ACE-pFL: Accurate, Efficient Personalized Federated Learning With Knowledge Distillation

Kun He, *Member, IEEE*, Hao Bai, Yuqing Li, *Member, IEEE*, Jing Chen, *Senior Member, IEEE*, and Ruiying Du

***Abstract*— Personalized Federated Learning (pFL) can collaboratively personalize models for multiple clients without sharing their private data. However, many pFL methods rely on server-side model parameters aggregation, which requires all models to have the same structure and size. One promising approach is leveraging knowledge distillation (KD) to transfer knowledge between models by exchanging soft predictions rather than model parameters, thus training heterogeneous models. Nevertheless, existing KD-based pFL solutions suffer from accuracy loss due to inadequate knowledge extraction as well as huge computing and communication overheads. In this paper, we present an accurate and efficient KD-based pFL framework, called ACE-pFL. Specifically, we first propose a privacy-preserving client clustering to reduce the impact of non-independent and identically distributed (non-IID) data on model accuracy and convergence, grouping clients with similar data distributions into the same cluster. Since the distillation temperature of traditional KD is fixed, which does not consider the dynamic model training process, we design a dynamic distillation temperature adjustment to accommodate this process, where clients incrementally increase the distillation temperature as training proceeds to facilitate model generalization to new data. Finally, we employ the triple distillation strategy to provide diverse and abundant knowledge, including explicit global knowledge, implicit local knowledge, and implicit global knowledge. Experiments on multiple datasets and tasks show that compared with existing schemes, ACE-pFL can significantly improve the test accuracy by 17.18%, reduce the training time by 57% and the communication overhead by 59.12× on average.**

***Index Terms*— Personalized federated learning, knowledge distillation, non-IID data.**

I. Introduction

FEDERATED Learning (FL) is currently one of the most promising distributed learning frameworks, which allows multiple clients to collaboratively train a global model with the coordination of a central server. The primary goal is to provide privacy protection for clients' local data and address the "data islands" problem. Its applications engage industries including finance [1], [2], healthcare [3], [4], and manufacturing [5], [6].

Nevertheless, a single global model trained from all clients may not satisfy those whose tasks or data distributions significantly deviate from the rest [7], [8]. Consequently, employing personalized models seems to be an effective solution in FL, namely, personalized federated learning (pFL). However, most recent works regarding pFL [9], [10], [11], [12] inevitably have to be identical in model structure and size to aggregate parameters from all clients. In practical scenarios, clients often prefer to employ customized model architectures (i.e., different structures and sizes) to accommodate heterogeneity in computation, communication, and storage capabilities, etc.

One promising approach is leveraging knowledge distillation (KD) to transfer knowledge by exchanging soft predictions instead of using model parameters. Therefore, KD-based pFL methods [13], [14], [15], [16] have been investigated for collaborative training of heterogeneous models. Unfortunately, KD-based pFL faces three main challenges: (1) due to the distributed nature of the collection, data across various clients is typically non-independent and identically distributed (non-IID), which leads to low accuracy and excessive training overhead (i.e., training time and communication overhead) [17], [18], [19]; (2) the distillation temperature of traditional KD is fixed, which does not consider the dynamic model training process [20], [21], [22]; (3) simple KD has a single source of knowledge, which may ignore knowledge from other sources, resulting in poor model performance [14], [15], [21].

In this paper, we present an accurate and efficient KD-based pFL framework to address these issues, called ACE-pFL.

Received 16 July 2024; revised 3 November 2024 and 1 January 2025; accepted 5 January 2025; approved by IEEE TRANSACTIONS ON NETWORKING Editor C. Wu. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3102100, in part by the State Key Laboratory of Intelligent Transportation System under Project 2024-B004, in part by the Fundamental Research Funds for the Central Universities under Grant 2042023kf0120, in part by the National Natural Science Foundation of China under Grant 62302343 and Grant 62172303, in part by the Key Research and Development Program of Hubei Province under Grant 2024BAB018, in part by the Hubei Province International Science and Technology Collaboration Program under Grant 2023EHA044, in part by the Rizhao Natural Science Foundation under Grant RZ2021ZR4, in part by the Key Research and Development Program of Shandong Province under Grant 2022CXPT055, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515110396. *(Corresponding author: Kun He.)*

Kun He is with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the State Key Laboratory of Intelligent Transportation System, Beijing 100083, China (e-mail: hekun@whu.edu.cn).

Hao Bai, Jing Chen, and Ruiying Du are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: bh0036@whu.edu.cn; chenjing@whu.edu.cn; duraying@whu.edu.cn).

Yuqing Li is with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with Shenzhen Research Institute, Wuhan University, Shenzhen 518057, China (e-mail: li.yuqing@whu.edu.cn).

Digital Object Identifier 10.1109/TON.2025.3527467

First, there may exist many clients that have similar data distributions. If these clients can be aggregated, they can mutually benefit from expanding more valuable data, which can be used for personalized models, thereby alleviating the non-IID issue. Therefore, we design an Earth Mover's Distance (EMD)-based client clustering, which groups clients with similar data distributions into the same cluster to mitigate the adverse effects of non-IID. Specifically, we randomly select a client, and the remaining clients individually measure the dissimilarity of their data distributions with it, after which they are grouped using the K-means algorithm [23].

Second, the distillation temperature in KD affects soft prediction distributions, which in turn influences the effectiveness of knowledge transfer. If the temperature can be dynamically adjusted, the performance of knowledge transfer will be further improved. The model may require diverse information at different stages of training [21]. For example, in the early stages, a lower temperature should be used to allow the model to focus on the most critical knowledge to accelerate model convergence. As it gradually converges, a higher temperature allows the model to consider more classes, which can help the model generalize better to new data. Therefore, we propose dynamic distillation temperature adjustment, in which clients only need to continuously increase the distillation temperature to accommodate pFL as the training rounds increases.

Third, limited sources of knowledge result in poor model performance and generalization ability, making it challenging to handle complex learning tasks. If we can provide diverse and abundant knowledge, it can further improve distillation efficiency. Therefore, we design the triple distillation strategy, which enables local personalized models to learn explicit and implicit knowledge. Concretely, we distill certain explicit global knowledge from the current aggregated soft predictions and distill the other two parts of implicit knowledge from the past soft predictions of the local model (i.e., implicit local knowledge) and the past aggregated soft predictions (i.e., implicit global knowledge), thus providing explicit and implicit knowledge for pFL. Therefore, ACE-pFL can improve the model performance by providing rich multi-source knowledge.

To demonstrate the generality of our framework, we utilize five benchmark datasets (e.g., CIFAR-100) covering three modalities (i.e., image, text, and audio) on multiple model architectures including ResNet50, etc. We also compare our framework with other personalized FL schemes in terms of testing accuracy, training time, and communication overhead. The results highlight the advantage of ACE-pFL in term of high accuracy and low overhead, confirming its potential for practical deployment in real-world scenarios.

To summarize, our main contributions are as follows:

- We propose an accurate and efficient KD-based pFL framework, ACE-pFL, which improves model accuracy, accelerates model training, and reduces communication overhead. This provides a possible solution for KD-based pFL deployment in practical scenarios.
- We mitigate the adverse effects of non-IID data by designing an EMD-based client clustering. Then, we also propose dynamic distillation temperature adjustment to accommodate the dynamic pFL process. Finally, we employ the triple distillation strategy to provide rich multi-source knowledge.
- We evaluate the performance of ACE-pFL through extensive experiments. The results show that ACE-pFL can achieve the model accuracy improvement of around 2.97%–56.93%, reduce the training time by 7.78× at most, and reduces the communication overhead 184× at most compared with the baselines under non-IID settings.

II. Preliminaries

We first briefly describe pFL. After that, we introduce KD-based pFL.

A. Personalized Federated Learning

pFL customizes individual models for each client collaboratively to accommodate the specific needs of each client [9], [10], [15], [16]. In each training round, each client trains a local model using its training data, and uploads the intermediate result (e.g., model parameters) to the server; the server aggregates the clients' intermediate results, and sends the aggregated result to the clients; each client achieves personalization by updating the local model with specific requirements.

Assume that there are N clients and a remote server. Each client n can only access to its private dataset $\mathcal{D}_n \triangleq \{x_i, y_i\}$, where x_i is the i-th data sample, and $y_i \in \{1, 2, \cdots, C\}$ is the label of x_i. Denote the number of data samples in dataset $\mathcal{D}_n$ as D_n. The global dataset $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$ is formed by concatenating local datasets, and $D = \sum_{n=1}^{N} D_n$ is the number of data samples in dataset $\mathcal{D}$. Additionally, we need to introduce a public dataset $\mathcal{D}_0$ to evaluate the generalization ability of each client's personalized model. The public dataset can be shared and used by all clients. In contrast to the private dataset, the public dataset is made publicly available with the consent of their contributors (e.g., volunteers), which is often collected to facilitate research, validate algorithms, or solve specific problems. Our goal is to minimize the $\arg\min \mathcal{L}(w) := \sum_{n=1}^{N} \frac{D_n}{D} \mathcal{L}_{ptrain,n}(w_n)$, where $\mathcal{L}_{ptrain,n}(w_n)$ is the personalized loss function of client n, and $w = [w_1, w_2, \cdots, w_N]$ is the connection of all weights. However, such model parameters sharing can certainly be a straightforward way of information exchange, it leads to a huge communication overhead.

B. KD-Based pFL

KD, as a knowledge transfer method, has been widely applied to pFL, where only soft predictions are exchanged, avoiding the need to share model parameters. KD allows multiple clients to absorb the inferred knowledge from others by comparing outputs on a public dataset [24], [25]. Specifically, the local client's personalized model learns from a global perspective by simulating the aggregated soft predictions distribution as the model output via Kullback-Leibler (KL) divergence loss, which is defined as follows.

$$\mathcal{L}_{KD}(s_n^r(x), s^r) = T^2 KL(s_n^r(x) || s^r), \tag{1}$$

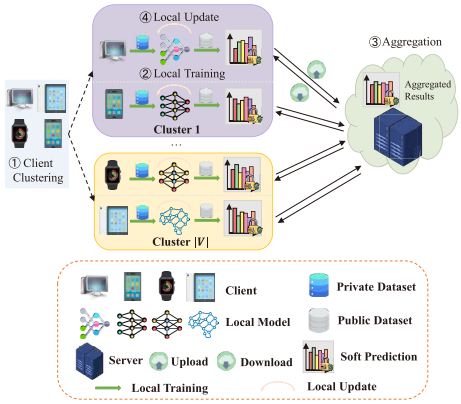


Fig. 1. An overview of ACE-pFL. Based on the data distribution of the clients, they are grouped into different clusters (①). Within each cluster, clients perform local training using their private dataset and test on a public dataset to obtain local soft predictions (②). The server receives local soft predictions from each cluster, aggregates them to generate the global soft predictions, and sends the aggregated results back to the clients (③). Clients then update locally based on the aggregated results and proceed to the next round of local training (④).

where T is the temperature parameter, used to control the softening degree of outputs. x denotes some data samples from a public dataset $\mathcal{D}_0$. $s_n^r(x)$ captures soft predictions of the client n, calculated with the softmax of logits z^n, i.e., $s_n^r(x) = [\frac{exp(z_1^n/T)}{\sum_{c=1}^{C} exp(z_c^n/T)}, \cdots, \frac{exp(z_C^n/T)}{\sum_{c=1}^{C} exp(z_c^n/T)}]$, where C is the number of classes. $s^r = [\frac{1}{N}\sum_{n=1}^{N} \frac{exp(z_1^n/T)}{\sum_{c=1}^{C} exp(z_c^n/T)}, \cdots, \frac{1}{N}\sum_{n=1}^{N} \frac{exp(z_C^n/T)}{\sum_{c=1}^{C} exp(z_c^n/T)}]$ is the aggregated result in the r-th round. Finally, client n updates the local model according to Eq. (1) and $\mathcal{L}_{ptrain,n}(w_n)$.

III. ACE-pFL Design

We present the design of our accurate and efficient KD-based pFL. We first give an overview of ACE-pFL. Then we propose the EMD-based client clustering and two optimization methods on the performance of KD, i.e., dynamic distillation temperature adjustment and the triple distillation strategy.

A. Overview

Architecture. ACE-pFL consists of multiple clients and a server, as shown in Figure 1. Since data across different clients is usually non-IID, it leads to low accuracy and high training overhead [17], [18], [19]. An EMD-based client clustering module is conducted to group clients with similar data distribution into the same cluster to mitigate the adverse effects of non-IID. Then, each client performs local model training on its private dataset and completes model prediction on the public dataset with dynamic distillation temperature adjustment. After that, the client uploads the soft predictions to the server, which completes the aggregation. Finally, the client downloads the aggregated results, performs local model update based on the triple distillation strategy, and continues a new round of local training. When the number of rounds reaches the preset value, the pFL is completed. The details of ACE-pFL are shown in Algorithm 1.

Design Goals. Our goal is to design an accurate and efficient KD-based pFL framework. Our scheme faces challenges such as low model accuracy, high training time, and excessive communication burden. Against the above challenges, we describe three objectives as follows:

- **Model accuracy.** Enhance model accuracy by accounting for the specific data distribution of individual clients.
- **Computation efficiency.** We should accelerate model convergence (i.e., fewer rounds) to reduce training time.
- **Communication efficiency.** The scheme reduces communication overhead between clients and the server.

Algorithm 1 ACE-pFL

Input: Local dataset $\mathcal{D}_n$ of client n, public dataset $\mathcal{D}_0$, number of clients N, training rounds R, number of clusters $|V|$, initial temperature T_0, filtering parameter K, learning rate η_1, η_2.

Output: Local personalized models $w = [w_1, w_2, \cdots, w_N]$.

Execute the EMD-based client clustering 2.
// Client $n \in [N]$.
Initialize the local personalization model w_n.
for $r = 1, \cdots, R$ **do**
  Local training on its private dataset $\mathcal{D}_n$.
  Adjust the distillation temperature based on Eq. (2).
  Compute the softmax value $s_n^r(x)$ of the logits z^n.
  Perform top-K filtering, and upload $s_n^r(x)$ to server.
  Update the local personalized model according to the aggregated result base on Eq. (4).
end for
// Server.
for $r = 1, \cdots, R$ **do**
  Aggregate $s_n^r(x)$ base on Eq. (3).
  Distribute the aggregated result s^r to all clients.
end for

B. EMD-Based Client Clustering

Data across various clients is often non-IID, which reduces the local personalized model accuracy. We design an EMD-based client clustering to group clients with similar data distribution into the same cluster. By exploiting the similarity between clients to aggregate them, information can be obtained from more valuable data to personalize the local model, thereby improving the local personalized model accuracy.

A straightforward and primitive method is for each client to upload its own data distribution and let the server complete the clustering. We adopt the EMD as the metric to calculate the similarity of data distribution between each pair of clients [26]. A larger EMD between data distributions of two clients indicates stronger dissimilarity. Specifically, the i-th client's data distribution is defined as $R_i = [r_1^i, r_2^i, \cdots, r_C^i]$ with C classes, where $r_{c\in[C]}^i$ denotes the frequency of each class of data samples in its private dataset $\mathcal{D}_i$; the j-th client's data distribution is defined as $R_j = [r_1^j, r_2^j, \cdots, r_C^j]$ with C classes, where $r_{c\in[C]}^j$ denotes the frequency of each class of data samples in its private dataset $\mathcal{D}_j$. Let $E = [e_{m,n}]$ be the ground distance between $r_{m\in[C]}^i$ and $r_{n\in[C]}^j$, i.e., $E[m][n] =$

$|r^i_{m\in[C]} - r^j_{n\in[C]}|$. Let $F = [f_{i,j}]$ be a 0/1 matrix, indicating whether to move the r^i_m in R_i to the r^j_n in R_j. We define the objective function $\min_F \sum_{m=1}^{C}\sum_{n=1}^{C} f_{m,n}e_{m,n}$, which subjects to the constraints:

$$f_{m,n} \geq 0,\ 1 \leq m \leq C, 1 \leq n \leq C$$
$$\sum_{n=1}^{C} f_{m,n} \leq C,\ 1 \leq m \leq C$$
$$\sum_{m=1}^{C} f_{m,n} \leq C,\ 1 \leq n \leq C$$

The optimal F is found by solving this linear optimization problem. Their similarity can be expressed by $E(R_i, R_j) = \frac{\sum_{m=1}^{C}\sum_{n=1}^{C} f_{m,n}e_{m,n}}{\sum_{m=1}^{C}\sum_{n=1}^{C} f_{m,n}}$. Finally, the server computes the similarity of any of two clients and completes clustering based on the similarity matrix. However, the complexity of it is $O(N^2)$, which may introduce significant computation and communication overhead when the number of clients is large.

To address this issue, we propose an EMD-based client clustering by computing the similarity of only one client instead of all clients. Concretely, we first randomly select the z-th client, and then compute the EMD between it and other clients, obtaining a similarity vector $[E(R_1, R_z), E(R_2, R_z), \cdots, E(R_n, R_z)]$. We do this for two main reasons: (1) comparability; (2) transitivity. First, if $E(R_i, R_z)$ is low, we consider that the i-th client has similar data distribution with the z-th client. This is because EMD is a standardized metric used to measure the difference between two data distributions. Based on optimal transport theory [27], the key idea is that a smaller EMD distance indicates higher similarity between the two distributions, ensuring comparability between the data distributions of different clients. Second, if $E(R_x, R_z)$ and $E(R_y, R_z)$ is close, the x-th and y-th clients can be considered to have similar data distribution. This is because the invariance and balance principles in optimal transport theory provide the theoretical foundation for the transitivity of EMD distance [27]. Invariance suggests that when multiple distributions are similar, their distribution characteristics will not change drastically. Specifically, if two distributions R_x and R_y are both similar to a third distribution R_z, the similarity between R_x and R_y will be preserved, resulting in a small EMD distance between them. The balance principle further indicates that if both R_x and R_y are similar to R_z, their difference will not increase significantly, as their similarity is balanced and propagated through R_z. These two principles ensure the transitivity of EMD distance, allowing the similarity between distributions to propagate progressively without significant change. This design makes sense because the choice of data distribution is often limited (e.g., the number of classes is not infinite). Therefore, this vector is sufficient to determine the similarity between clients. Finally, the server employs a clustering algorithm such as K-means [23] to cluster the similarity vector. Compared with the original method, the complexity is reduced from $O(N^2)$ to $O(N)$, effectively accelerating the client clustering. The details are shown in Algorithm 2.

Algorithm 2 EMD-Based Client Clustering

Input: Data distribution R_n of client n, number of clients N, number of clusters $|V|$.
Output: Client n belongs to cluster v.

1: Select a client z at random.
// Client $n \in [N]$**.**
2: Compute the similarity $E(R_n, R_z)$with client z.
3: Upload the result to server.
// Server.
4: Employ a K-means algorithm to cluster the similarity vector.
5: Broadcast the outputs to all clients.

C. Dynamic Distillation Temperature Adjustment

In KD, temperature is utilized to control the sharpness of soft prediction distributions. The lower temperature makes soft prediction distributions sharper and therefore more concentrated on the class with the largest probability, whereas the higher temperature makes the distribution flatter [20]. The model may require different information at various stages of training. According to the characteristics of the model training, the lower temperature should be used to let the model focus on the most critical knowledge to accelerate model convergence in the early stage of pFL training [21]. As pFL proceeds, the model will gradually converge, and the higher temperature allows the model to take into account more classes, which can help the model to better generalize to new data [22]. However, in traditional KD, the fixed temperature results in static knowledge transfer that cannot dynamically adapt to the local model training. Therefore, it is essential to design a strategy for dynamically adjusting the temperature. This process can dynamically adjust the attention of the model, ensuring that the knowledge transfer can be fully utilized and balanced during different stages of training.

Local Training. In the r-th communication round, we fix the number of classes C and train w_n for several epochs locally. Specifically, the n-th client first trains w_n relying on its own private dataset $\mathcal{D}_n$ by applying a gradient descent step: $w_n \leftarrow w_n - \eta_1 \nabla \mathcal{L}_n(w_n)$, where η_1 is the learning rate. The loss function of client n: $\mathcal{L}_n(w_n) = \frac{1}{D_n}\sum_{i=1}^{D_n} \mathcal{L}_{CE}(w_n; x_i, y_i)$, where $\mathcal{L}_{CE}$ is the cross-entropy loss function that measures the difference between the predicted results and the true labels on the private dataset. Then, client n completes model performance testing on the public dataset $\mathcal{D}_0$ and outputs the softmax value of the logits z^n, i.e., $s^r_n(x) = \frac{exp(z^n_c/T_r)}{\sum_{c=1}^{C} exp(z^n_c/T_r)}$, where T_r is the distillation temperature in the r-th round, and x denote data samples from $\mathcal{D}_0$. The designed T_r primarily adheres to the following three design principles: (1) the rate of increase for the distillation temperature transitions from fast to slow; (2) after reaching a certain temperature, it undergoes piecewise repetitive changes, where the rate of increase returns to the initial speed of the previous stage; and (3) it possesses good flexibility and adjustability. Therefore, T_r is computed as follows:

$$T_r = T_0 + \frac{k_1}{1 + e^{-k_2}} \lfloor \frac{r}{r_0} \rfloor + \frac{2k_1}{1 + e^{-k_2 \Delta r}},$$

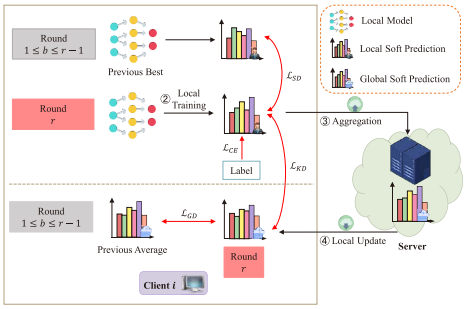


Fig. 2. The client i's workflow of ACE-pFL in the r-th round. During training, each client trains its local model with four kinds of knowledge: labels ($\mathcal{L}_{CE}$), aggregated results($\mathcal{L}_{KD}$), and past predictions from previous rounds in both local ($\mathcal{L}_{SD}$) and global ($\mathcal{L}_{GD}$) perspectives.

where

$$\Delta r = r - r_0 \lfloor \frac{r}{r_0} \rfloor, \tag{2}$$

where k_1, k_2, and r_0 denotes dynamic distillation temperature adjustment hyperparameters. T_0 denotes the initial temperature. $\frac{k_1}{1+e^{-k_2}}$ is similar to the sigmoid function, which makes the growth rate change from fast to slow. The use of $\lfloor \frac{r}{r_0} \rfloor$ means that when r reaches a certain threshold r_0, the function will experience piecewise changes. The parameters k_1 and k_2 can control the growth rate of the function, making it very flexible and adaptable. When the total number of classes is large, the dimension of soft predictions will be high, leading not only to increased communication overhead but also to a lot of noise (due to the addition to each coordinate). To address these issues, we exploit a simple but effective filtering strategy: top-K. In top-K filtering, we only keep the top K largest soft prediction probabilities, and the remaining probabilities are set to zero for each client. This strategy can not only retain important information, but also avoid unimportant information interfering with KD.

Aggregation. To obtain the aggregated results, the server aggregates the soft prediction distributions uploaded by the clients according to the EMD-based client clustering:

$$s^r = \sum_v \frac{\sum_{n \in v} D_v}{D} \{ \frac{1}{|v|} \sum_{n \in v} s_n^r(x) \}, \tag{3}$$

where $|v|$ is the number of clients in cluster v. Considering the previous client clustering results, the aggregation rule is the intra-cluster arithmetic average and the inter-cluster weighted average, where the weights are assigned according to the number of data samples in the cluster. It has two benefits: (1) intra-cluster arithmetic averaging ensures that all clients within a cluster contribute equally to the aggregated result. This is important because it reflects the central tendency of the data within that cluster, preventing any single client from disproportionately influencing the outcome; (2) inter-cluster weighted averaging accounts for the varying sizes of clusters. Larger clusters, which may have more data and provide more reliable updates, can contribute more to the final aggregation. This reflects the principle that more representative data should influence the overall model more.

D. The Triple Distillation Strategy

In traditional KD, reliance on aggregated global results as the sole knowledge source may overlook other valuable information, resulting in less-than-ideal distillation effectiveness. Therefore, we propose a triple distillation strategy (i.e., local-to-global distillation, local-to-local distillation, global-to-global distillation) to perform knowledge distillation from different sources. Specifically, we extract explicit knowledge from aggregated global soft predictions of the latest round and exploit implicit knowledge in both local and global perspectives of previous rounds. The key idea of using models trained in previous rounds is to treat their outputs as different perspectives of the features [28]. This method helps the latest model integrate multiple feature representations, providing more information for training, thereby enhancing the learning ability of the model. Therefore, ACE-pFL can improve the model performance by providing rich multi-source knowledge.

Local-to-Global Distillation. We extract global knowledge by performing knowledge distillation on local soft predictions and aggregated global soft predictions. The key idea is that the ensemble of local models often improves personalization performance compared to the performance of individual models. To extract the knowledge, we take the aggregated results of the local model soft predictions as the global knowledge. By imitating the global knowledge, the local personalized model is able to learn from the global. Specifically, this distillation loss is given by the KL divergence between local soft predictions $s_n^r(x)$ and aggregated global soft predictions s^r:

$$\mathcal{L}_{KD}(s_n^r(x), s^r) = T_r^2 KL(s_n^r(x) || s^r).$$

Local-to-Local Distillation. To fully exploit implicit knowledge in the local model, a local and local distillation strategy is proposed to enable the local model to extract knowledge from itself. Specifically, the local personalized model in the previous rounds performs knowledge distillation with the personalized model in the current round. The key idea is that we treat the outputs of the local personalization model from the previous rounds as a different representation of the features, which provides more information for personalized training.

We design an efficient method to perform local and local knowledge distillation. Specifically, the local personalized models in the previous rounds are considered as candidates, and the best performing one is selected for knowledge distillation with the local personalized model in the current round. Given the large size of deep learning models, we maintain only one optimal model per client and replace it when a more optimal model becomes available. The distillation is implemented as follows:

$$\mathcal{L}_{SD}(s_n^r(x), s_n^b(x)) = T_r^2 KL(s_n^r(x) || s_n^b(x)),$$

where $s_n^b(x)$ is the soft predictions in b-th round.

Global-to-Global Distillation. To fully extract implicit knowledge in the global perspective, we propose a global and global distillation strategy, which enables the local model to better focus on the global dynamics. Specifically, the aggregated soft predictions in previous rounds are used for

knowledge distillation with the aggregated soft predictions in the current round. The main idea is that we focus on the convergence rate of the model from the aggregated results of previous rounds, which provides guidance for the local personalized model training.

Similar to the local-to-local approach, we average the aggregated soft predictions in the previous rounds and perform knowledge distillation with the aggregated soft predictions in the current round. To reduce the local storage overhead, we store only the average of aggregated soft predictions for each client in the previous rounds, updating it when a new round is completed. The distillation is as follows:

$$\mathcal{L}_{GD}(s^r, \frac{\sum_{i=1}^{r-1} s^i}{r-1}) = T_r^2 KL(s^r || \frac{\sum_{i=1}^{r-1} s^i}{r-1}).$$

Local Update. Finally, we utilize four kinds of knowledge to define the loss function for the local personalized model update: the loss with the public dataset labels, the local-to-global distillation loss, the local-to-local distillation loss, and the global-to-global distillation loss. The loss with the public dataset labels is calculated using the cross-entropy loss function $\mathcal{L}_{CE}$. The personalized loss function of client n is a weighted combination of four loss terms:

$$\mathcal{L}_{ptrain,n}(w_n) = \alpha_1 \mathcal{L}_{CE} + \alpha_2 \mathcal{L}_{KD} + \alpha_3 \mathcal{L}_{SD} + \alpha_4 \mathcal{L}_{GD}, \tag{4}$$

where $\alpha_1, \alpha_2, \alpha_3$, and α_4 are the corresponding weight hyperparameters. We discuss the effect of different weight hyperparameters on ACE-pFL in Section IV-D. Each client completes a local update based on the obtained knowledge by applying a gradient descent step: $w_n \leftarrow w_n - \eta_2 \triangledown \mathcal{L}_{ptrain,n}(w_n)$, where η_2 is the learning rate. Then, it continues a new round of local training. When the number of rounds reaches the preset value, the pFL task is completed. Figure 2 shows the client i's workflow of ACE-pFL in the r-th round.

E. Discussion

In pFL, privacy guarantees are provided by keeping the data local and only intermediate results (e.g., data distributions and soft predictions) are passed. However, previous works have demonstrated that if soft predictions may contain sensitive information, the model is likely to transfer the privacy of its own training data to another model [20], [29], [30]. Additionally, the local data distribution should be shared among clients in EMD-based client clustering, which may also lead to privacy leakage. Differential privacy is an effective means to prevent privacy leakage, and can be well compatible with ACE-pFL. The basic idea is to inject noise into the soft predictions (data distributions) of each batch of data, thereby reducing the differences between soft predictions (data distributions). The data distribution is a $1 \times C$ vector, which has a small data volume and only needs to be transmitted once in Algorithm 1, unlike the repeated transmission of soft predictions during pFL. Therefore, it does not incur high communication overhead. Furthermore, the number of the public dataset may become an issue in practice. We can apply random transformations such as rotation, translation, cropping, padding, and horizontal flipping, which correspond to observing the same object from different angles. Alternatively, we can modify the color, brightness, sharpness, contrast, and saturation of images, representing variations in observing the same object to different degrees. The public dataset is also available on data modeling and analysis competition platforms such as Kaggle. Therefore, the problem of insufficient data volume can be alleviated by the above way.

IV. Evaluation

Our evaluation answers the following Research Questions (RQs).

- **RQ1:** What is the performance of ACE-pFL, and how does it compare to other schemes?
- **RQ2:** What is the performance of EMD-based client clustering in distinguishing clients with different data distributions?
- **RQ3:** How do varying parameters, such as the number of clusters, initial temperature, privacy budget, Top-K, and the weights of loss terms affect the performance of ACE-pFL?
- **RQ4:** Why can ACE-pFL improve the accuracy of KD-based pFL?

A. Implementation and Experimental Setup

Implementation. We implement ACE-pFL based on the PyTorch framework in Python 3.8. All experiments are conducted on a server running Ubuntu 18.04, which is equipped with an Intel(R) Xeon(R) Gold 6133 CPU @ 2.5GHz, 128GB RAM, and three NVIDIA 4090 GPUs with 24GB memory each. The CUDA version is 12.0. Our evaluation primarily employs three metrics: the average testing accuracy of all local personalized models, the total training time, and communication overhead.

Datasets. The datasets encompass three classification tasks, namely image (i.e., CIFAR-10 [31], CIFAR-100 [31], and CheXpert [32]), text (i.e., Reuters-21578 [33]), and audio (i.e., Speech Commands [34]), as follows:

- CIFAR-10 contains 60,000 32 $\times$ 32 color images equally divided into 10 classes, while 50,000 is used for training and 10,000 is used for testing.
- CIFAR-100 has 100 classes with 600 images each, which is equivalent to CIFAR-10 augmented version.
- CheXpert is a large chest X-ray dataset comprising 224,316 images. Each image is associated with predictions for 14 common chest X-ray findings.
- Reuters-21578 is a collection of documents containing news articles, with 90 classes, 7769 training files, and 3019 testing files. We extract 20 classes from them.
- Speech Commands is an audio dataset of 10 single spoken words designed to help train keyword spotting systems. The training data has about 40,000 samples.

We randomly select 10% from each of the above datasets as the public dataset.

TABLE I
THE TESTING ACCURACY (%) OF DIFFERENT SCHEMES IN TWO NON-IID SETTINGS (I.E., $\beta = 0.1$ AND $\pi = 2$) ON FIVE DATASETS (I.E., CIFAR-10, CIFAR-100, CHEXPERT, REUTERS-21578, AND SPEECH COMMANDS)

Non-IID	$\beta = 0.1$ (Accuracy ↑)					$\pi = 2$ (Accuracy ↑)				
Schemes	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
FedAvg	24.32	15.62	53.32	50.94	47.93	33.63	18.93	52.12	52.39	52.36
FedProx	38.22	19.79	58.93	52.39	62.16	35.73	22.93	59.03	50.39	66.32
SCAFFOLD	49.93	33.51	62.36	53.69	53.19	52.79	36.49	63.74	56.81	58.41
Per-FedAvg	50.59	42.92	67.39	68.36	59.46	53.95	40.79	69.62	64.23	62.25
pFedMe	64.98	48.69	69.99	73.69	68.26	68.32	49.58	70.49	66.26	69.32
CD^2-pFed	62.84	35.71	68.07	69.92	65.26	65.98	36.98	66.03	63.51	67.98
FedALA	75.64	55.95	74.48	78.92	76.68	78.63	57.93	72.82	70.23	72.36
FedDF	70.29	39.33	70.32	72.36	68.93	73.98	41.46	71.69	68.92	69.46
FedAD	73.11	49.44	78.57	76.36	72.39	75.63	52.36	77.82	71.36	75.92
KT-pFL	69.98	43.72	75.26	77.69	70.36	70.46	43.59	75.43	73.98	71.64
FedKD	68.91	44.86	77.92	79.36	74.36	72.98	48.13	76.87	75.63	76.26
ACE-pFL	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**

Models. For the image classification task, we employ four different model structures including GoogLeNet [35], VGG-16 [36], and ResNet-34/50 [37]. ACE-pFL has 20 clients assigned to them in four different model structures, i.e., five clients per model. Each client uses the pre-trained DistilBERT [38] language model for the text classification task. To solve the audio classification task, we use the mel-spectrogram instead of raw audio for training the models (i.e., VGG-19 [39] and ResNet-34 [37]). There are 10 clients per model.

Dataset Partitions. We introduce two dataset partitions, i.e., the Pathological partition $P(\pi)$ [8] and the Dirichlet partition $D(\beta, p)$ [40]. π represents the number of classes owned by each client in the Pathological partition. p is prior distribution of labels, and β measures the different distributions among clients in the Dirichlet partition. A smaller β value indicates a higher degree of non-IID data. They all adhere to the standard partitioning scheme for training, validation, and test datasets.

Parameter Settings. We set the number of clients to 20 and the number of clusters to 3. In addition we set the batch size B as 128, the training rounds R as 300, and the initial temperature T_0 as 30. The hyperparameter settings are as follows: $k_1 = 5$, $k_2 = 5$, $r_0 = 100$, $\alpha_1 = 0.4$, $\alpha_2 = 0.3$, $\alpha_3 = 0.2$, $\alpha_4 = 0.1$, $\eta_1 = 0.01$, and $\eta_2 = 0.015$.

Baselines. To validate the proposed ACE-pFL, we introduced the following FL schemes for comparison. FedAvg [8], FedProx [41], and SCAFFOLD [42] are traditional FL schemes, which are used as baseline benchmarks for comparison. Per-FedAvg [9], pFedMe [10], CD^2-pFed [11], and FedALA [12] are pFL schemes, but they only support homogeneous models. Therefore, we should ensure that all clients' models have the same architecture, that is, we employ ResNet-50 for the image classification task, use the pre-trained DistilBERT language model for the text classification task, and utilize ResNet-34 for the audio classification task. The comparison with these schemes aims to illustrate that even in the presence of model heterogeneity, our proposed scheme can achieve relatively superior results compared with homogeneous model deployment schemes. FedDF [13], FedAD [14], KT-pFL [15], and FedKD [16] are pFL schemes that can support heterogeneous models. The heterogeneous model deployment settings are as described above.

B. RQ1: Overall Results

We compare the testing accuracy, training time, and communication overhead of ACE-pFL with traditional FL schemes and other pFL schemes. We consider two scenarios: $\beta = 0.1$ and $\pi = 2$, which are two extreme examples of non-IID.

Testing Accuracy. Table I shows the testing accuracy of different schemes in two non-IID settings. We observe that ACE-pFL outperforms 2.97%–56.93% of all the baselines. The reasons are three fold. First, due to the poor ability of the personalized requirements, FedAvg, FedProx, and SCAFFOLD perform poorly in different non-IID settings. Second, the personalized schemes which only support homogeneous models may not well capture the needs of an individual client, so the testing accuracy is still relatively low. Although Per-FedAvg and FedALA proposed their methods that could adaptively aggregate the global model and local model, it still could not extract information effectively. pFedMe used the proximal term to extract the desired information from the local model and learned additional personalized models. However, learning the personalized models with the proximal term is an implicit way to extract the desired information. Third, the KD-based pFL schemes result in slightly lower testing accuracy than ACE-pFL due to the limited knowledge source and the relatively fixed distillation process (i.e., fixed distillation temperature). ACE-pFL improves the testing accuracy in pFL because it can better satisfy the individual needs.

Training Time and Communication Overhead. Table II shows the training time and communication overhead of different schemes in two non-IID settings. We find that the training time is 7.78× lower than other schemes at most, and the communication overhead is about 184× lower at most. Furthermore, we compare ACE-pFL with other baseline methods in terms of time and communication overhead under two non-IID settings and across five datasets, and calculate

TABLE II

THE TRAINING TIME (MIN) AND COMMUNICATION OVERHEAD (GB) OF DIFFERENT SCHEMES IN TWO NON-IID SETTINGS (I.E., $\beta = 0.1$ AND $\pi = 2$) ON FIVE DATASETS (I.E., CIFAR-10, CIFAR-100, CHEXPERT, REUTERS-21578, AND SPEECH COMMANDS)

Non-IID	$\beta = 0.1$ (Time and Communication ↓)					$\pi = 2$ (Time and Communication ↓)				
Schemes	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
FedAvg	394.40 115.23	551.16 266.37	1052.16 1148.43	1754.43 1724.06	888.65 1071.28	385.77 106.48	406.09 283.02	916.96 1014.45	1822.91 1637.85	784.88 1019.77
FedProx	390.51 102.43	485.02 226.41	946.89 1045.07	1723.69 1658.78	827.71 1023.93	383.43 102.07	360.96 249.72	843.23 957.03	1780.52 1564.58	763.23 917.77
SCAFFOLD	383.65 89.63	440.93 249.72	901.83 1083.35	1675.45 1586.13	787.09 995.74	372.45 94.51	381.26 224.75	798.33 926.40	1695.73 1508.55	738.87 933.93
Per-FedAvg	258.23 96.36	448.56 233.07	808.64 953.20	1635.97 1547.34	700.76 947.67	297.51 85.54	351.93 216.42	753.44 872.81	1653.34 1422.35	676.62 858.39
pFedMe	1241.64 87.36	1536.97 224.75	1751.53 987.65	2530.71 1478.38	1520.49 865.26	1287.26 79.38	1333.89 191.45	1714.97 815.39	1577.03 1301.66	1603.54 789.72
CD^2-pFed	224.38 80.32	363.77 183.13	727.45 865.15	1473.69 1405.11	502.72 765.69	269.68 72.45	306.81 166.48	679.77 757.96	1538.87 1262.84	671.22 686.71
FedALA	189.86 64.16	335.10 166.94	673.44 765.62	1434.22 1293.04	573.81 688.78	251.48 63.78	284.25 133.18	737.41 689.06	1483.76 1206.84	638.73 618.04
FedDF	216.63 2.81	352.74 6.84	901.83 9.35	1754.42 25.38	599.20 14.25	198.54 2.89	338.43 5.12	641.23 7.87	1271.86 21.37	557.59 7.28
FedAD	176.51 1.46	291.01 3.87	691.38 7.23	1662.29 22.44	479.87 13.11	231.63 1.49	270.72 4.89	702.35 6.23	1314.19 20.70	568.36 7.83
KT-pFL	240.72 2.70	308.65 6.23	751.59 7.79	1535.19 23.37	558.58 12.46	215.08 2.26	293.28 5.34	673.29 6.54	1356.58 22.04	622.45 8.19
FedKD	208.60 **0.08**	264.56 **2.00**	661.32 **3.11**	1403.52 **6.67**	507.83 **4.52**	181.99 **0.08**	225.69 **2.65**	609.17 **3.75**	1229.46 **8.36**	514.23 **5.73**
ACE-pFL	**160.46** 1.89	**220.46** 4.47	**601.23** 6.23	**1315.81** 20.03	**450.93** 10.68	**165.45** 0.97	**180.48** 3.56	**577.11** 5.61	**1187.01** 18.73	**487.17** 6.41

the overall average improvement. We also find that ACE-pFL can reduce the training time by 57% and the communication overhead by 59.12× on average. There are two main reasons. On one hand, the model convergence speed is accelerated, which leads to a reduction in the total number of training rounds of the personalized model, and therefore the total training time is reduced. On the other hand, transmitting soft predictions can significantly reduce communication overhead compared to exchanging a large number of high-dimensional model updates. This reduction is more pronounced as the number of training rounds increases.

C. RQ2: Effectiveness of EMD-Based Client Clustering

We validate the effectiveness of the proposed EMD-based client clustering in different non-IID scenarios. Specifically, we consider six non-IID data situations (i.e., $\beta = 0.1, 1.0, 10$ and $\pi = 2, 5, 7$), and compare ACE-pFL with other schemes on CIFAR-10 in terms of testing accuracy, training time, and communication overhead.

Table III presents the testing accuracy in different non-IID settings, while Table IV displays the training time and communication overhead. We have two observations. First, ACE-pFL outperforms other schemes in both testing accuracy and training time with lower communication overhead.

TABLE III

THE TESTING ACCURACY (%) OF DIFFERENT SCHEMES IN SIX NON-IID SETTINGS (I.E., $\beta = 0.1, 1.0, 10$ AND $\pi = 2, 5, 7$) ON CIFAR-10

Non-IID	$\beta = 0.1$	$\beta = 1.0$	$\beta = 10$	$\pi = 2$	$\pi = 5$	$\pi = 7$
FedAvg	24.32	40.97	70.93	33.63	48.97	69.62
FedProx	38.22	43.58	81.56	35.73	51.78	79.36
SCAFFOLD	49.93	55.37	79.36	52.79	63.93	75.39
Per-FedAvg	50.59	57.39	83.32	53.95	59.94	83.69
pFedMe	64.98	68.92	84.79	68.32	69.34	83.77
CD^2-pFed	62.84	67.19	82.79	65.98	66.39	81.34
FedALA	73.64	75.32	86.92	78.63	74.36	85.66
FedDF	70.29	73.69	84.36	73.98	75.92	83.12
FedAD	73.11	79.03	86.79	75.63	79.36	84.54
KT-pFL	69.98	74.98	83.39	70.46	78.97	80.98
FedKD	68.91	73.29	85.94	73.98	77.36	82.95
ACE-pFL	**81.25**	**83.96**	**88.36**	**79.96**	**82.39**	**89.73**

Second, ACE-pFL exhibits a maximum decrease in testing accuracy of 12.67% as the non-IID becomes more severe. As β (or π) decreases, other schemes experience an average testing accuracy decline of 23.94% (or 18.84%). We believe this is attributed to the EMD-based client clustering, which groups clients with similar data distributions. This enables each cluster of clients to focus more on specific data during

TABLE IV

THE TRAINING TIME (MIN) AND COMMUNICATION OVERHEAD (GB) OF DIFFERENT SCHEMES IN SIX NON-IID SETTINGS (I.E., $\beta = 0.1, 1.0, 10$ AND $\pi = 2, 5, 7$) ON CIFAR-10

Non-IID	$\beta = 0.1$	$\beta = 1.0$	$\beta = 10$	$\pi = 2$	$\pi = 5$	$\pi = 7$
FedAvg	394.40 115.23	346.08 96.79	248.22 85.48	385.77 106.48	326.75 95.72	275.18 78.79
FedProx	390.51 102.43	321.44 86.42	235.23 75.79	383.43 102.07	328.63 87.78	268.13 75.58
SCAFFOLD	383.65 89.63	314.06 75.28	241.29 66.32	372.45 94.51	317.69 82.78	254.39 69.93
Per-FedAvg	258.23 96.36	213.24 79.98	162.54 71.63	297.51 85.54	252.45 73.64	201.97 63.96
pFedMe	1241.64 87.36	1017.62 72.58	781.83 54.46	1287.26 79.38	1093.95 68.68	900.78 58.72
CD^2-pFed	224.38 80.32	183.68 66.65	141.12 59.68	269.68 72.45	228.68 62.37	188.32 56.26
FedALA	189.86 64.16	154.98 53.28	139.07 47.84	251.48 63.78	213.35 54.85	175.73 47.97
FedDF	216.63 2.81	179.28 2.33	127.71 2.09	198.54 2.89	168.34 2.48	138.69 2.13
FedAD	176.51 1.46	147.39 1.18	118.96 1.08	231.63 1.49	196.35 1.28	161.73 1.12
KT-pFL	240.72 2.70	195.36 2.41	155.61 1.98	215.08 2.26	182.75 1.94	150.36 1.67
FedKD	208.60 **0.08**	188.66 **0.08**	131.04 **0.08**	181.99 **0.08**	153.85 **0.08**	126.73 **0.08**
ACE-pFL	**160.46** 1.89	**132.39** 1.51	**103.89** 1.14	**165.45** 0.97	**140.25** 0.85	**115.52** 0.69

TABLE V

THE TESTING ACCURACY (%), COMPUTATION (S), AND COMMUNICATION OVERHEAD (MB) USING A SIMILARITY MATRIX OR A SIMILARITY VECTOR FOR EMD-BASED CLIENT CLUSTERING IN SIX NON-IID SETTINGS ON CIFAR-10

	$\beta = 0.1$	$\beta = 1.0$	$\beta = 10$
Matrix	82.69 99.36/23.42	85.16 100.13/23.40	88.96 98.43/23.41
Vector	81.25 7.36/1.16	83.96 8.26/1.15	88.36 8.02/1.16
	$\pi = 2$	$\pi = 5$	$\pi = 7$
Matrix	81.36 101.39/23.43	83.75 99.63/23.42	90.96 99.77/23.41
Vector	79.96 7.93/1.16	82.39 7.43/1.15	89.73 7.63/1.16

local training, thereby contributing to the enhancement of model performance. The accelerated convergence results in a decrease in overall training time and communication overhead.

We compare the testing accuracy, time, and communication overhead of ACE-pFL using a similarity matrix and a similarity vector for EMD-based client clustering. According to the results in Table V, we can observe that the accuracy loss is only about 2%, but the computation overhead is reduced by 13 $\times$ on average and the communication overhead is reduced by 20 $\times$ on average. Since z is randomly selected, it may impact the stability of system performance. To validate this, we conducted experiments with 5, 10, and 20 repetitions, using different client z. Table VI presents the averages of testing accuracy, training time, and communication overhead across different repetition counts. The results indicate that the system performance is stable, with no significant variations in testing accuracy, training time, or communication overhead, demonstrating the robustness of ACE-pFL.

TABLE VI

THE TESTING ACCURACY (%), TRAINING TIME (MIN), AND COMMUNICATION OVERHEAD (GB) FOR DIFFERENT SELECTIONS OF z FOR EMD-BASED CLIENT CLUSTERING ON CIFAR-10. THE CONTENTS ARE THE AVERAGE RESULTS OF RANDOMLY SELECTING z 5, 10, OR 20 TIMES

Times of Repetition	$\beta = 0.1$	$\beta = 1.0$	$\beta = 10$
5	81.93 162.06/1.91	83.45 133.09/1.54	88.45 104.09/1.15
10	82.69 163.63/1.92	83.74 134.39/1.63	87.36 106.95/1.23
20	81.43 159.36/1.85	82.96 130.96/1.48	89.03 102.66/1.11
Times of Repetition	$\pi = 2$	$\pi = 5$	$\pi = 7$
5	79.62 166.06/0.97	82.93 142.25/0.87	88.93 116.25/0.70
10	79.42 167.39/1.02	81.43 143.93/0.92	88.73 113.26/0.67
20	78.63 166.48/0.99	83.46 139.42/0.83	89.03 118.42/0.73

D. RQ3: Effect of Hyperparameters

We evaluate the impact of hyperparameters on ACE-pFL. The considered hyperparameters include the number of clusters $|V|$, initial temperature T_0, Top-K filtering strategy, and the weights of loss terms $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. We consider two scenarios: $\beta = 0.1$ and $\pi = 2$.

Effect of the Number of Clusters $|V|$. We investigate how different the number of clusters $|V|$ affects the performance of ACE-pFL. We set the number of clients N to 20 and consider different $|V|$, namely 2, 3, and 5. The results in Table VII and VIII show the testing accuracy, training time, and communication overhead in different $|V|$ settings. We observe that the performance of ACE-pFL is optimal when the number of clusters is 3. When the number of clusters is too small, the data within each cluster may be overly diverse. Conversely, when the number of clusters is too large, it may lead to a loss of crucial data features. Both scenarios can result in a decline in local model performance, affecting global aggregation.

We display the results of EMD-based client clustering in different $|V|$ settings in Figure 3. To illustrate that we cluster clients with similar data distributions, in each cluster we calculate the EMD between the data distributions of every pair of clients and take the average as the EMD value of the cluster. We have two findings. First, the small values of EMD for each cluster (i.e., EMD close to 0) indicate that the data distributions within each cluster are nearly an IID scenario. Second, if the

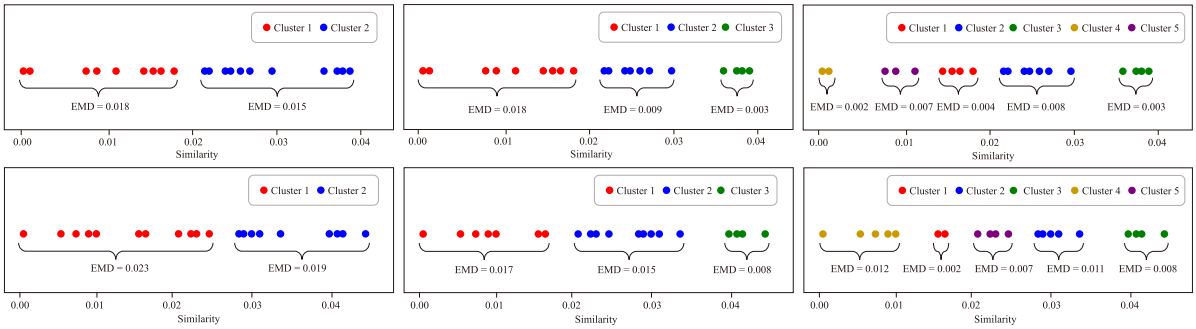


Fig. 3. Schematic of the clustering of ACE-pFL. The first row is $\beta = 0.1$, the second row is $\pi = 2$. Each point represents a client. Points with the same color indicate that the clients are grouped into a cluster.

TABLE VII

THE TESTING ACCURACY (%) OF ACE-PFL IN TWO NON-IID SETTINGS (I.E., $\beta = 0.1$ AND $\pi = 2$) ON FIVE DATASETS (I.E., CIFAR-10, CIFAR-100, CHEXPERT, REUTERS-21578, AND SPEECH COMMANDS) IN THE $|V| = 2, 3, 5$

	$\|V\|$	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	2	77.52	57.96	74.69	75.92	72.16
	3	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**
	5	78.93	61.92	78.29	79.92	77.89
$\pi = 2$	2	72.29	57.38	76.32	73.29	74.39
	3	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**
	5	78.39	64.09	79.41	75.39	78.91

TABLE VIII

THE TRAINING TIME (MIN) AND COMMUNICATION OVERHEAD (GB) OF ACE-PFL IN TWO NON-IID SETTINGS (I.E., $\beta = 0.1$ AND $\pi = 2$) ON FIVE DATASETS (I.E., CIFAR-10, CIFAR-100, CHEXPERT, REUTERS-21578, AND SPEECH COMMANDS) IN THE $|V| = 2, 3, 5$

	$\|V\|$	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	2	180.32 2.06	260.38 5.03	712.95 7.06	1723.69 27.36	580.78 14.25
	3	**160.46** **1.89**	**220.46** **4.47**	**601.23** **6.23**	**1315.81** **20.03**	**450.93** **10.68**
	5	170.28 1.93	230.25 4.87	650.36 6.75	1436.26 23.87	503.98 12.72
$\pi = 2$	2	188.98 2.13	278.93 5.24	743.98 7.25	1845.98 28.96	588.54 11.39
	3	**165.45** **0.97**	**180.48** **3.56**	**577.11** **5.61**	**1187.01** **18.73**	**487.17** **6.41**
	5	170.49 0.99	190.36 3.98	678.29 6.88	1365.98 22.69	539.74 10.49

number of clusters is too large, the number of clients within each cluster becomes too small, it may hinder the improvement in local model performance, which aligns with the previous results. Therefore, considering both the differences in data distribution within clusters and the number of clients within each cluster, it is essential to carefully select the number of clusters to ensure the optimal performance.

Effect of Initial Temperature T_0. We explore how different initial temperature T_0 affects the testing accuracy of ACE-pFL. In KD, the use of temperature helps soften the probability outputs, directing the attention of the student model towards

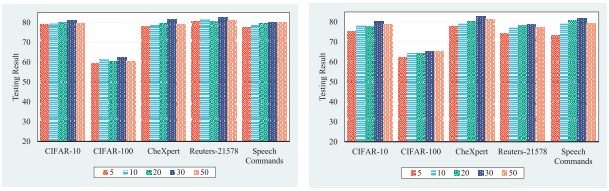


Fig. 4. The testing accuracy (%) of ACE-pFL in two non-IID settings (i.e., $\beta = 0.1$ (the left) and $\pi = 2$ (the right)) on five datasets (i.e., CIFAR-10, CIFAR-100, CheXpert, Reuters-21578, and Speech Commands) in the $T_0 = 5, 10, 20, 30, 50$.

logits with lower magnitudes. We set the initial temperatures T_0 to be 5, 10, 20, 30, and 50. Figure 4 shows the testing accuracy in different T_0 settings on five benchmark datasets. We observe that as the initial temperature increases, the testing accuracy initially rises and then decreases. ACE-pFL achieves the highest testing accuracy when the initial temperature is 30. This is because when the initial temperature is too low, the soft prediction distributions may become too sharp, making it challenging for the local model to capture some complex features in the global information. On the other hand, when the initial temperature is too high, the soft prediction distributions may be too smooth, leading to losing the ability to capture the true data distribution. An appropriate initial temperature strikes a balance between a smooth soft prediction distributions and capturing the features of the true data distribution, ultimately achieving the highest testing accuracy. Selecting an appropriate initial temperature ensures that ACE-pFL performs optimally.

Effect of Top-K Filtering Strategy. We investigate how different Top-K filtering strategy affects the performance of ACE-pFL. Table IX shows the testing accuracy, training time, and communication overhead in different Top-K filtering strategy settings on five datasets. We find that as K increases, the testing accuracy gradually rises, but communication overhead also increases, with little difference in training time. This is because as K increases, more information from the retained soft prediction distributions aids the model in extracting more crucial details, contributing to improved accuracy but at the cost of increased communication overhead. For example, the testing accuracy with $K = 5$ only decreases by an average of 1.78% compared to $K = 10$ across two data distribution scenarios on CIFAR-10, which is within an acceptable range. Additionally, the training time and communication overhead for $K = 5$ are lower. Therefore, the choice of the Top-K filtering strategy should consider both time and

TABLE IX

THE TESTING ACCURACY (%), TRAINING TIME (MIN), AND COMMUNICATION OVERHEAD (GB) OF ACE-PFL IN TWO NON-IID SETTINGS (I.E., $\beta = 0.1$ AND $\pi = 2$) ON FIVE DATASETS (I.E., CIFAR-10, CIFAR-100, CHEXPERT, REUTERS-21578, AND SPEECH COMMANDS) IN THE DIFFERENT K

CIFAR-10	K = 3	K = 5	K = 7	K = 10
$\beta = 0.1$	78.47 182.93/1.14	81.25 160.46/1.89	82.74 170.90/2.74	83.01 166.59/3.89
$\pi = 2$	76.56 184.80/0.82	79.96 165.45/0.97	80.41 192.73/1.58	81.79 171.68/1.99
CIFAR-100	K = 10	K = 30	K = 50	K = 100
$\beta = 0.1$	62.53 220.46/4.47	63.42 221.08/13.59	64.74 236.29/23.52	64.16 233.95/47.36
$\pi = 2$	65.39 180.48/3.56	67.51 188.30/11.69	68.25 249.67/18.72	66.93 233.99/37.56
CheXpert	K = 3	K = 7	K = 10	K = 14
$\beta = 0.1$	77.49 632.01/3.83	81.76 601.23/6.23	80.49 639.30/8.22	82.34 632.36/13.62
$\pi = 2$	79.32 585.47/2.85	82.48 577.11/5.61	84.02 578.86/8.28	82.73 585.70/12.16
Reuters-21578	K = 5	K = 15	K = 50	K = 90
$\beta = 0.1$	80.41 1337.48/6.97	82.33 1315.81/20.03	84.02 1331.56/67.98	83.47 1335.18/128.12
$\pi = 2$	75.26 1198.67/6.33	78.92 1187.01/18.73	79.34 1415.53/64.73	77.75 1495.29/138.12
Speech Commands	K = 3	K = 5	K = 8	K = 10
$\beta = 0.1$	77.59 472.52/4.51	78.03 472.47/6.75	80.13 450.93/10.68	82.36 483.04/14.53
$\pi = 2$	75.82 497.71/2.75	77.92 531.32/4.62	81.59 487.17/6.41	80.47 547.31/8.94

communication overhead, as well as potential testing accuracy loss.

Effect of the Weights of Loss Terms. We investigate how different weights of loss terms affect the testing accuracy of ACE-pFL. We consider equal weights and unequal weights. For unequal weights, given that the cross-entropy loss (i.e., $\mathcal{L}_{CE}$) is crucial for model training, we fix its weight at 0.4 and focus on exploring the contributions of the remaining three knowledge terms to the testing accuracy of ACE-pFL. Table X presents the testing accuracy on five datasets in two non-IID settings with various weights. We have two observations. First, the testing accuracy under unequal weights is consistently higher than that under equal weights, indicating that although providing multiple sources of knowledge contributes to enhancing KD, the contributions of these knowledge sources are not equal. Second, the highest accuracy is achieved under unequal weight ②, indicating the relative importance of the three knowledge terms (i.e., local-to-global > local-to-local > global-to-global). Therefore, it is crucial to appropriately set the weights for each loss term.

TABLE X

THE TESTING ACCURACY (%) OF ACE-PFL IN THE $\beta = 0.1$ AND $\pi = 2$ ON FIVE DATASETS.① DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.25, 0.25, 0.25, 0.25$; ② DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.4, 0.3, 0.2, 0.1$;③ DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.4, 0.3, 0.1, 0.2$;④ DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.4, 0.2, 0.3, 0.1$;⑤ DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.4, 0.2, 0.1, 0.3$;⑥ DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.4, 0.1, 0.2, 0.3$;⑦ DENOTES $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 0.4, 0.1, 0.3, 0.2$

	Case	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	①	72.88	54.21	67.41	73.14	66.53
	②	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**
	③	80.40	60.15	72.30	81.32	77.17
	④	80.15	59.79	75.22	82.31	69.66
	⑤	78.77	54.97	67.41	79.04	73.03
	⑥	73.13	59.44	71.48	79.27	67.73
	⑦	75.89	54.97	70.90	78.38	67.12
$\pi = 2$	①	68.98	53.08	66.02	68.34	69.91
	②	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**
	③	79.46	58.74	73.05	77.67	77.06
	④	75.40	57.13	72.34	75.02	71.26
	⑤	73.45	56.96	71.68	74.15	72.66
	⑥	72.73	54.02	68.70	73.59	70.86
	⑦	74.18	55.88	66.02	75.81	71.04

TABLE XI

THE TESTING ACCURACY (%) OF ACE-PFL IN THE $\beta = 0.1$ AND $\pi = 2$ ON FIVE DATASETS."W/O" DENOTES THE ABSENCE OF EMD-BASED CLIENT CLUSTERING."W/" DENOTES THE PRESENCE OF EMD-BASED CLIENT CLUSTERING

		CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	w/o	75.19	53.49	74.29	70.98	73.91
	w/	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**
$\pi = 2$	w/o	71.36	49.63	73.98	63.21	70.32
	w/	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**

E. RQ4: Ablation Experiments

We conduct ablation experiments to examine the effectiveness of our proposed EMD-based client clustering, dynamic distillation temperature adjustment, and the triple distillation strategy. We evaluate the testing accuracy of ACE-pFL on five datasets in two non-IID data situations (i.e., $\beta = 0.1$ and $\pi = 2$).

EMD-based Client Clustering. We examine the effectiveness of EMD-based client clustering. Table XI displays the testing accuracy of ablation experiments with EMD-based client clustering in the $\beta = 0.1$ and $\pi = 2$ on five datasets. We find that using EMD-based client clustering effectively enhances testing accuracy. For instance, when $\beta = 0.1$, the testing accuracy on CIFAR-10 increases by 6.06%; when $\pi = 2$, the testing accuracy on CIFAR-10 increases by 8.60%. This improvement is attributed to the grouping of clients with similar data distributions, which allows clients within each cluster to capture key features of the data more effectively. Therefore, EMD-based client clustering is essential for enhancing the performance of ACE-pFL.

Dynamic Distillation Temperature Adjustment. We examine the effectiveness of dynamic distillation temperature

TABLE XII
THE TESTING ACCURACY (%) OF ACE-PFL IN THE $\beta = 0.1$ AND $\pi = 2$ ON FIVE DATASETS."FIX." DENOTES A FIXED DISTILLATION TEMPERATURE."DYNA." DENOTES A DYNAMIC DISTILLATION TEMPERATURE

		CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	Fix.	75.68	58.63	74.26	78.96	75.62
	Dyna.	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**
$\pi = 2$	Fix	72.84	60.89	76.29	73.48	77.81
	Dyna.	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**

adjustment. Table XII presents the testing accuracy of ablation experiments with dynamic distillation temperature in the $\beta = 0.1$ and $\pi = 2$ on five datasets. We observe that using dynamic distillation temperature adjustment effectively enhances testing accuracy. For example, when $\beta = 0.1$, the testing accuracy on CIFAR-10 increases by 5.57%; when $\pi = 2$, the testing accuracy on CIFAR-10 increases by 7.12%. This improvement is due to the influence of temperature on the extent of soft prediction distributions. Gradually increasing the temperature as the model converges makes the distribution smoother, allowing the model to focus on more information and, consequently, improving model performance. Therefore, dynamic distillation temperature adjustment is crucial for enhancing the effectiveness of KD.

Dynamic distillation temperature adjustment can be done through linear and nonlinear functions. We set the initial temperature $T_0 = 30$ and the training rounds $r = 300$. We present five different functions in Figure 5. $f_5(x)$ represents a linear function to adjust temperature, while $f_1(x)$ to $f_4(x)$ represent nonlinear functions. $f_1(x)$ and $f_4(x)$ are more complex nonlinear functions, featuring segmented linear parts (i.e., $\frac{k_1}{1+e^{-k_2}}\lfloor\frac{r}{r_0}\rfloor$ and $\frac{1+e^{k_1}}{k_2}\lfloor\frac{r}{r_0}\rfloor$) and sigmoid-like parts (i.e., $\frac{2k_1}{1+e^{-k_2\Delta r}}$ and $\frac{1+e^{k_1\Delta r}}{k_2}$), resulting in different temperature adjustment behaviors during various training stages. The difference lies in the gradual slowdown in the growth rate of $f_1(x)$, while the growth rate of $f_4(x)$ gradually accelerates. $f_2(x)$ and $f_3(x)$ are simpler sigmoid-like nonlinear functions. The difference lies in the gradual slowdown in the growth rate of $f_2(x)$, while the growth rate of $f_3(x)$ gradually accelerates.

We obtain the testing accuracy of these five temperature adjustment curves on five datasets and observe that using the $f_1(x)$ temperature adjustment curve yields the highest testing accuracy, as shown in Table XIII. This is mainly due to the following three reasons. First, gradually increasing the temperature during different training stages guides the model to focus more on global knowledge. Second, higher temperatures make soft prediction distributions smoother, facilitating more exploration in the parameter space, which allows the model to better capture global features of the data. Third, gradually increasing the distillation temperature at a slower pace helps avoid drastic fluctuations during the training process.

The Triple Distillation Strategy. We examine the effectiveness of triple distillation strategy. Table XIV displays the testing accuracy of ablation experiments with the triple knowledge distillation strategy in the $\beta = 0.1$ and $\pi = 2$ on five datasets. We find that the triple distillation strategy achieves the highest testing accuracy. For instance, when $\beta = 0.1$, the testing accuracy on CIFAR-10 increases by 11.09% and 6.57% compared to the other two strategies; when $\pi = 2$, the testing accuracy on CIFAR-10 increases by 11.87% and 7.04%. This is because the triple distillation strategy provides a richer and more diverse source of knowledge, contributing to an enhanced the effect of KD.

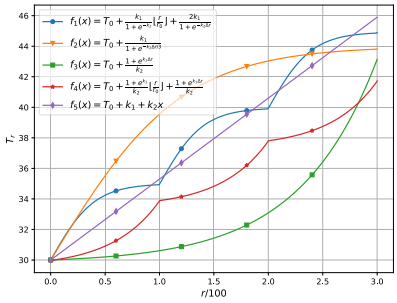


Fig. 5. Dynamic temperature adjustment curve.

TABLE XIII
THE TESTING ACCURACY (%) OF ACE-PFL IN THE $\beta = 0.1$ AND $\pi = 2$ ON FIVE DATASETS. $f_i(x)$ DENOTES DYNAMIC TEMPERATURE ADJUSTMENT CURVE, WHERE $i \in \{1, 2, 3, 4, 5\}$

	$f_i(x)$	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	1	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**
	2	78.36	60.72	78.36	80.94	79.36
	3	76.45	55.63	75.36	74.42	75.11
	4	77.14	58.63	76.41	76.33	76.32
	5	74.91	52.78	72.47	73.44	71.28
$\pi = 2$	1	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**
	2	77.78	59.74	77.26	79.32	78.96
	3	75.02	55.79	73.28	74.01	73.19
	4	76.48	57.46	75.87	75.99	75.44
	5	73.45	51.84	70.33	70.42	70.22

TABLE XIV
THE TESTING ACCURACY (%) OF ACE-PFL IN THE $\beta = 0.1$ AND $\pi = 2$ ON FIVE DATASETS."SINGLE" DENOTES SINGLE DISTILLATION STRATEGY (I.E., LOCAL-TO-GLOBAL DISTILLATION)."DOUBLE" DENOTES DOUBLE DISTILLATION STRATEGY (I.E., LOCAL-TO-GLOBAL DISTILLATION, AND LOCAL-TO-LOCAL DISTILLATION)."TRIPLE" DENOTES TRIPLE DISTILLATION STRATEGY (I.E., LOCAL-TO-GLOBAL DISTILLATION, LOCAL-TO-LOCAL DISTILLATION, AND GLOBAL-TO-GLOBAL DISTILLATION)

	Strategy	CIFAR-10	CIFAR-100	CheXpert	Reuters-21578	Speech Commands
$\beta = 0.1$	Single	70.16	54.36	73.69	70.98	68.32
	Double	74.68	58.39	77.29	74.26	77.92
	Triple	**81.25**	**62.53**	**81.76**	**82.33**	**80.13**
$\pi = 2$	Single	68.09	53.19	70.18	66.48	70.41
	Double	72.92	59.26	77.16	73.19	76.49
	Triple	**79.96**	**65.39**	**82.48**	**78.92**	**81.59**

V. RELATED WORK

Personalized Federated Learning. FL [8] is a machine learning setup in which multiple clients collaborate to train a global model without exposing raw data locally. However,

it entails training a single global model, failing to meet the heterogeneous and personalized model requirements of individual clients. To address this issue, pFL aims to tailor models for each client. We consider the following four classes of pFL methods: hybrid model, meta-learning, regularization technique, and knowledge distillation.

The first approach to pFL is hybrid model, where clients learn a mixture of the global model and local models. FedPer [43] introduced a novel neural network architecture consisting of a base layer and a personalized layer. The server trains the base layer through FedAvg, while the personalized layer is locally trained to create individualized models. FedALA [12] designed an adaptive local aggregation module, which can adaptively aggregate the global model and local model towards the local objective on each client to initialize the local model before training in each iteration. The second approach to pFL is meta-learning. Per-FedAvg [9] was influenced by model-agnostic meta-learning and constructed an initial meta-model that can be effectively updated after a single gradient descent step. However, it is computationally expensive due to reliance on the Hessian matrix. The third approach to pFL is using different regularization terms. Reference [10] introduced pFedMe utilizing Moreau envelopes as clients' regularized loss functions, which helped decouple personalized model optimization from the global model learning for pFL.

KD-based pFL. Different from the above methods, we mainly focus on the knowledge distillation approaches because they are more convenient and effective. CD^2-pFed [11] introduced a novel cyclic distillation-guided channel decoupling framework for personalizing the global model under diverse non-IID settings. FedDF [13] proposed an ensemble distillation method for model fusion, specifically training a global model using unlabeled data on the outputs of the clients' models. FedAD [14] introduced a novel distillation algorithm that aggregates structural knowledge, addressing the inherent heterogeneity in FL while explicitly balancing local model diversity and consensus. KT-pFL [15] enhanced collaboration among clients with similar data distributions by adaptively reinforcing personalized soft predictions for each client through linear combinations of all local soft predictions. FedKD [16] employed a one-time offline knowledge distillation using a public dataset, developing a privacy-preserving and communication-efficient pFL framework. Regrettably, the above pFL schemes lead to more loss of accuracy due to inadequate knowledge extraction and complex learning mechanism, and have large training time and communication overhead. In this work, we propose ACE-pFL to improve the accuracy, and reduce the training time and communication overhead.

VI. Conclusion

In this paper, we propose a new accurate and efficient pFL framework based on KD, called ACE-pFL. Our proposed EMD-based client clustering addresses the adverse effect of non-IID data in pFL. We design the dynamic distillation temperature adjustment method to better adapt to the dynamic training process of pFL. We employ the triple distillation strategy to leverage not only the knowledge from aggregated global soft predictions but also the implicit knowledge from local and global perspectives. Experimental results demonstrate that ACE-pFL achieves a tripartite balance between testing accuracy, training time, and communication overhead.

References

[1] Y. Liu et al., "FedVision: An online visual object detection platform powered by federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 13172–13179.

[2] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[3] Z. Liu et al., "Contribution-aware federated learning for smart healthcare," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 12396–12404.

[4] J. Ogier du Terrail et al., "Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–20.

[5] S. A. E. M. Nasri, I. Ullah, and M. G. Madden, "Compression scenarios for federated learning in smart manufacturing," *Proc. Comput. Sci.*, vol. 217, pp. 436–445, Jan. 2023.

[6] W. Zellinger et al., "Beyond federated learning: On confidentiality-critical machine learning applications in industry," *Proc. Comput. Sci.*, vol. 180, pp. 734–743, Jan. 2021.

[7] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, "Federated multi-task learning under a mixture of distributions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–14.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[9] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[10] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.

[11] Y. Shen, Y. Zhou, and L. Yu, "CD2-pFed: Cyclic distillation-guided channel decoupling for model personalization in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10031–10040.

[12] J. Zhang et al., "Fedala: Adaptive local aggregation for personalized federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 11237–11244.

[13] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–13.

[14] X. Gong et al., "Ensemble attention distillation for privacy-preserving federated learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15056–15066.

[15] J. Zhang, S. Guo, X. Ma, H. Wang, W. Xu, and F. Wu, "Parameterized knowledge transfer for personalized federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–13.

[16] X. Gong et al., "Preserving privacy in federated learning with ensemble cross-domain knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11891–11899.

[17] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "FedDC: Federated learning with non-IID data via local drift decoupling and correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10102–10111.

[18] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-IID federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10164–10173.

[19] J. Shao, Y. Sun, S. Li, and J. Zhang, "Dres-fl: Dropout-resilient secure federated learning for non-iid clients via secret data sharing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–13.

[20] J. Li, X. Wu, W. Dong, S. Wu, C. Bian, and D. Xiong, "Swing distillation: A privacy-preserving knowledge distillation framework," 2022, *arXiv:2212.08349*.

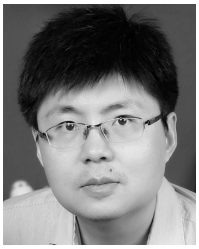
[21] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 5191–5198.

[22] Z. Li et al., "Curriculum temperature for knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 1504–1512.
[23] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[24] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4793–4801.
[25] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2599–2608.
[26] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, pp. 99–121, Nov. 2000.
[27] L. Caicedo Torres, L. Manella Pereira, and M. Hadi Amini, "A survey on optimal transport for machine learning: Theory and applications," 2021, *arXiv:2106.01963*.
[28] Z. Wen and Y. Li, "Toward understanding the feature learning process of self-supervised contrastive learning," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 11112–11122.
[29] J. Shao et al., "A survey of what to share in federated learning: Perspectives on model utility, privacy leakage, and communication efficiency," 2023, *arXiv:2307.10655*.
[30] J. Shao, F. Wu, and J. Zhang, "Selective knowledge sharing for privacy-preserving federated distillation without a good teacher," *Nature Commun.*, vol. 15, no. 1, p. 349, Jan. 2024.
[31] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.
[32] J. Irvin et al., "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 590–597.
[33] F. Debole and F. Sebastiani, "An analysis of the relative hardness of Reuters-21578 subsets," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 56, no. 6, pp. 584–596, Apr. 2005.
[34] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.
[35] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[36] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 169–175.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[38] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
[39] J. Jaworek-Korjakowska, P. Kleczek, and M. Gorgon, "Melanoma thickness prediction based on convolutional neural network with VGG-19 model transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2748–2756.
[40] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," 2020, *arXiv:2002.06440*.
[41] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 1–22.
[42] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1–12.
[43] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.

Kun He (Member, IEEE) received the Ph.D. degree in computer science from the Computer School, Wuhan University. He is currently an Associate Professor with Wuhan University. He has published more than 30 research papers in many international journals and conferences, such as IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON MOBILE COMPUTING, USENIX Security, CCS, and INFOCOM. His research interests include cryptography, network security, mobile computing, and cloud computing.

Hao Bai received the B.E. degree in information security from Wuhan University, Wuhan, China, in 2022, where he is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering. His research interests include privacy-preserving distributed machine learning.

Yuqing Li (Member, IEEE) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019. From 2019 to 2020, she was a Post-Doctoral Fellow with The Hong Kong University of Science and Technology, Hong Kong. From 2020 to 2022, she was a Researcher with the Huawei Hong Kong Research Center, Hong Kong. She is currently an Associate Professor with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. Her research interests include data privacy and security, distributed machine learning, and edge computing.



Jing Chen (Senior Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan. He has been a Full Professor with Wuhan University, since 2015. He has published more than 100 research papers in many international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, USENIX Security, CCS, and INFOCOM. His research interests include computer science, network security, and cloud security. He acts as a reviewer for many journals and conferences, such as IEEE TRANSACTIONS ON INFORMATION FORENSICS, IEEE TRANSACTIONS ON COMPUTERS, and IEEE/ACM TRANSACTIONS ON NETWORKING.

Ruiying Du received the B.S., M.S., Ph.D. degrees in computer science from Wuhan University, Wuhan, China, in 1987, 1994, and 2008, respectively. She is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. She has published more than 80 research papers in many international journals and conferences, such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *International Journal of Parallel and Distributed Systems*, INFOCOM, SECON, TrustCom, and NSS. Her research interests include network security, wireless networks, cloud computing, and mobile computing.