factorization to mine the user interest preferences for collaborative filtering recommendation. These methods establish a set of user modeling processing based on neural networks to mine unobservable information of users and build the hidden relationship between users and items in some particular scenarios.

Currently, most of the existing user modeling methods expect to fully optimize models with the centralized training process. However, there are growing concerns regarding the risks of revealing the user data privacy, leading to obstacles in practical applications. Therefore, we raise the federated user modeling task, which aims to process user modeling for isolated and inconsistent clients via federated learning technique.

2.2 Federated Learning

Federated learning (FL) has become a promising machine learning technique in recent years. Federated learning was first proposed to solve the problem of model updating in distributed mobile terminals [60]. Clients in the federated learning process are asked to train and deliver their models independently. An aggregated global model from the server is then provided to update the local models. In the process, the data is kept locally, and only the models participate in the collaboration, meaning that the workflow guarantees data isolation and privacy protection. Owing to the growing

focus on privacy protection, regulations such as the **General Data Protection Regulation (GDPR)** have been developed that limit the collection and use of personal data, meaning that federated learning has received extensive attention. From a technical perspective, existing federated learning frameworks can be categorized into three types [102, 103], i.e., horizontal federated learning, vertical federated learning, and federated transfer learning. Specifically, horizontal federated learning adapts to the cases where the data in the different clients shares the same feature space of items, but the users are different; in vertical federated learning, the data shares the same user space, but the feature spaces of items are not quite overlapped [13]; finally, federated transfer learning faces the scenarios where both the feature spaces and the user spaces are inconsistent [41, 52, 53].

Since then, some model fusion methods for improving the process have been proposed [39]. FedSGD and FedAvg [60] train the local model in parallel. The server here simply generates a global model by the weighted average of the local model parameters according to data sizes. Attentive aggregation method, FedAtt [38] considers the different importances of local models and aggregates local models by applying a layer-wise soft attention mechanism between local models and the global model; And FedAmp [34] employs federated attentive message passing to facilitate similar clients, which does not use a single global model on the cloud server to conduct collaborations. Regularization method, FedProx [47] subjoins a proximal term to close the local model and the global model, which aims to avoid excessive drift during optimization.

However, current works are proposed based on the assumption of consistent clients and accordingly provide a uniform model for all clients, which is out of operation in practical scenarios [58]. Moreover, the methods discussed above still introduce the risk of privacy leakage, especially when the models submitted contain sensitive representation information, e.g., the user representations in user models. Unfortunately, the common privacy protection method, differential privacy federated learning [23, 61], faces the dilemma of that the confidentiality and accuracy are not entirely available simultaneously [8, 33, 82]. Therefore, some difficulties in applying federated learning still remain.

2.3 Augmented Mechanism

Data is often corrupted, therefore some augmented mechanisms which aim to augment useful information and erase low-quality information such as noise, sparse information, and redundant information from the original data, receive extensive attention [77]. In recent years, some augmented mechanisms have been widely used in audio augmented [44] and image augmented tasks [4]. Regarding tasks of this kind, **SVD (singular value decomposition)**-based methods [1, 18] and **AE (autoencoder)**-based methods [85, 86] are the typical ones of the representative works. SVD is a matrix decomposition-based method, which decomposes the original data into different components according to singular value decomposition and selects some effective components to reconstruct new significant information. While AE is a self-supervised deep learning method comprising both an encoder and decoder. The former encodes the original representation into the hidden representation, while the latter decodes the hidden representation into the original representation with the goal of minimizing the reconstruction error. Ultimately, the autoencoder has the ability to extract more valuable information from the original data while avoiding noise. In this paper, we adopt these classical augmented methods to filter low-quality information from inconsistent clients in federated learning, so as to obtain high-quality user models.

3 PRELIMINARIES

In this section, we first formally introduce the federated user modeling issue and define the hierarchical information in the isolated scenarios. We then consider two real scenario specifications for user modeling.

3.1 Problem Definition

Before outlining the framework design, we formally provide the issue of federated user modeling. It is a cross-silo federated learning scenario. In the federated user modeling scenario, there are $|C|$ clients which store local user data independently. In a specific client c, there are $|U_c|$ users and $|V_c|$ items, which can be represented as: $U_c = \{u_1, u_2, \ldots\}$ and $V_c = \{v_1, v_2, \ldots\}$. The attributes of items from different clients are consistent as items of the same type share the labels. In all, there are K attributes. Moreover, the users interact with the items in the same client, generating $|R_c|$ interaction records. These interaction records consist of the triplet (u, v, g), which contains user u, item v of attribute k and their interaction result g. To tackle the problem in this paper, we expect to obtain $|C|$ local user models, i.e., $\{\Theta_1, \Theta_2, \ldots\}$ for each client, where the c-th user model $\Theta_\mathbf{c}$ can model the potential characteristics of users in the client c for predicting the interaction results while maintaining data isolation.

As mentioned earlier, clients are inconsistent in the federated user modeling scenarios. For federated user modeling on inconsistent clients, we define different information based on privacy heterogeneity. Further, we divide the information of both data and model according to it. In the real world, some knowledge information is public and shared among all clients, such as attributes of items, which are relatively public and can be freely communicated. In addition, clients also tightly hold some strictly private information in personal data for privacy protection, such as distributions of users and items. In order to reasonably utilize information with different degrees of privacy intensity as much as possible while avoiding the risk of privacy disclosure, we define hierarchical information to balance the protection and utilization, which is denoted as follows:

Public information, Inf_k. It contains the prior domain knowledge which is shared among clients, such as the K attributes from R_c. In this case, the public information is relatively private and incompetent to expose the sensitive information of users. Obviously, the owner of the data will tolerate the direct access to and use of the public information.

Private information, Inf_r. It refers to the sensitive information which is proprietary for clients, such as the interaction in R_c. The private information is generated from the unique distributions of users and items among each client, which represents specific characteristic or preference information. Apparently, it is expected to be strictly protected.

Notably, these two kinds of information are also used in the user model. Therefore, each local user model Θ can be divided into public components (denoted as Θ_k) for public information and private components (denoted as Θ_r) corresponding to the hierarchical information. In practical scenarios, the user data in devices is too proprietary to be collected so that it is hard to conduct centralized training, which results in isolated and inconsistent user modeling.

3.2 Scenario Specifications

User modeling has been applied in many scenarios, including education, e-commerce, catering, and so on. In this work, two representative issues in real user modeling scenarios are chosen and our frameworks are applied to them. The first task is to model user capability such as student knowledge state in education. The goal of the task is to predict the student performances [51]. Correspondingly, the user u, item v and information of K attributes in our problem are denoted as the student, question, and knowledge concepts in the question, respectively. The result of interactive behavior g is the student's response to the question, and the target in this scenario is to model student's mastery of questions and predict the student performances. Then the other task is to model user preferences in scenarios such as recommendation. This task is regarded as customer rating prediction. Similarly, user u, item v and K attributes are here viewed as customer, product, and product categories. The interactive behavior is the user evaluation of the product. The ultimate

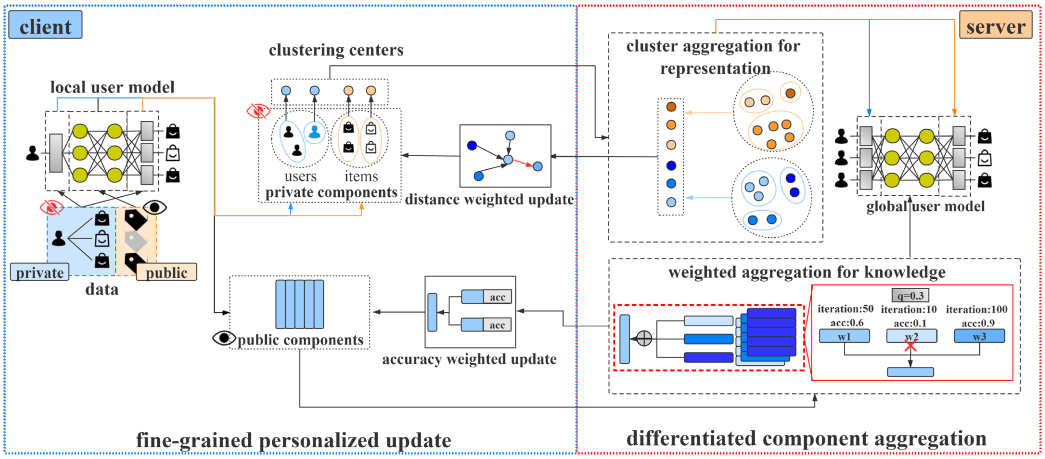


Fig. 2. Hierarchical Personalized Federated Learning framework, HPFL.

objective is to model customers' interests to predict the users' ratings on items. Please note that, in our paper, we focus on the cross-silo federated learning scenario. In that case, the clients can be regarded as the data silos or data centers, thus there is more than one user in a client.

4 HIERARCHICAL PERSONALIZED FEDERATED LEARNING

In this section, we describe our Hierarchical Personalized Federated Learning (HPFL) framework [97] for user modeling in more detail. Specifically, we first sketch out an overview of our framework. We then cover the technical details of the client and server design. Next we design a **general user model** as the local user model for hierarchical information, namely **GUM**. Finally, we review the entire workflow of HPFL.

4.1 Model Overview

Heterogeneities impede user collaboration in federated learning [47]. To resolve this dilemma, we propose a novel Hierarchical Personalized Federated Learning (HPFL) framework as illustrated in Figure 2. HPFL is developed from the client-server architecture of federated learning. The client is responsible for training a simple while proprietary user model with private records, that is a general user model (GUM) in our framework. Besides, it delivers the different components of the user model and updates a personalized user model using the fine-grained personalized update strategy based on the global model received. The server is in charge of fusing heterogeneous local user models to a global one by different components with the differentiated component aggregation strategy. In a nutshell, the client maintains the personalization from Non-IID user data and the server allows to aggregate different components of the heterogeneous user models without compromising privacy. We will introduce the technical details of the two parts in the following subsections, respectively.

4.2 Client Design

An isolated client only holds and processes its own data. While in our framework, the client participates the federated learning workflow by two phases: one is to upload the trained local user model, and the other is to obtain the global model to update its personalized user model. Specially, the client first independently initializes and trains a general user model named GUM. The GUM contains both the public and private components, which are designed for hierarchical information

(GUM will be introduced in more detail in Section 4.4). The user model is trained with only local data and aims to model local user characteristics appropriately.

To join the federated learning with an upload phase, the client delivers the local user model. Specially, a submission by different components that makes full use of the information is adopted. In particular, the public component with public information is delivered directly. While the privacy component is maintained by the client, since it is sensitive and its centralized use can lead to privacy leaks. Instead, the client only provides some drafts, which are generated as a rough estimation for user or item representations. Specifically, as shown in Figure 2, the client is required to process a clustering task on Θ_r to obtain the local cluster centers as drafts, which are representative for representations in the user model, but low sensitive. In some special situations, such as only one user in a client, we can simply mask the embedding with noises to generate the drafts, or we can maintain the user embedding while only delivering the drafts of item embeddings. Then the two components will be aggregated in the server, which will be introduced in Section 4.3.

To obtain the benefit from federated learning with update phase, after accepting the aggregated global model, the client is mainly responsible for absorbing the abundant information from the global model to update the local GUM. An appropriate user model will then be applied to further applications. However, clients in federated user modeling have personalized information due to statistical heterogeneity. Methods such as model interpolation [58, 63] are utilized to retain personalized information and customize user models for clients. To some extent, it enhances the adaptability of the framework to inconsistent scenarios. Unfortunately, since the black-box model interpolation may introduce poor results [3], we regulate a fine-grained personalized update strategy to fuse the local personalized information and global generalized information into GUMs by different components as shown in Algorithm 1. The fusion process is in reference to a certain weight, which can reflect the importance of the local model. Note that the comparison on weights is not our focus. Therefore, we choose local test accuracy as an intuitive dynamic weight in principle.

For public component in GUM, that is, the knowledge vectors of attributes, at round t, client i adds the local attribute knowledge vector $\mathbf{c}_{k,i}^t$ and the global attribute knowledge $\mathbf{c}_k^{t,g}$ on attribute k via the corresponding accuracy $Acc_{k,i}^t$ to accuracy-weighted update the new knowledge vector as:

$$\mathbf{c}_{k,i}^t = \mathbf{c}_{k,i}^t \times Acc_{k,i}^t + \mathbf{c}_k^{t,g} \times (1 - Acc_{k,i}^t). \tag{1}$$

Apparently, the better the performances achieved by the local attribute knowledge, the more likely it is to retain its own knowledge. However, the poor knowledge vector that requires a further optimization will be out of the local optima after synthesizing the global information.

Correspondingly, for private component in GUM, that is, the user and item representations, the client i distance-weighted updates new representation with all of the N global cluster centers to facilitate extensive collaboration with other participants, via the distances between the j-th local embedding in the client i ($\mathbf{Emb}_{j,i}$) and global centers Θ_r^g. The update process can be denoted as follows:

$$\begin{aligned} \mathbf{Emb}^g &= \sum_{n=1}^{N} \frac{||\mathbf{Emb}_{j,i} - \Theta_{r,n}^g|| \times \Theta_{r,n}^g}{\sum_{m=1}^{N} ||\mathbf{Emb}_{j,i} - \Theta_{r,m}^g||}, \\ \mathbf{Emb}_{j,i} &= \mathbf{Emb}_{j,i} \times \mathbf{Acc}_i + \mathbf{Emb}^g \times (1 - \mathbf{Acc}_i). \end{aligned} \tag{2}$$

In particular, $\mathbf{Acc}_i$ is an accuracy vector, representing the accuracy of the local model on each attribute. Similarly, we maintain the more accurate representations and initialize the more incompetent representation to a general one before the next local iteration.

ALGORITHM 1: Fine-grained Personalized Update.

Input: The aggregated global public components, Θ_k^g; The aggregated global private components, Θ_r^g; The original local public components, Θ_k and private components, Θ_r;
Output: The updated local public components, Θ_k and private components, Θ_r;
Update$(\Theta_k, \Theta_k^g, \Theta_r, \Theta_r^g)$ **:**
compute new local Θ_k on attributes by Equation (1)
If HPFL:
    compute new local Θ_r by Equation (2)
If AHPFL:
    compute new local Θ_r following augmented mechanism by Equations (10) or (11)
return Θ_k and Θ_r

4.3 Server Design

HPFL extends the available information as much as possible with aggregation of local user models to the global user model in the server. In existing federated learning frameworks, the server follows the conventional aggregation strategy to fuse all local models. However, the client inconsistency in federated user modeling causes privacy heterogeneity and model heterogeneity, which makes simple strategy inappropriate [2]. In HPFL, we propose a flexible aggregation for inconsistent user models to separately aggregate the different components in GUM, i.e., the public components and private components. It is the differentiated component aggregation strategy as presented in Algorithm 2.

For the public component Θ_k in GUM, that is the user and item embeddings, which is with relative openness and tolerated to be directly fused, the server weighted aggregates the same attribute to obtain the global public components in Figure 2. In particular, in round t, after the local public components are delivered from clients, the server fuses each knowledge vector ($\mathbf{c}_{k,i}^t$) on attribute k from clients i in reference to both the number of iterations on attribute k ($I_{k,i}^t$) as well as the local validation accuracy ($Acc_{k,i}^t$). Finally, the global attribute knowledge, $\mathbf{c}_k^{t,g}$ in global public component Θ_k^g is denoted as:

$$\mathbf{c}_k^{t,g} = \frac{\sum_{i=1}^{C} \delta(Acc_{k,i}^t, p) \times (I_{k,i}^t \times Acc_{k,i}^t) \times \mathbf{c}_{k,i}^t}{\sum_{i=1}^{C} \delta(Acc_{k,i}^t, p) \times (I_{k,i}^t \times Acc_{k,i}^t)}. \tag{3}$$

In particular, $\delta(x, p)$ is an indicator function with a dynamic threshold p, where $\delta(x, p) = 1$, if $x > p$; otherwise, $\delta(x, p) = 0$. That is, the server randomly discards some knowledge vectors to dynamically select vectors in user models that are eligible to partake their knowledge.

For the private component Θ_r in GUM, it is strictly private. Since it is proprietary, unscrupulous alignment and fusion processes will reveal private preference information [31]. Therefore, the server receives the cluster centers in the clients which are regarded as the drafts of their representations from local user models. Subsequently, the server aggregates the information in representations via further clustering with all cluster centers as shown in Figure 2. The new cluster centers in the server, which are defined as the global private components Θ_r^g, represent the comprehensive representations of the bunches which involve similar users or items from different clients.

4.4 Local User Model Design

Clients use the local user model to properly model the characteristics of the users in various tasks. Indeed, more complex models (e.g., **Neural Cognitive Diagnosis (NeuralCD)** [88] and **Neural Collaborative Filtering (NCF)** [30]) are likely to be more suitable for the corresponding scenario,

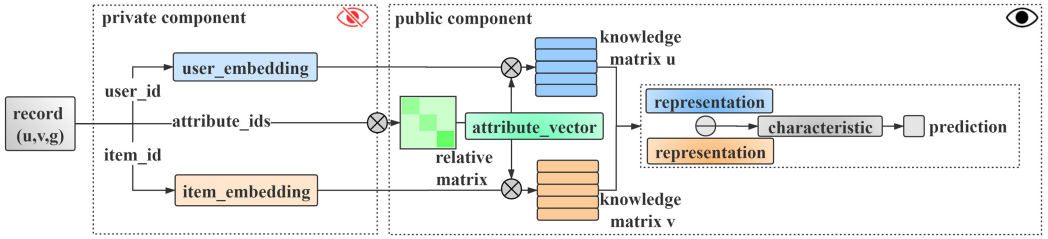


Fig. 3. General user model, GUM.

ALGORITHM 2: Differentiated Component Aggregation.

Input: The set of public components from clients, S_k; The set of drafts of private components from clients, S_r; The set of local validation results from clients, S_{acc};

Output: The aggregated global public components, Θ_k^g; The aggregated global private components, Θ_r^g;

initialize p.
Aggregation(S_k, S_r, S_{acc})**:**
If HPFL:
    compute Θ_k^g on attributes with S_k, S_{acc} and p by Equation (3)
If AHPFL:
    compute Θ_k^g on attributes following augmented mechanism by Equations (13) or (14)
$\Theta_r^g \leftarrow$ **Cluster**(S_r)
return Θ_k^g and Θ_r^g

while it has weak generalization for other tasks [9]. In addition, both of the above two models focus on low dimensions for user and item representations on attributes that lack the ability to model deep information [91]. We therefore propose a General User Model (GUM) with the deep representation as the Θ in our proposed HPFL framework, as shown in Figure 3.

GUM aims to model the potential characteristics of users and can be divided into public and private components, which are correspondingly designed for the hierarchical information. As shown in Figure 3, we use the triplet record (u, v, g) as the input of GUM. We then fetch the K-dimensional user embedding ($\mathbf{Emb_u}$) and item embedding ($\mathbf{Emb_v}$) via the user id and item id, respectively. The embeddings reflect the distributions of the inputs on the attributes. As mentioned before, the embeddings contain extremely private information, which can be used to infer local data distributions, so that they are defined as private components. In addition, we convert the attribute ids of the item v into the multi-hot vectors, which are used to correspondingly strengthen the information of certain attributes.

Next, we fuse the distributions on the attributes to the multidimensional vector representation. The representations of a user or item are treated as the summary of the characteristics. We design three matrixes as the mapping matrixes: i.e., the knowledge matrix u (KM_u), relation matrix (RM), and knowledge matrix v (KM_v) in Figure 3. KM_u and KM_v are $K \times N$ matrixes which represent the N-dimensional knowledge vectors of K attributes at user and item perspectives, respectively. Moreover, RM is a $K \times K$ matrix that indicates the knowledge relation among K attributes. With the multi-hot vector of the attribute information of the current item, we fetch the K-dimensional attribute vector ($\mathbf{Emb_c}$) from RM, which is on behalf of the relation of attributes. Finally, we compute the user representation ($\mathbf{R_u}$) and item representation ($\mathbf{R_v}$) as follows:

$$\mathbf{R_u} = \mathbf{Emb_u} \cdot \mathbf{Emb_c}^T \cdot KM_u. \tag{4}$$

$$\mathbf{R_v} = \mathbf{Emb_v} \cdot \mathbf{Emb_c}^T \cdot KM_v. \tag{5}$$

Obviously, the three mapping matrixes represent attributes with some multidimensional vectors, where the knowledge is relatively public and difficult to be used to detect private information. Therefore, we define these elements of GUM as public components.

Finally, we simply calculate the differences between the representations of user and item as the user reflex on an item as:

$$\mathbf{P_{uv}} = \mathbf{R_u} - \mathbf{R_v}. \tag{6}$$

The reflex $\mathbf{P_{uv}}$ measures the relation between user and item and can be used for different tasks. In our work, we use it to predict the objective in user modeling tasks. In particular, we multiply it with a simple $N \times 1$ **hidden mapping matrix (HM)** to obtain a continuous value, representing the final prediction as:

$$\mathbf{F_{uv}} = \mathbf{P_{uv}} \cdot HM. \tag{7}$$

In addition, we can represent the hidden characteristic, e.g., capability or preference of user out of the items based on the multidimensional representations as:

$$\mathbf{h_u} = \mathbf{Emb_u} \cdot (KM_u \cdot HM). \tag{8}$$

GUM can be the function structure for federated user modeling. Specially, GUM establishes both the private and public components for hierarchical information. The private components are superficial private representations of users and items, while the public components contain the multidimensional knowledge information on attributes. GUM can map the private representations to the multidimensional representation vectors, i.e., user representations and item representations, with mapping matrixes. Finally, a reflex between user and item is calculated for further prediction in user modeling tasks. Our proposed GUM is a flexible basic user model, so that it can be transformed to other common models. For instance, if we remove all the mapping matrix, the GUM will degenerate into a matrix factorization-like model. It is noting that we focus on serving federated learning for general user modeling tasks, GUM is selected due to the convenience, and the comparison between user models is not important.

4.5 HPFL Workflow

Algorithm 3 presents the iteration workflow of HPFL between clients and the server. For each global round, the client first processes local training with only local data on GUM and then delivers the local model to the server via different components. Hereafter, as shown in Algorithm 2, the server aggregates all local public components and private components by the differentiated component aggregation strategy and distributes the global components to all clients. Finally, as shown in Algorithm 1, the client receives the different aggregated components from the server and executes a fine-grained personalized update to improve its own user model. It is worth noting that, in our process, clients can independently initialize the local user models, since the federated learning on different information and structures of user models is achieved.

5 AUGMENTED HIERARCHICAL PERSONALIZED FEDERATED LEARNING

HPFL can effectively overcome the statistical, privacy, and model heterogeneities arising due to inconsistent client scenarios in federated learning. However, HPFL expects to absorb an extensive amount of information. Specially, clients indiscriminately fuse all global clustering centers to local representations as in Equation (2), and the server fuses the knowledge vectors from clients by a faint selection according to test indicators as in Equation (3). In fact, these naive methods may lead to harmness to the subsequent user models because of quality heterogeneity. In federated user modeling, local data distributions are not always of high quality as shown in Figure 1(b). For example, in the clients, there are always various reasons for low data quality; for example, the

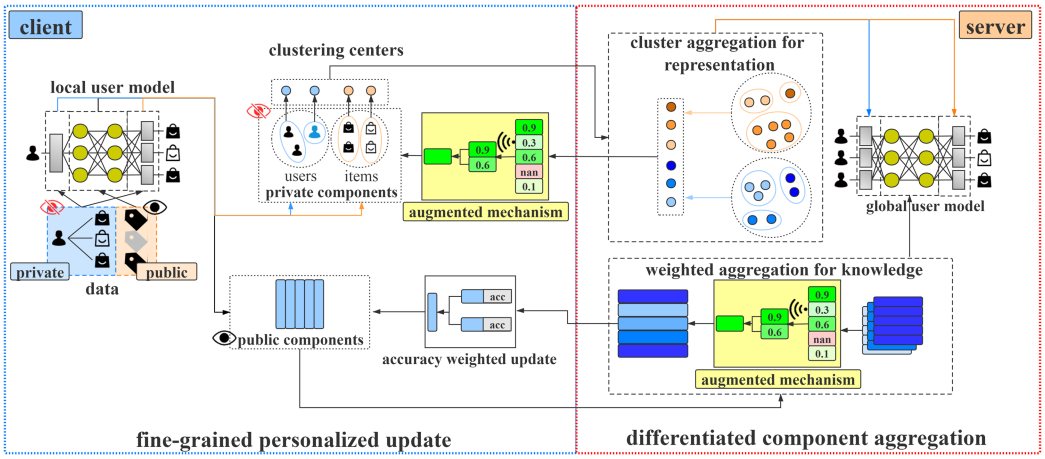


Fig. 4. Augmented Hierarchical Personalized Federated Learning framework, AHPFL.

ALGORITHM 3: The HPFL framework.

Server executes:
initialize Θ_k^0.
for each round $t = 1, 2, \ldots$ **do**
  initialize $S_k = \{\}, S_r = \{\}, S_{acc} = \{\}$
  for each client index $c \in C$ **in parallel do**
    $\Theta_{k,c}^{t+1}, Lc_c^{t+1}, \mathbf{Acc}_c^{t+1} \leftarrow$ **ClientUpdate**$(c, \Theta_k^{t,g}, \Theta_r^{t,g})$
    $S_k = S_k \cup \Theta_{k,c}^{t+1}, S_r = S_r \cup Lc_c^{t+1}, S_{acc} = S_{acc} \cup \mathbf{Acc}_c^{t+1}$
  $\Theta_k^{g,t+1}, \Theta_r^{g,t+1} \leftarrow$ **Aggregation**(S_k, S_r, S_{acc}) according to Algorithm 2

ClientUpdate$(c, \Theta_k^g, \Theta_r^g)$**:**
$\Theta_k, \Theta_r \leftarrow$ **Update**$(\Theta_k, \Theta_k^g, \Theta_r, \Theta_r^g)$ according to Algorithm 1
$\Theta_k, \Theta_r, \mathbf{Acc} \leftarrow$ **LocalTraining**(c, Θ_k, Θ_r)
$Lc \leftarrow$ **Cluster**(Θ_r)
return Θ_k, Lc and **Acc** to server

noise caused by an accidental touch, sparse or missing values due to disconnection of applications and redundant ones resulting from repeated operations, all of which result in user models of different quality. The naive methods mentioned for information fusion inevitably introduce such low-quality information. Naturally, reasonable augmented of high-quality information in the clients for local user models is beneficial to the user modeling tasks.

From this view, in this section, we further address the quality heterogeneity challenge in federated learning. We extend the current HPFL and propose an Augmented Hierarchical Personalized Federated Learning (AHPFL) framework by introducing the augmented mechanism, indicated by the yellow parts shown in Figure 4. In both aggregation in the server and update process in the client, we introduce the augmented mechanism to filter the original information, where some low-quality information has been incorporated. Specifically, we replace the distance-weighted update (Equation (2)) in the client and the weighted aggregation for knowledge (Equation (3)) in the server with some typical augmented mechanisms, such as SVD and AE as the augmented mechanism in our framework to implement the AHPFL-SVD and AHPFL-AE, respectively. In the following, we will introduce the details of the augmented client and server design.

5.1 Augmented Client Design

In clients, AHPFL follows the same upload phase. As for the update phase, different from the client design in HPFL, we add an augmented mechanism to update the high-quality private components as Algorithm 2. The original distance-weighted aggregation strategy is easily affected by low-quality information, such as outliers in clustering centers, resulting in damages for user models. Therefore, we design an augmented mechanism to filter out low-quality centers and integrate high-quality ones to *j-th* local embeddings in the client i $\mathbf{Emb}_{j,i}$ as shown in Figure 4, which is transformed from Equation (2) as:

$$\begin{aligned} \mathbf{Emb}^g &= AM(\{\Theta^g_{r,1}, \Theta^g_{r,2}, \ldots, \Theta^g_{r,N}\}), \\ \mathbf{Emb}_{j,i} &= \mathbf{Emb}_{j,i} \times \mathbf{Acc}_i + \mathbf{Emb}^g \times (1 - \mathbf{Acc}_i). \end{aligned} \tag{9}$$

Where $AM(\cdot)$ is the augmented mechanism used in our framework. $AM(\cdot)$ receives all global public components (N clustering centers) as the input and then outputs a high-quality vector representing purification of the embedding centers. There are many ways to achieve it, in this paper, we choose two of the typical methods, SVD and AE. In particular, we propose two implementations, namely SVD-based augmented and AE-based augmented mechanism.

For SVD-based $AM(\cdot)$, we first implement a matrix decomposition process on the matrix from all of N global clustering centers. Subsequently, we intercept the parameters with the largest M singular values and carry out weighted aggregation to compress all global clustering centers lossy. The SVD-based $AM(\cdot)$ is denoted as Equation (10). With the SVD augmented mechanism, the useless information, which is always with lower singular values, is filtered out, while the high-quality global components participate in fine-grained personalized update instead of the original information.

$$\begin{aligned} \{\Theta^g_{r,1}, \Theta^g_{r,2}, \ldots, \Theta^g_{r,N}\} &= \mathbf{U}\Sigma\mathbf{V}^{\mathbf{T}}, \\ \mathbf{Emb}^g &= \sum_{m=1}^{M} U_m \cdot \frac{\Sigma_m}{\sum_{i=1}^{M} \Sigma_i} \cdot V_m^T. \end{aligned} \tag{10}$$

For AE-based $AM(\cdot)$, we design a regularization autoencoder to directly mine the useful information vectors from the global clustering centers. All of N global clustering centers are encoded to a compressed vector, which contains valuable information. The vector is then decoded back to the unfolded vectors. The goal of the autoencoder is to reconstruct the original information as accurately as possible. The process is denoted as Equation (11). With the AE augmented mechanism, we can effectively mine the potential distribution information of components from the centers.

$$\begin{aligned} \mathbf{X} &= \{\Theta^g_{r,1}, \Theta^g_{r,2}, \ldots, \Theta^g_{r,N}\}, \\ \mathbf{Emb}^g &= Encoder(\mathbf{X}), \\ \mathbf{Y} &= Decoder(\mathbf{Emb}^g), \\ loss &= ||\mathbf{X} - \mathbf{Y}|| + \lambda|w|. \end{aligned} \tag{11}$$

5.2 Augmented Server Design

Different from the previous server design, we also add an augmented mechanism to replace the weighted aggregation for knowledge as Algorithm 1. The original heuristic method for public components does not intuitively filter out the low-quality information. Therefore, in the differentiated component aggregation, we add an augmented mechanism to filter out low-quality knowledge information from C clients and integrate the high-quality knowledge information as shown in

Figure 4, which is modified from Equation (3) as:

$$\mathbf{c}^g = AM(\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_C\}). \tag{12}$$

Similarly, $AM(\cdot)$ processes all the local knowledge vectors from clients by attributes and outputs high-quality vectors representing purification of the knowledge information. We further use two augmented mechanisms as mentioned earlier, that is SVD and AE-based augmented mechanisms.

For SVD-based $AM(\cdot)$, we first process a matrix decomposition process on the matrix of all knowledge vectors from clients on a certain attribute k. After that, we intercept the parameters with the largest M singular values and carry out the weighted aggregation to compress knowledge information. The SVD-based $AM(\cdot)$ is denoted as Equation (13). With the SVD augmented mechanism, the useless knowledge, which always has lower singular values, is filtered out, while the high-quality knowledge information is retained.

$$\begin{aligned} \{\mathbf{c}_{k,1}, \mathbf{c}_{k,2}, \ldots, \mathbf{c}_{k,C}\} &= \mathbf{U}\Sigma\mathbf{V}^{\mathbf{T}}, \\ \mathbf{c}_k^g &= \sum_{m=1}^{M} U_m \cdot \frac{\Sigma_m}{\sum_{i=1}^{M} \Sigma_i} \cdot V_m^T. \end{aligned} \tag{13}$$

For AE-based $AM(\cdot)$, we also use a regularization autoencoder to directly mine the useful information from the knowledge vectors on a certain attribute k. Similarly, the autoencoder encodes the knowledge vectors to a pure vector and expects to reconstruct the original information as accurately as possible. The AE-based $AM(\cdot)$ is denoted as Equation (14). With the AE augmented mechanism, we can mine the more valuable information on attributes as:

$$\begin{aligned} &\mathbf{X} = \{\mathbf{c}_{k,1}, \mathbf{c}_{k,2}, \ldots, \mathbf{c}_{k,C}\}, \\ &\mathbf{c}_k^g = Encoder(\mathbf{X}), \\ &\mathbf{Y} = Decoder(\mathbf{c}_k^g), \\ &loss = ||\mathbf{X} - \mathbf{Y}|| + \lambda|w|. \end{aligned} \tag{14}$$

5.3 Framework Summary

In summary, our proposed HPFL framework, which focuses on the federated user modeling scenarios with inconsistent clients, has the following advantages. First, HPFL achieves a personalized model update through fine-gained model fusion without neglecting statistical heterogeneity. Furthermore, through the aggregation process, we specifically aggregate the model components with different privacy intensity to make use of as much information as possible on the premise of protecting the privacy information, which conforms to privacy heterogeneity. At the same time, HPFL does not need to align the original representations in user models, so it is adaptive to the heterogeneous representation spaces among user models to suit model heterogeneity. In all, through the aggregation and updating of user models by components, HPFL enables expansion for extensive information from clients and inheritance for personalized information locally under the premise of privacy protection. Further, our proposed AHPFL adds augmented mechanisms based on the HPFL framework. First, AHPFL inherits the effective inconsistent client collaboration features. Then with the augmented high-quality information, AHPFL provides high-quality user models for inconsistent clients, which mitigates the damage from quality heterogeneity. Finally, our framework achieves good performances on multiple user modeling tasks, as covered in the next section.

6 EXPERIMENTS

In this section, we conduct extensive experiments to demonstrate the advancements achieved by our proposed HPFL and AHPFL frameworks. Specifically, we first introduce our experimental

Table 1. Statistics of the Datasets: ASSIST and MovieLens

Statistics	ASSIST	MovieLens-100K	MovieLens-1M
# of clients	59	10	10
# of records	327,058	96,538	1,000,209
# of users	3,477	925	6040
# of items	17,561	1,679	3,883
# of attributes	122	19	18
# attributes per item	1.20	1.72	1.65
# attributes per record	1.20	2.21	2.10

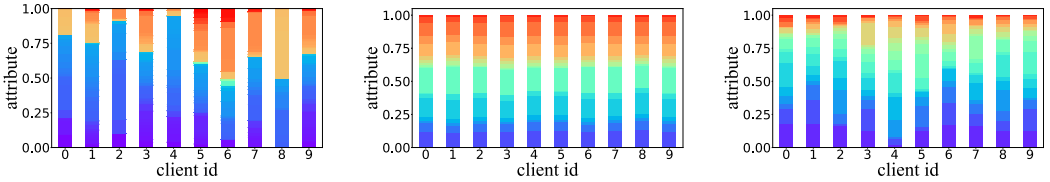
datasets related to the real tasks and experimental settings for the two frameworks in detail (Sections 6.1 and 6.2). We then carry out experiments from the following three aspects: (1) We demonstrate the quantitative performances on prediction tasks for two user modeling scenarios (Section 6.3); (2) We illustrate the modeling rationality at the parameter level by components (Section 6.4); (3) We compare the augmented extent of AHPFL with other methods (Section 6.5).

6.1 Experimental Datasets

In experiments, three real-world public datasets for two typical user modeling tasks, which are referred to in Section 3.2 are employed to verify our proposed HPFL and AHPFL frameworks, namely ASSIST and MovieLens. **ASSIST** (short for **Assistments**) is a common educational dataset for user capability modeling on students. In ASSIST (2009-2010 "Non-skill builder"[1]), mathematics learning logs were collected during the online learning activities on the online tutoring program. The others are MovieLens-100K and MovieLens-1M,[2] which are commonly used for user preference modeling in recommendation tasks. In MovieLens-100K and MovieLens-1M, there are rating records for movies obtained through the MovieLens website.

We conduct experiments in the form of the cross-silo federated learning. To simulate the real isolated scenario, we try to divide the two datasets as naturally as possible. To be specific, in ASSIST, we divide the data into clients by teacher ids, which causes isolation between classes. We further filter out some clients which are not sufficient to support local user modeling training, such as those for which the average number of records or students is fewer than five in the client. Finally, we generate a pruned dataset from ASSIST, where there are over 3,477 students and over 17K questions which belong to 122 concepts (i.e., "Equivalent Fractions" and "Pythagorean Theorem") and are responded to on a two-point scale. During the learning process, there are over 300K interaction records between the above students and questions. All of the data is naturally dispersed in 59 clients. Similarly, we divide the data in MovieLens-100K and MovieLens-1M according to users' location via national area in zipcode (e.g., 1xxxx–Delaware, New York, Pennsylvania), which causes a geographic isolation scenario in the data. To support the training, we delete clients with fewer than five customers as before. We subsequently obtain a divided dataset from MovieLens. There are over 96K records, where 925 customers rate on 1,676 products (i.e., movies) belonging to 19 categories (i.e., "Comedy" and "Romance") on a five-point scale in Movielens-100K, while there are over 1M records, where 6,040 customers rate on 3,883 products belonging to 18 categories ("unknown" type is removed) on a five-point scale in Movielens-1M. All of the data is naturally dispersed in 10 clients. Table 1 summarizes more basic statistics. Specifically, as defined in Section 3.1, there is both public and private information in the data. Obviously, the attributes,

[1] https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/non_skill-builder-data-2009-2010.
[2] https://grouplens.org/datasets/movielens.



(a) Distribution in ASSIST. (b) Distribution in MovieLens-100K. (c) Distribution in MovieLens-1M.

Fig. 5. Distribution of attributes by clients of three datasets: ASSIST (left), MovieLens-100K, and MovieLens-1M (right).

such as the concepts of questions and categories of products are often public and shared between clients. Correspondingly, the interaction records of users and items, like answer logs to questions of students and users' rating logs on movies are private, since they contain some sensitive information, which represent the preferences of users used in user modeling. It is worth noting that, in our isolated scenario, each client processes user modeling only with its own data during the training process. We aim to update local GUMs with isolated and inconsistent datasets in such a way that we provide high-quality user models for clients. Consequently, we solve the federated user modeling problem for inconsistent local clients.

In isolated scenarios, the distributions of data in the clients are the important influence factor for federated learning. We then analyze the data to compare the data distributions on the attributes of different clients. Figure 5 shows the distributions of all the attributes of both datasets. For illustrative purposes, we choose 10 clients with the largest data volumes. In the figure, different colors represent different attributes in the client and cover areas representing the frequency of attribute occurrence; the larger the area, the more frequent the occurrences of a certain attribute. Taking the line of client 8 in Figure 5(a) as an example, there is an orange area in the line that is larger than the others; this reflects that distributions of the attributes represented by orange color, appear more frequently than those of any other clients. The inconsistent color distributions apparently show that the frequency of occurrence of attributes is inconsistent, which results in Non-IID data across clients in ASSIST. On the contrary, the distributions of attributes in MovieLens are almost consistent, which demonstrates that the data in MovieLens-100K is closer to being IID. Therefore, ASSIST is a natural Non-IID dataset, whereas MovieLens-100K is used as an approximate IID dataset in our experiments. While the distributions in MovieLens-1M are compromised between IID and Non-IID. Although the distribution of attributes in Movielens-100K is IID, there are still inconsistencies in the user space and item space that bring inconsistency to the structures of user models. Thus, we conduct experiments on both natural IID and Non-IID datasets to compare the user modeling effects of HPFL and AHPFL frameworks under different distribution scenarios. A stable framework is expected to fit not only simple but also complex scenarios.

6.2 Experimental Settings

In this subsection, we clarify the technical details to set up our HPFL and AHPFL frameworks. Then, we introduce the comparison baselines and evaluation metrics.

6.2.1 Data Partition. The user records in both datasets, i.e., scores on questions in ASSIST and rating on movies in MovieLens, are randomly partitioned into 80%/20% training/testing subsets. It is worth noting that, during the training process, the local data is kept in each client and is not allowed to be shared across clients in the experiments for federated user modeling methods, while we make no guarantees on the data privacy in the client across users.

6.2.2 Model Implementations. To better illustrate the implementation of our frameworks, we will introduce the framework settings as well as training settings in detail.

Framework settings. We specify the framework settings in HPFL and AHPFL from local model settings and framework settings. As for the local model, we proposed a general neural user model, namely GUM, with N dimensions of knowledge vector (N) in KM_u and KM_v equal to 5. As for framework settings, in both frameworks, we process the K-means [57] as the clustering methods to achieve desensitization of private components. Furthermore, we set the adaptive number of initial centers in clustering, that is, 1/10 of the number of individuals participating in the clustering. In addition, in AHPFL, we retain the largest 90% of information to filter the low-quality information and we adopt the regularized autoencoder with one hidden layer as the two augmented mechanisms.

Training settings. We initialize all parameters in both the HPFL and AHPFL with xavier initialization following [25] with the uniform distribution in the range $(-\sqrt{(6/ni + no)}, \sqrt{(6/ni + no)})$, where *ni* and *no* are the dimensions of the input and output, respectively. We then train the GUM models with mini-batches of 32 and a learning rate of 0.001 under the Adam optimizer. As for training the autoencoders, an Adam optimizer with learning rate of 0.005 and L1 regularization weight of 0.001 is employed. To facilitate further research into HPFL and AHPFL, we have published our code.[3]

6.2.3 Baselines. First, we compare the effectiveness of our proposed GUM with two typical user modeling methods, i.e., **NeuralCD** and **NCF**, in the centralized training process on all collected data. **NeuralCD** and **NCF** are two common user modeling models, which focus on cognitive diagnosis and collaborative filtering recommendation tasks, respectively.

- **NeuralCD** [88] is a state-of-the-art cognitive diagnosis model, which models the complex cognitive relationships of shallow representations of both students and questions.
- **NCF** [30] is also a state-of-the-art collaborative filtering model based on deep neural networks, which models the shallow features of both users and items.

Then, we conduct experiments in data isolated scenarios. To refine the contrast, we train the user models locally without any collaboration with other clients, which is denoted as **Distributed**. Furthermore, we repeat some representative universal model fusion methods of federated learning, which achieve the collaboration among clients to extend the available information. All federated methods are built on our proposed GUM for user modeling tasks. These methods are primarily used for processing in inconsistent scenarios by simply using the general settings.

- **FedSGD** [60] is a standard federated learning method based on stochastic gradient descent, where the server takes a simple weighted average of all models to obtain a united global model and clients perform one epoch of gradient descent per training process.
- **FedAvg** [60] also aggregates models to a united global model. However, FedAvg processes more computation steps in gradient descent to accelerate the training and convergence.
- **Fednoise** is an extension method based on a traditional federated process. It follows local differential privacy [14, 20, 92] which adds some random disturbances, like Laplace noise to local models before transmission.
- **FedProx** [47] adds a reference loss in local training for each client, that is, the distances between the local model and the global model, so that it constrains the local personalized optimization process not to drift excessively.

[3] https://github.com/bigdata-ustc/hierarchical-personalized-federated-learning.

- **FedAtt** [38] incorporates soft-attention for aggregation. The server considers the importance of the models and aggregates local models in layers with the distances between the global model and local models. It then weighted aggregates local models to obtain the global model.
- **FedAmp** [34] employs federated attentive message passing to facilitate similar clients, which does not use a single global model on the cloud server to conduct collaborations.

In addition to verifying the different federated processing, we conduct experiments on the user modeling tasks to compare our proposed **HPFL** [97] and **AHPFL** based frameworks, i.e., **AHPFL-SVD** and **AHPFL-AE**, which implement the augmented mechanism following SVD (singular value decomposition) and AE (autoencoder), respectively. In the following experiments, all above-mentioned baselines and our proposed **HPFL** and **AHPFL** are implemented by PyTorch. For fairness, all the methods are trained on the same Linux server with four 2.20GHz Intel Xeon E5-2620 CPUs, two NVIDIA Tesla K80 GPUs ,and 256GB memory to achieve the best performance for comparison.

6.2.4 Evaluation Metrics. A qualifying user model should achieve good results in our two scenarios from regression and classification perspectives. In this paper, we observe the accuracy performances of our proposed methods in user modeling tasks by employing widely used metrics [15, 22, 36].

In the experiments on ASSIST, which is the dataset for user capability modeling in education to predict the two-point student performances, we use the ROC Curve (AUC) and Prediction Accuracy (ACC) to measure the prediction from a classification perspective in the range of $[0, 1]$, the larger the values are, the better the results. Correspondingly, in MovieLens, which is the dataset for continuous 5-point rating prediction, the ACC is not suitable for evaluating the prediction performances. For example, we can round a prediction score 3.5 to 3 or 4. However, it will make great numerical differences in ACC, which leads to errors in results. Therefore, we use the widely used Coefficient of Determination (R^2) and **Mean Absolute Error (MAE)** to measure the proximity between prediction and ground truth from a regression perspective. The ranges of both metrics are $[0, 1]$, the higher the values are, the better the R^2; while the lower the values are, the better MAE. Besides, for both scenarios, we use the **Root Mean Square Error (RMSE)** to quantify the distance between predicted and the actual ones with the range of $[0, 1]$, the lower the values, the better the results.

In the practical user modeling tasks, we should be successful in predicting the extent of the preference of a user, rather than simply reporting whether he is interested in the item or not [65]. Therefore, we adopt some commonly used ranking measurement indicators to count whether or not the predicted ranking of the practically more preferred item is higher. In this case, **Degree of Agreement (DOA)** [35] and **Normalized Discounted Cumulative Gain (NDCG)** [29] are favored to reflect the ranking effectiveness of the models.

6.3 Quantitative Performances

6.3.1 Accuracy Performances. To evaluate the quantitative performances of our frameworks, we first evaluate the accuracy performances in isolated user modeling scenarios and conduct the prediction tasks as discussed before. We implement the tasks as student performance prediction and user rating prediction and use $\mathbf{F_{uv}}$ to predict the interactions for two typical user modeling tasks, i.e., cognitive diagnosis and collaborative filtering recommendation, respectively. Since we focused on serve model fusion of federated learning for general user modeling task, the comparison between user models is not important and we select some typical and universal federated user modeling frameworks for a general user model fusion and update. We repeat the experiments five

times and summarize the average of the results. Table 2 lists the overall results on both datasets with the evaluation metrics mentioned. It should be noted that for student performance prediction of a two-point scale, we usually focus on AUC and ACC. While for rating prediction, especially for non-two-point scale rates, R^2 and MAE are the more reasonable indicators.

Some key observations are as follows: (1) Our proposed GUM model performs better than NCF and NeuralCD on both central and distributed datasets. It shows our general user model that is capable of deep representation for users and items is general and appropriate for user modeling tasks. (2) Most federated methods perform better than distributed training processes. It shows federated learning settings can harness more information and promote collaboration among isolated clients, which usually results in better user models. Obviously, our proposed HPFL and AHPFL-based methods achieve better performances than any other methods on both datasets. This means that our methods can more effectively accommodate user modeling tasks. (3) Among our proposed frameworks, we find that AHPFL-based methods (AHPFL-SVD, AHPFL-AE) outperform the HPFL, indicating the effectiveness of the augmented mechanism in promoting a high-quality user model for inconsistent clients. (4) Specially, there is a little performance lift across ASSIST and MovieLens (MovieLens-100K, MovieLens-1M). It may be due to the two reasons as follows: (1) As we mentioned in Section 6.1, the fundamental difference between the two datasets is the data distribution. ASSIST is a natural Non-IID dataset, whereas the MovieLens datasets are used as an approximate IID dataset in our experiments. For the IID dataset, our design for public information probably worked better, which facilitates collaboration among clients. (2) In addition, the numbers of users and items in ASSIST datasets are very different among clients. Perhaps a finer design for the number of cluster centers will help improve the effect on the data.

Specially, we do the Student t-test to clarify whether the AHPFL-based methods perform better than HFPL. We choose the important and concerned indicators of the different tasks respectively, that is, AUC on Assist and R^2 on Movielens. The p-value results are shown in Table 3. Most results are smaller than the significance level 0.05, thus we reject the hypothesis that the performances between AHPFL and HPFL are approximate. It suggests that AHPFL-based methods are more effective than HPFL in most tasks.

6.3.2 Individual Improvement. In our federated user modeling task, while overall performances are important, our primary concern is with the percentage of distributed clients that will benefit. Thus, we compare the performances of clients in the distributed training method with federated methods and accordingly count the "proportion" of clients with improvements according to the metrics. In this way, we can determine which method will improve more participants in real scenarios, which will be beneficial to the application of the method.

Table 4 reports the proportion of clients with individual improvements. From the table, we can draw the following conclusions: (1) Federated learning methods generally bring effectiveness to clients on both datasets. However, the differential privacy settings cause a loss of performances as a result of Fednoise. (2) Our methods significantly increase the proportion of clients with individual improvements. It illustrates that the aggregation and update by components can expand information from other participants and are helpful for training local user models. In particular, our AHPFL-based methods, i.e., AHPFL-SVD and AHPFL-AE have a higher proportion, indicating that the augmented mechanisms augment high-quality information from clients, which ultimately leads to more high-quality user models.

6.3.3 Ablation Study. To verify the effectiveness of different components in our proposed frameworks, i.e., HPFL and AHPFL, we conduct several ablation experiments. Specifically, for all the implementations, i.e., HPFL, AHPFL-SVD, and AHPFL-AE, we compare the complete framework with two simplified versions: ***-K** and ***-R**, which process only on public or private

Table 2. Accuracy Performances of User Modeling Tasks for Metrics on Datasets

(a) Accuracy performances on ASSIST

Methods	ASSIST		
	ACC	AUC	RMSE
NeuralCD	0.724 ± .0017	0.739 ± .0038	0.434 ± .0016
GUM	**0.740 ± .0017**	**0.775 ± .0003**	**0.419 ± .0007**
Distributed-NeuralCD	0.689 ± .0025	0.697 ± .0015	0.451 ± .0010
Distributed-GUM	0.701 ± .0038	0.719 ± .0009	0.442 ± .0008
FedSGD	0.698 ± .0032	0.711 ± .0021	0.449 ± .0022
FedAvg	0.705 ± .0012	0.721 ± .0028	0.445 ± .0012
Fednoise	0.707 ± .0039	0.720 ± .0032	0.443 ± .0021
FedProx	0.706 ± .0014	0.722 ± .0018	0.443 ± .0015
FedAtt	0.712 ± .0019	0.726 ± .0007	0.439 ± .0005
FedAmp	0.720 ± .0017	0.732 ± .0006	0.435 ± .0007
HPFL	0.725 ± .0013	0.740 ± .0020	0.433 ± .0011
AHPFL-SVD	**0.727 ± .0007**	**0.743 ± .0002**	**0.431 ± .0002**
AHPFL-AE	0.725 ± .0012	0.742 ± .0011	**0.431 ± .0004**

(b) Accuracy performances on MovieLens-100K

Methods	MovieLens-100K		
	R^2	MAE	RMSE
NCF	0.288 ± .0026	0.761 ± .0021	0.955 ± .0017
GUM	**0.310 ± .0035**	**0.742 ± .0010**	**0.940 ± .0023**
Distributed-NCF	0.138 ± .0030	0.822 ± .0010	1.038 ± .0020
Distributed-GUM	0.167 ± .0094	0.804 ± .0045	1.018 ± .0086
FedSGD	0.114 ± .0072	0.856 ± .0044	1.052 ± .0043
FedAvg	0.210 ± .0023	0.799 ± .0020	0.994 ± .0017
Fednoise	0.194 ± .0017	0.795 ± .0037	1.008 ± .0045
FedProx	0.223 ± .0026	0.792 ± .0028	0.989 ± .0007
FedAtt	0.219 ± .0019	0.791 ± .0023	0.989 ± .0010
FedAmp	0.227 ± .0018	0.780 ± .0020	0.983 ± .0008
HPFL	0.234 ± .0012	0.776 ± .0003	0.981 ± .0008
AHPFL-SVD	**0.239 ± .0014**	0.775 ± .0009	**0.977 ± .0009**
AHPFL-AE	0.236 ± .0006	**0.774 ± .0007**	0.978 ± .0003

(c) Accuracy performances on MovieLens-1M

Methods	MovieLens-1M		
	R^2	MAE	RMSE
NCF	0.266 ± .0006	0.767 ± .0008	0.957 ± .0004
GUM	**0.282 ± .0010**	**0.752 ± .0005**	**0.947 ± .0007**
Distributed-NCF	0.231 ± .0025	0.779 ± .0005	0.992 ± .0013
Distributed-GUM	0.242 ± .0005	0.768 ± .0002	0.985 ± .0003
FedSGD	0.176 ± .0047	0.834 ± .0055	1.028 ± .0058
FedAvg	0.275 ± .0025	0.774 ± .0014	0.965 ± .0011
Fednoise	0.254 ± .0051	0.779 ± .0095	0.977 ± .0062
FedProx	0.273 ± .0015	0.774 ± .0014	0.965 ± .0011
FedAtt	0.275 ± .0025	0.772 ± .0024	0.963 ± .0017
FedAmp	0.299 ± .0020	0.745 ± .0019	0.947 ± .0009
HPFL	0.304 ± .0004	0.742 ± .0003	0.944 ± .0003
AHPFL-SVD	0.307 ± .0007	0.740 ± .0003	0.942 ± .0005
AHPFL-AE	**0.309 ± .0009**	**0.739 ± .0003**	**0.940 ± .0006**

Table 3. P-value between AHPFL and HPFL on Datasets

Methods	ASSIST	MovieLens-100K	MovieLens-1M
AHPFL-SVD/HPFL	0.040	0.002	0.00017
AHPFL-AE/HPFL	0.061	0.022	0.00022

Table 4. Proportion of Clients with Individual Improvement on Both Datasets

(a) Proportion on ASSIST

	FedAvg	Fednoise	FedProx	FedAtt	FedAmp	HPFL	AHPFL-SVD	AHPFL-AE
Rate	0.559	0.457	0.559	0.372	0.576	0.847	**0.864**	0.847

(b) Proportion on MovieLens-100K

	FedAvg	Fednoise	FedProx	FedAtt	FedAmp	HPFL	AHPFL-SVD	AHPFL-AE
Rate	0.40	0.20	0.50	0.50	0.50	0.70	**1.00**	**1.00**

(c) Proportion on MovieLens-1M

	FedAvg	Fednoise	FedProx	FedAtt	FedAmp	HPFL	AHPFL-SVD	AHPFL-AE
Rate	0.80	0.60	0.80	0.80	0.9	**1.00**	**1.00**	**1.00**

Table 5. Results of Ablation Experiment

	ASSIST			MovieLens-100K			MovieLens-1M		
Methods	ACC	AUC	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE
HPFL	**0.725**	**0.740**	**0.433**	**0.234**	**0.776**	**0.981**	**0.304**	**0.742**	**0.944**
HPFL-K	0.715	0.730	0.437	0.221	0.792	0.987	0.299	0.754	0.946
HPFL-R	0.723	0.738	0.433	0.229	0.798	0.991	0.296	0.745	0.949
AHPFL-SVD	**0.727**	**0.743**	**0.431**	**0.239**	**0.775**	**0.977**	**0.307**	**0.740**	**0.942**
AHPFL-SVD-K	0.721	0.733	0.438	0.231	0.777	0.981	0.304	0.743	0.943
AHPFL-SVD-R	0.723	0.740	0.432	0.226	0.780	0.984	0.303	0.742	0.944
AHPFL-AE	**0.725**	**0.742**	**0.431**	**0.236**	**0.774**	**0.978**	**0.309**	**0.739**	**0.940**
AHPFL-AE-K	0.719	0.731	0.437	0.231	0.775	0.981	0.306	0.741	0.941
AHPFL-AE-R	0.720	0.738	0.433	0.228	0.780	0.983	0.305	0.740	0.943

components, respectively. Table 5 reports the results of each case. From the results, we can conclude the following: (1) HPFL and AHPFL perform the best performances on both tasks. While the performances of the simplified methods, *-K and *-R are poorer than complete HPFL and AHPFL frameworks, because both of these methods lack some information of model components, i.e., lack of private component and public component, respectively. (2) Relatively, the *-R methods perform a better result than the *-K methods in the Non-IID datasets, ASSIST; while in the IID datasets, overall, *-K methods are better. It shows that for Non-IID scenarios, the private information is more important, since it represents the personalized characteristics of the clients better; while in IID scenarios, public information is suggested to be focused for accurately user modeling, since the private information may be consistent.

6.3.4 Ranking Effectiveness. In practice, rather than simply classifying the reflex on the item, we are more concerned about whether we can more accurately model the partial order of hidden characteristics. Specifically, in the recommendation domain the characteristics refer to the user's preference, while in the education domain the characteristics refer to students' knowledge

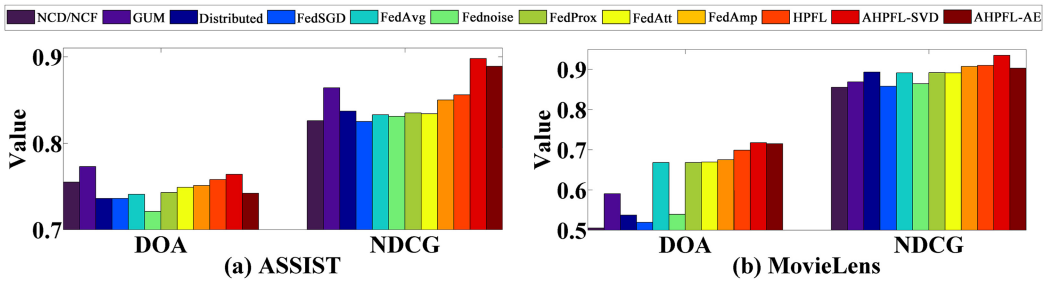


Fig. 6. Ranking effectiveness for metrics on both datasets.: ASSIST (left), MovieLens-100K (right).

proficiency. In recommendation systems, applications expect to rank the most preferred items to users. In the field of education, the determination of partial order of ability among students is concerned with comparability and test equating problem [45]. As some works on intelligent education [88] suggest, compared to the scores the knowledge proficiency of students is the output to be concerned of the cognitive diagnosis model. Similar requirements apply to models in recommendation system. Thus, we evaluate the ranking effectiveness of our frameworks on both datasets with some commonly used indicators. We adopt two indicators to measure the ranking effectiveness from two aspects: one is the extra ranking effectiveness, for which we use the Degree of Agreement (DOA) to measure the consistency of preferences and predictions in the group. That is, whether one prefers the same item over another user as the user model reflects or one outperforms on the same item than another. Specifically, a DOA result on a specific attribute k is defined as:

$$DOA(k) = \sum_{a=1}^{|U_{c_1}|} \sum_{b=1}^{|U_{c_2}|} I_{abk} \frac{\delta(h_{ak}, h_{bk}) \cap \delta(\bar{g}_{ak}, \bar{g}_{bk})}{\delta(h_{ak}, h_{bk})}. \tag{15}$$

Here, U_{c_1} and U_{c_2} denote the two different users in two different clients c_1 and c_2, while h_{ak} indicates the hidden characteristic, e.g., capability or preference of user a on attribute k obtained by our user models as Equation (8), and $\bar{g}_{ak}$ is the average response of user a on attribute k. $\delta(x, y)$ is an indicator function, where $\delta(x, y) = 1$, if $x > y$; otherwise, $\delta(x, y) = 0$. I_{abk} is another indicator function, where $I_{abk} = 1$ if both user a and user b have previously interacted on the attribute k. Furthermore, we average the DOA(k) of all attributes as DOA to measure the extra ranking effectiveness, which is denoted as $DOA = \sum_{k=1}^{K} DOA(k)/K, DOA \in [0.0, 1.0]$, the larger the DOA, the better the extra ranking.

The other is the inter ranking with Normalized Discounted Cumulative Gain (NDCG) for measuring the consistency of real preferences and predictions for users. That is, whether one prefers an item to another item as the user model reflects. First, we define the DCG of a specific user u, which is formulated as follows:

$$DCG(u) = \sum_{k=1}^{K} \frac{h_{uk}}{\log_2 k + 1}. \tag{16}$$

Here, K denotes the total attributes and the K attributes are ordered by $\bar{g}_{uk}$ as the recall order. Then we define $NDCG(u) = DCG(u)/IDCG(u)$, where the $IDCG(u)$ is the ideal $DCG(u)$, that is, we apply the h_{uk} descending sorted to $DCG(u)$. Furthermore, we average the NDCG(u) of all users as NDCG to measure the inter ranking effectiveness as $NDCG = \sum_{u=1}^{|U|} NDCG(u)/|U|, NDCG \in [0.0, 1.0]$. A larger NDCG indicates a better inter ranking performance.

Figure 6 reports the ranking effectiveness on DOA and NDCG. We can conclude the following from the results: (1) GUM performs better than other centralized methods, meaning that our high-dimensional user model adds more comparability for both inter and extra ranking. (2) HPFL and

Table 6. Similarity of Different Methods on Both Datasets

Methods	ASSIST	MovieLens
Distributed	3.259	3.309
FedAvg	0.191	0.773
HPFL	0.472	0.155
AHPFL-SVD	4.032	1.583
AHPFL-AE	14.143	2.274

AHPFL-based methods (AHPFL-SVD, AHPFL-AE) perform outstanding results, while AHPFL-SVD performs better than AHPFL-AE, since the matrix decomposition method retains more original information than the autoencoder method when compressing the user modeling information from clients. (3) Compared with standard federated learning methods, the distributed training method performs a comparable result in NDCG, while it performs inferior results in DOA. It demonstrates that standard federated methods bring a coordination among clients so that it is beneficial to extra ranking but causes weakness in inter ranking to some extent for user modeling.

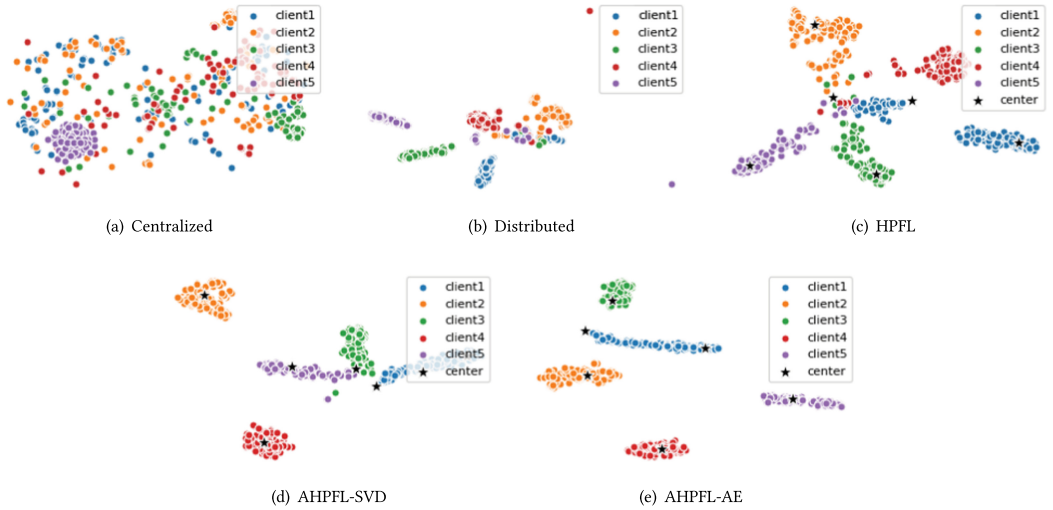
6.4 Modeling Rationality

In addition to quantitatively comparing the effects of our frameworks on some application tasks, we further analyze the rationality of user models at the parameter level. We expect HPFL and AHPFL to facilitate the creation of more rational user models. As mentioned earlier in Section 4.4, there are two components in local GUMs, i.e., public component and private component for hierarchical information. In our framework, we process the local user models by components to obtain appropriate user models. In order to compare the effects of hierarchical information, we deeply analyze the user models from different frameworks by different components. In particular, we conduct similarity analysis of public components and personalization analysis of private components to observe the similarities and differences between clients in federated learning.

6.4.1 Similarity Analysis of Public Components. We expect public components to represent information collaboration between clients. Therefore, we calculate the similarity of public components from clients of different methods. Specifically, we compare the multi-client methods, such as distributed training process, the standard and clear FedAvg and our methods, then we calculate the similarity of the corresponding public component between different clients. We define total similarity as:

$$Simi = \frac{\sum_{k=1}^{K} \sum_{i=1}^{C} \sum_{j=1}^{C} dis(\mathbf{c}_{k,i}, \mathbf{c}_{k,j})}{K}, \tag{17}$$

where the $dis(x, y)$ is the cosine distance function in the range of [0.0, 2.0] and it is applied to the pair-wise knowledge vectors in user models from different clients. The lower the value of Simi, the higher the similarity. To facilitate better comparison, we choose 10 clients with the largest data volumes on both datasets. Table 6 reports the results of similarity in models from both datasets. According to the results, we obtain the following conclusions: (1) On both datasets, public components across clients in the distributed training method differ more significantly, since there is no federated process that clients communicate on public components. (2) The similarity on ASSIST, which is Non-IID is much higher than those on MovieLens which is more IID, it shows that models trained on IID data are more likely to learn a similar distribution for parameters that represent the global distribution to some extent to obtain a better local user model, GUM. While models on Non-IID data should have some personalization, because in case the consistent user models can lead to errors. Just as FedAvg has lower similarity, while it performs worse on



(a) Centralized (b) Distributed (c) HPFL

(d) AHPFL-SVD (e) AHPFL-AE

Fig. 7. The user characteristics of different methods reduced dimension by t-SNE on five clients in ASSIST. Cluster centers are marked with "center".

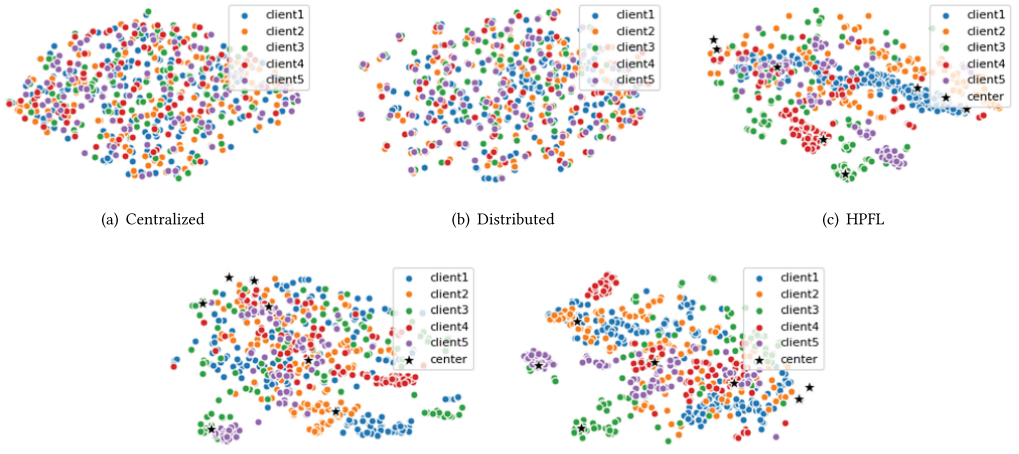
ASSIST as shown in Table 2. (3) It is worth noting that on both datasets, AHPFL-AE makes the clients more diverse than AHPHL-SVD or HPFL, since the augmented mechanism, especially a regularized autoencoder, may augment the special valuable information, which leads to loss of co-occurring low-quality information.

6.4.2 Personalization Analysis of Private Components. We expect to validate the ability of private components to capture personalized information. Specifically, we choose the conventional training methods, i.e., centralized and distributed training process for user modeling with our methods to analyze the rationality of embeddings in user models on clustering impressions. Specifically, we visualize the user characteristics from Equation (8) after reducing their dimension by t-SNE [56], which is commonly used for the visualization of high-dimensional data, to reduce the dimensionality of each exercise vector to a 2D data. For better illustration, we choose five clients with the most data. In particular, we annotate the cluster centers of users on figures of HPFL and AHPFL. Finally, we label users from each client with different colors.

Figures 7 and 8 illustrate the user characteristics on both datasets. Through analyzing the visual representation of the figures, we come to the following conclusions: (1) On both datasets, private components in user models are not distinguishable in the centralized training process, while a distributed training process may enhance the gathering effect. (2) We observe that on both datasets, HPFL has a stronger ability to promote collaboration among clients and discover clusters than AHPFL methods, especially the cross-client clusters. It reveals that the augmented mechanism may focus on augmenting the high-quality personalized information. (3) In MovieLens, the aggregation effect, even in the distributed training method, is not noticeable, since the IID distributions weaken the personality of the clients. Under such severe cases, our HPFL and AHPFL methods that process the private components, still capture the personalized information, which shows that on both types of distributions, our methods have advantages when it comes to mining the idiosyncrasies of clients from user characteristics in user modeling.

6.5 Augmented Extent

Furthermore, we observe the augmented extent of AHPFL-based methods. AHPFL is able to aggregate information from inconsistent clients lossily and augment the high-quality information.



(a) Centralized (b) Distributed (c) HPFL

(d) AHPFL-SVD (e) AHPFL-AE

Fig. 8. The user characteristics of different methods reduced dimension by t-SNE on five clients in Movielens. Cluster centers are marked with "center".

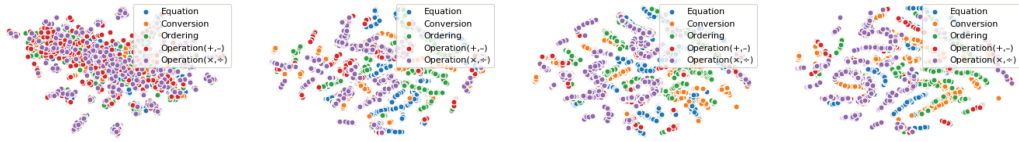
Table 7. Augmented Performances between Centralized and Federated Methods for OAR on Both Datasets

Methods	ASSIST	MovieLens
FedAvg	9.361	2.047
HPFL	7.495	2.532
AHPFL-SVD	9.916	3.603
AHPFL-AE	10.066	4.087

6.5.1 Augmented Performances on Public Components. In order to compare the effects of the two augmented mechanisms-based frameworks, we contrast global public components from centralized and federated methods to observe the quality of information aggregated and come up with a metric, **OAR (original information to augmented information ratio)**, which is similar to **SNR (signal-to-noise ratio)** [42]. Generally, a substantial amount of comprehensive data can guarantee the stability of neural network architectures during training, resulting in high-quality models [40, 66]. Therefore, we calculate the OAR of global public components in the server under different methods with the components in the centralized method, which are considered to be of higher quality. Specifically, we analyze the federated methods without augmented mechanism, that is FedAvg and HPFL, with our AHPFL, and calculate the OAR as:

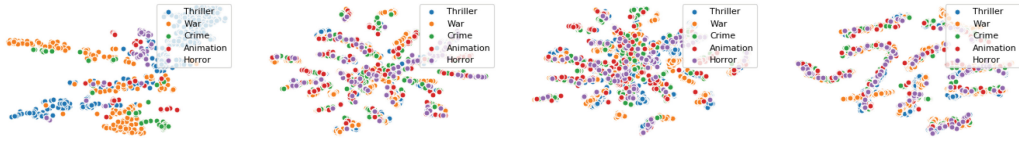
$$OAR = \frac{\sum_{i=1}^{K}\sum_{j=1}^{N}\Theta_{k,i,j}^{2}}{\sum_{i=1}^{K}\sum_{j=1}^{N}(\Theta_{k,i,j}-\Theta_{k,i,j}^{g})^{2}}, \tag{18}$$

where Θ_k denotes the public components of the centralized training method and the $\Theta_{k,i,j}^{g}$ is the global components of a certain federated learning method. The higher the OAR, the closer the components, which means information with higher quality and lower impurity. Table 7 reports the results of augmented performances on OAR from both datasets. According to the results, we obtain the following conclusions: (1) On both datasets, there is a higher OAR between centralized training method with AHPFL-based methods, i.e., AHPFL-SVD and AHPFL-AE. It shows that



(a) Centralized (b) HPFL (c) AHPFL-SVD (d) AHPFL-AE

Fig. 9. The item characteristics of different methods with reduced dimension by t-SNE on five clients in ASSIST.



(a) Centralized (b) HPFL (c) AHPFL-SVD (d) AHPFL-AE

Fig. 10. The item characteristics of different methods reduced dimension by t-SNE on five clients in Movielens.

AHPFL-based methods can better purify the original information from inconsistent clients, so as to obtain the more reliable global components. (2) Further, AHPFL-AE has stronger augmented ability than AHPFL-SVD, since the autoencoder is more capable of mining the effective potential distributions of components than the naive matrix factorization method.

6.5.2 Augmented Analysis on Private Components. Further, we expect to validate the augmented extent on private components to capture high-quality information. Similarly, we choose the above training methods and visualize the item characteristics after reducing their dimensions by t-SNE. For better illustration, we choose five attributes with the most data and label the items with different attributes using different colors.

Figures 9 and 10 illustrate the item characteristics on both datasets. Through the visual representation of the figures, we come to the following conclusions: (1) Centralized training process can share and gather the attribute knowledge to some extent. This shows that a centralized training method with sufficient data can learn valuable knowledge, especially in an IID dataset. (2) Indeed, HPFL is able to aggregate representations of the same attributes across clients due to the collaboration, while AHPFL gathers the same attributes more accurately. It shows that the augmented mechanisms better augment the special valuable information for attributes. Specially, AHPFL-AE shows purer information for items of different attributes, which means the autoencoder method is notably robust for low-quality information that mines high-quality information.

7 DISCUSSION AND EMERGING TRENDS

Generally, there are two ways of evaluating federated learning. One focuses on global effects [60], which tests the effects of the global model locally, or using a global validation set. In such scenarios, in order to prevent the local model from over-fitting and leading to drift in the global model, researchers usually train several epochs (5-20) in each local model and then aggregate the local models. In such a setup, it needs multiple communication processes for aggregation to make sure that the global model works best, so communication efficiency is a point of concern. The other is the personalized federated learning [34, 58], which pays more attention to the effect of the local models updated by federated learning on respective local datasets. In such a scenario, the training settings can be different for each client to get the better local performances [32, 47, 64]. In that

case, more training processes can be done locally, even if it brings in the risk of over-fitting locally. Naturally, in such scenarios, the global effect tends to converge quickly in several epochs.

In recent years, federated learning has gained a lot of attention. In addition to global or personalized federated learning, a number of federated learning paradigms have also emerged [39]. These federated learning approaches focus on applying federated learning to different scenarios and tasks. Few shot [26], semi-supervised [37], or unsupervised federated learning [106] face the data scarce scenes. Multitask [78], transfer [52, 53], or domain adaptive [67] federated learning face the vastly different federated learning scenarios among clients. Besides, cluster federated learning [24, 65] and federated distillation learning [46] face the clients with different distributions or structures. Some work focuses on combining federated learning with practical applications, such as education [96], news recommendation [69], recommendation systems [101], medical [6, 11], and financial [81].

Federated learning is also being combined with other machine learning approaches to address privacy concerns in applications. Federated graph learning [62, 95, 99, 109] applies the federated learning into graph neural network for privacy-preserving. As a distributed algorithm with privacy protection characteristics, federated learning is also applied into the multi-agent [49, 94] or reinforcement learning [89] domain.

In addition to the development of federated learning paradigms, there are efforts to improve the effectiveness of universal federated learning. Some works focus on enhancing privacy guarantees for federated learning [79, 83, 93]. Some works hope to improve the efficiency of federated learning, from model compression [27, 72, 73], asynchronous communication [12], and other methods. There has also been a lot of research on general scenarios to improve the effectiveness of model fusion in federated learning [34, 38, 47]. As our work, we expect HPFL/AHPFL to provide a general method for model fusion in federated user modeling. We believe that these research directions could help the federated learning framework.

8 CONCLUSION

In this article, we focused on the federated user modeling problem with inconsistent clients. We first proposed a novel federated user modeling framework, called Hierarchical Personalized Federated Learning (HPFL) to serve federated learning in user modeling. More specifically, the HPFL aggregated and updated the local user models by components with isolated data. It greatly expanded the available information from multiple clients and flexibly allowed for information fusion and inheritance. Though HPFL effectively processed user models, overcoming statistical, privacy and model heterogeneity, it ignored the existence of quality heterogeneity in federated user modeling. We therefore further extended HPFL to the Augmented Hierarchical Personalized Federated Learning (AHPFL) by incorporating the typical augmented mechanisms. Specially, we designed two implementations, i.e., AHPFL-SVD and AHPFL-AE following SVD and AE, respectively. AHPFL could selectively filter out low-quality information from clients, which ultimately enhanced the availability of the local models and created a more robust framework. Our results on real-world user modeling tasks additionally demonstrated that our methods outperform existing federated learning methods, revealing them to be more suitable in wide user modeling scenarios.

In the future, we will consider various encryption techniques to give the framework even tighter privacy. Further, we hope to apply the framework to more scenarios in combination with the practical requirements. We hope that our work will lead to more studies in the future.

ACKNOWLEDGMENTS

Qi Liu acknowledges the support of the USTC-JD joint lab.

REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* 54, 11 (2006), 4311–4322.

[2] Hilal Asi, John Duchi, and Omid Javidbakht. 2019. Element level differential privacy: The right granularity of privacy. *arXiv preprint arXiv:1912.04042* (2019).

[3] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*. PMLR, 634–643.

[4] Marcus D. Bloice, Christof Stocker, and Andreas Holzinger. 2017. Augmentor: An image augmentation library for machine learning. *arXiv preprint arXiv:1708.04680* (2017).

[5] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[6] Theodora S. Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch. Paschalidis, and Wei Shi. 2018. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics* 112 (2018), 59–67.

[7] Peter Carey. 2018. *Data Protection: A Practical Guide to UK and EU Law*. Oxford University Press, Inc.

[8] Kamalika Chaudhuri and Claire Monteleoni. 2009. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems (NeurIPS)*. 289–296.

[9] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[10] Xiaolin Chen, Xuemeng Song, Ruiyang Ren, Lei Zhu, Zhiyong Cheng, and Liqiang Nie. 2020. Fine-grained privacy detection with graph-regularized hierarchical attentive representation learning. *ACM Transactions on Information Systems (TOIS)* 38, 4 (2020), 1–26.

[11] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020. FedHealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems* 35, 4 (2020), 83–93.

[12] Yang Chen, Xiaoyan Sun, and Yaochu Jin. 2019. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Transactions on Neural Networks and Learning Systems* 31, 10 (2019), 4229–4238.

[13] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. 2019. SecureBoost: A lossless federated learning framework. *arXiv preprint arXiv:1901.08755* (2019).

[14] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems (NeurIPS)*. 3571–3580.

[15] Yadolah Dodge. 2006. Coefficient of determination. *Alphascript Publishing* 31, 1 (2006), 63–64.

[16] Gitta O. Domik and Bernd Gutkauf. 1994. User modeling for adaptive visualization systems. In *Proceedings Visualization'94*. IEEE, 217–223.

[17] Fabon Dzogang, Thomas Lansdall-Welfare, Saatviga Sudhahar, and Nello Cristianini. 2015. Scalable preference learning from data streams. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*. 885–890.

[18] Michael Elad and Michal Aharon. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing* 15, 12 (2006), 3736–3745.

[19] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*. 278–288.

[20] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 1054–1067.

[21] Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan Eeik Tan, Suleiman A. Khan, and Muhammad Ammad-Ud-Din. 2020. Federated multi-view matrix factorization for personalized recommendations. *arXiv preprint arXiv:2004.04256* (2020).

[22] James Fogarty, Ryan S. Baker, and Scott E. Hudson. 2005. Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*. 129–136.

[23] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[24] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *arXiv preprint arXiv:2006.04088* (2020).

[25] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 249–256.

[26] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-shot federated learning. *arXiv preprint arXiv:1902.11175* (2019).

[27] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. 2020. FedBoost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*. PMLR, 3973–3983.
[28] Filip Hanzely and Peter Richtárik. 2020. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516* (2020).
[29] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.
[30] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.
[31] Hossein Hosseini, Sungrack Yun, Hyunsin Park, Christos Louizos, Joseph Soriaga, and Max Welling. 2020. Federated learning of user authentication models. *arXiv preprint arXiv:2007.04618* (2020).
[32] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. 2020. LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data. *PloS One* 15, 4 (2020), e0230706.
[33] Xixi Huang, Ye Ding, Zoe L. Jiang, Shuhan Qi, Xuan Wang, and Qing Liao. 2020. DP-FL: A novel differentially private federated learning framework for the unbalanced data. *World Wide Web* (2020), 1–17.
[34] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-IID data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7865–7873.
[35] Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for READING problems in standard tests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
[36] Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, Guoping Hu, et al. 2019. EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering* (2019).
[37] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. 2020. Federated semi-supervised learning with inter-client consistency. *arXiv E-prints* (2020), arXiv–2006.
[38] Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. 2019. Learning private neural language modeling with attentive aggregation. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[39] Shaoxiong Ji, Teemu Saravirta, Shirui Pan, Guodong Long, and Anwar Walid. 2021. Emerging trends in federated learning: From model fusion to federated x learning. *arXiv preprint arXiv:2102.12920* (2021).
[40] Di Jiang, Yongxin Tong, Yuanfeng Song, Xueyang Wu, Weiwei Zhao, Jinhua Peng, Rongzhong Lian, Qian Xu, and Qiang Yang. 2021. Industrial federated topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12, 1 (2021), 1–22.
[41] Qinghe Jing, Weiyan Wang, Junxue Zhang, Han Tian, and Kai Chen. 2019. Quantifying the performance of federated transfer learning. *ArXiv abs/1912.12795* (2019).
[42] Don H. Johnson. 2006. Signal-to-noise ratio. *Scholarpedia* 1, 12 (2006), 2088.
[43] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).
[44] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.
[45] Michael J. Kolen and Robert L. Brennan. 2013. *Test Equating: Methods and Practices*. Springer Science & Business Media.
[46] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* (2019).
[47] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).
[48] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of FedAvg on non-IID data. *arXiv preprint arXiv:1907.02189* (2019).
[49] Boyi Liu, Lujia Wang, and Ming Liu. 2019. Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters* 4, 4 (2019), 4555–4562.
[50] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized travel package recommendation. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 407–416.
[51] Qi Liu, Runze Wu, Enhong Chen, Guandong Xu, Yu Su, Zhigang Chen, and Guoping Hu. 2018. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 4 (2018), 1–26.

[52] Shuchang Liu, Shuyuan Xu, Wenhui Yu, Zuohui Fu, Yongfeng Zhang, and Amelie Marian. 2021. FedCT: Federated collaborative transfer for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 716–725.

[53] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. 2020. A secure federated transfer learning framework. *IEEE Intelligent Systems* 35, 4 (2020), 70–82.

[54] Yang Liu, Yingting Liu, Zhijie Liu, Yuxuan Liang, Chuishi Meng, Junbo Zhang, and Yu Zheng. 2020. Federated forest. *IEEE Transactions on Big Data* (2020).

[55] Hongyu Lu, Min Zhang, Weizhi Ma, Yunqiu Shao, Yiqun Liu, and Shaoping Ma. 2019. Quality effects on user preferences and behaviors in mobile news streaming. In *The World Wide Web Conference*. 1187–1197.

[56] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov. (2008), 2579–2605.

[57] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. Oakland, CA, USA, 281–297.

[58] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* (2020).

[59] Jiaxin Mao, Yiqun Liu, Noriko Kando, Min Zhang, and Shaoping Ma. 2018. How does domain expertise affect Users' search interaction and outcome in exploratory search? *ACM Transactions on Information Systems (TOIS)* 36, 4 (2018), 1–30.

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

[61] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963* (2017).

[62] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. 2021. Cross-node federated graph neural network for spatio-temporal data modeling. *arXiv preprint arXiv:2106.05223* (2021).

[63] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic federated learning. *arXiv preprint arXiv:1902.00146* (2019).

[64] Hesham Mostafa. 2019. Robust federated learning through representation matching and adaptive hyper-parameters. *arXiv preprint arXiv:1912.13075* (2019).

[65] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1234–1242.

[66] Maryam M. Najafabadi, Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, 1 (2015), 1–21.

[67] Daniel Peterson, Pallika Kanani, and Virendra J. Marathe. 2019. Private federated learning with domain adaptation. *arXiv preprint arXiv:1912.06733* (2019).

[68] Jacob Poushter et al. 2016. Smartphone ownership and internet usage continues to climb in emerging economies. *Pew Research Center* 22, 1 (2016), 1–44.

[69] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model training via federated learning. *arXiv preprint arXiv:2003.09592* (2020).

[70] Hanchi Ren, Jingjing Deng, and Xianghua Xie. 2020. Privacy preserving text recognition with gradient-boosting for federated learning. *arXiv preprint arXiv:2007.07296* (2020).

[71] Leonardo Filipe Rodrigues Ribeiro and Daniel Ratton Figueiredo. 2017. Ranking lawyers using a social network induced by legal cases. *Journal of the Brazilian Computer Society* 23, 1 (2017), 6.

[72] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. 2020. FetchSGD: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*. PMLR, 8253–8265.

[73] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-IID data. *IEEE Transactions on Neural Networks and Learning Systems* (2019).

[74] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. 824–831.

[75] Chuan Shi, Xiaotian Han, Li Song, Xiao Wang, Senzhang Wang, Junping Du, and S. Yu Philip. 2019. Deep collaborative filtering with multi-aspect information in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2019), 1413–1425.

[76] Shaoyun Shi, Weizhi Ma, Min Zhang, Yongfeng Zhang, Xinxing Yu, Houzhi Shan, Yiqun Liu, and Shaoping Ma. 2020. Beyond user embedding matrix: Learning to hash for modeling large-scale users in recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 319–328.

[77] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.
[78] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated multi-task learning. *arXiv preprint arXiv:1705.10467* (2017).
[79] Lichao Sun, Jianwei Qian, Xun Chen, and Philip S. Yu. 2020. LDP-FL: Practical private aggregation in federated learning with local differential privacy. *arXiv preprint arXiv:2007.15789* (2020).
[80] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation. In *Proceedings of The Web Conference 2020*. 837–847.
[81] Toyotaro Suzumura, Yi Zhou, Natahalie Baracaldo, Guangnan Ye, Keith Houck, Ryo Kawahara, Ali Anwar, Lucia Larise Stavarache, Yuji Watanabe, Pablo Loyola, et al. 2019. Towards federated graph learning for collaborative financial crimes detection. *arXiv preprint arXiv:1909.12946* (2019).
[82] Aleksei Triastcyn and Boi Faltings. 2019. Federated learning with Bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2587–2596.
[83] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 1–11.
[84] Jacob M. Victor. 2013. The EU General Data Protection Regulation: Toward a property regime for protecting data privacy. *Yale LJ* 123 (2013), 513.
[85] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*. 1096–1103.
[86] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, 12 (2010).
[87] W. Gregory Voss. 2016. European Union data privacy law reform: General Data Protection Regulation, privacy shield, and the right to delisting. *The Business Lawyer* 72, 1 (2016), 221–234.
[88] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. 2020. Neural cognitive diagnosis for intelligent education systems. In *34th AAAI Conference on Artificial Intelligence, AAAI 2020*. 6153–6161.
[89] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020. Optimizing federated learning on non-IID data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1698–1707.
[90] Haoyu Wang, Yuanchun Li, Yao Guo, Yuvraj Agarwal, and Jason I. Hong. 2017. Understanding the purpose of permission use in mobile apps. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 1–40.
[91] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1064–1072.
[92] Yansheng Wang, Yongxin Tong, and Dingyuan Shi. 2020. Federated latent dirichlet allocation: A local differential privacy based framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6283–6290.
[93] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.
[94] Xiguang Wei, Quan Li, Yang Liu, Han Yu, Tianjian Chen, and Qiang Yang. 2019. Multi-agent visualization for explaining federated learning. In *IJCAI*. 6572–6574.
[95] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. FedGNN: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925* (2021).
[96] Jinze Wu, Zhenya Huang, Qi Liu, Defu Lian, Hao Wang, Enhong Chen, Haiping Ma, and Shijin Wang. 2021. Federated deep knowledge tracing. In *Proceedings of the 14th International Conference on Web Search and Data Mining*.
[97] 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 *Proceedings of The Web Conference 2021*. 957–968.
[98] Peizhi Wu, Yi Tu, Zhenglu Yang, Adam Jatowt, and Masato Odagaki. 2018. Deep modeling of the evolution of user preferences and item attributes in dynamic social networks. In *Companion Proceedings of The Web Conference 2018*. 115–116.
[99] Han Xie, Jing Ma, Li Xiong, and Carl Yang. 2021. Federated graph classification over non-IID graphs. *Advances in Neural Information Processing Systems* 34 (2021).
[100] Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang. 2009. User language model for collaborative personalized search. *ACM Transactions on Information Systems (TOIS)* 27, 2 (2009), 1–28.

[101] Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. 2020. Federated recommendation systems. In *Federated Learning*. Springer, 225–239.
[102] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.
[103] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. 2019. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 13, 3 (2019), 1–207.
[104] Georgios N. Yannakakis and Julian Togelius. 2018. *Artificial Intelligence and Games*. Vol. 2. Springer.
[105] Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu, and Xiaofang Zhou. 2015. Dynamic user modeling in social media systems. *ACM Transactions on Information Systems (TOIS)* 33, 3 (2015), 1–44.
[106] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. 2020. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982* (2020).
[107] Fuzheng Zhang, Nicholas Jing Yuan, Kai Zheng, Defu Lian, Xing Xie, and Yong Rui. 2016. Exploiting dining preference for restaurant recommendation. In *Proceedings of the 25th International Conference on World Wide Web*. 725–735.
[108] Hongke Zhao, Qi Liu, Hengshu Zhu, Yong Ge, Enhong Chen, Yan Zhu, and Junping Du. 2017. A sequential approach to market state modeling and analysis in online P2P lending. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 1 (2017), 21–33.
[109] Longfei Zheng, Jun Zhou, Chaochao Chen, Bingzhe Wu, Li Wang, and Benyu Zhang. 2021. ASFGNN: Automated separated-federated graph neural network. *Peer-to-Peer Networking and Applications* 14, 3 (2021), 1692–1704.
[110] Ingrid Zukerman and David W. Albrecht. 2001. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction* 11, 1–2 (2001), 5–18.

Received 3 June 2021; revised 11 June 2022; accepted 2 August 2022