FedCross: Towards Accurate Federated Learning via Multi-Model Cross-Aggregation

Ming Hu[1], Peiheng Zhou[2], Zhihao Yue[2], Zhiwei Ling[2], Yihao Huang[1], Anran Li[1],
Yang Liu[1], Xiang Lian[3], Mingsong Chen[2]
[1]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[2]*MoE Engineering Research Center of SW/HW Co-Design Tech. and App., East China Normal University, China*
[3]*Department of Computer Science, Kent State University, Ohio, USA*

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00170

***Abstract*—As a promising distributed machine learning paradigm, Federated Learning (FL) has attracted increasing attention to deal with data silo problems without compromising user privacy. By adopting the classic one-to-multi training scheme (i.e., FedAvg), where the cloud server dispatches one single global model to multiple involved clients, conventional FL methods can achieve collaborative model training without data sharing. However, since only one global model cannot always accommodate all the incompatible convergence directions of local models, existing FL approaches greatly suffer from inferior classification accuracy. To address this issue, we present an efficient FL framework named *FedCross*, which uses a novel multi-to-multi FL training scheme based on our proposed *multi-model cross-aggregation* approach. Unlike traditional FL methods, in each round of FL training, FedCross uses multiple middleware models to conduct weighted fusion individually. Since the middleware models used by FedCross can quickly converge into the same flat valley in terms of loss landscapes, the generated global model can achieve a well-generalization. Experimental results on various well-known datasets show that, compared with state-of-the-art FL methods, FedCross can significantly improve FL accuracy within both IID and non-IID scenarios without causing additional communication overhead.**

***Index Terms*—Federated learning, gradient divergence, loss landscape, multi-model cross-aggregation, non-IID**

I. Introduction

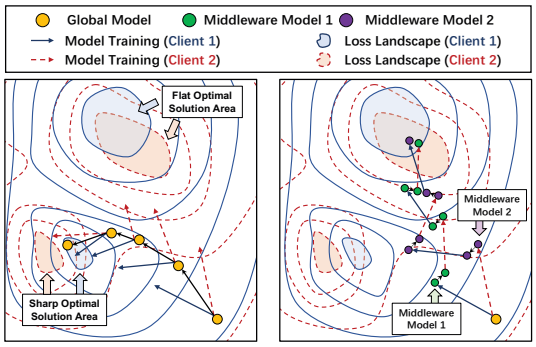
Along with the prosperity of Artificial Intelligence (AI) and Internet of Things (IoT) techniques, more and more Artificial Intelligence of Things (AIoT) applications [1] (e.g., autonomous driving [2], smart transportation [3], medical monitoring [4]) resort to Deep Neural Network (DNN) models to enable accurate sensing and intelligent control. Although such DNN models can deal with various complex tasks, due to the limited learning capabilities of IoT devices and stringent requirements for their data privacy, traditional centralized DNN training methods suffer a lot from the problem of low classification performance. Alternatively, to facilitate the design of large-scale AIoT applications, Federated Learning (FL) [5]–[9] has been used as a promising distributed machine learning-based infrastructure, which allows knowledge sharing among AIoT devices without compromising their privacy. Typically, FL adopts a cloud-client architecture, where the cloud server periodically updates the global model by aggregating the received local gradients and dispatching the updated global model to clients for a new round of training. Since none of the clients send their raw data to the cloud server, their privacy can be safely preserved.

Although FL is good at knowledge sharing among clients, it often fails to withstand low classification performance in deploying real-world applications, especially when client data are non-IID (Independent and Identically Distributed) [10]–[14]. This is mainly because most existing FL methods rely on the classic aggregation scheme, i.e., Federated Averaging (FedAvg) [5], where the cloud server only dispatches one single global model to selected clients in a one-to-multi manner. Since the raw data on clients are different, the optimization directions of local models will gradually become divergent during the training, resulting in conflicting gradients among local models. In this case, by simply averaging the collected gradients from all the selected clients, the knowledge and efforts of local models accumulated in previous rounds of FL training are inevitably eclipsed. Due to such notorious phenomenon of gradient divergence [15], [16], the classification capability of the global model is greatly limited. To alleviate the gradient divergence problem, various approaches have been investigated to guide the optimization directions of local training, striving to derive local models with fewer conflicting parameters. However, since such methods cannot prevent the knowledge learned by individual clients from being damaged by the coarse-grained aggregation strategy (i.e., FedAvg), the classification capability of the global model is still restricted.

According to [17]–[20], a well-generalized DNN training solution tends to be located in flat valleys rather than sharp ravines from the perspective of *loss landscapes* [17]. Inspired by this observation, designing an FL method to guide client model training towards a flatter valley to achieve a more generalized global model would be wise. As a motivating example, Figure 1(a) presents the loss landscapes of FedAvg involving two clients, where blue (solid) and red (dotted) contours indicate the loss landscapes of client 1 and client 2, respectively. Here, we assume that each client has two optimal solutions (i.e., the sharp and flat optimal solutions), where the blue and red shaded areas are for client 1 and client 2, respectively. Note that from the perspective of loss landscapes, a larger overlap exists between optimal solution areas if clients' data are more similar. Here, we use yellow circles to denote intermediate aggregated global models along the FL training process, where the black solid arrow lines form the optimization route of the global model. We can find that

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00170

the global model converges into the blue sharp solution area. In this case, the remaining FL training process will inevitably get stuck in this area due to the one-to-multi style aggregation. In this case, although the obtained global model works well for client 1, it is unsuitable for client 2, although the global model is located near (rather than in) the red-shaded area, resulting in an inferior global model with bad generalization.



(a) FedAvg (b) FedCross

Fig. 1. A motivating example of FedAvg and FedCross training.

Ideally, we can achieve a better global model for FL if local models can access the raw data of all the clients. However, this will violate the privacy-preserving requirement of FL, since both raw data and their distributions of clients are assumed to be private. Without such information, existing FedAvg-based FL methods can only tune the conflicting parameters by coarse-grained aggregations during the FL training, where the conflicting parameters of locally trained models are not properly treated. Apparently, *how to break through the limit of FedAvg to enable the fine-grained training of local models and wisely resolve the conflicting gradients to generate a well-generalized global model that performs well in all the clients with different data distributions* is becoming an urgent issue in FL design.

To address the challenge above, this paper presents a novel FL framework named FedCross based on our proposed multi-model cross-aggregation-based training scheme, where we adopt *middleware models* to simultaneously respect the local training of clients and increase the chance of accessing different clients' data. Figure 1 illustrates the basic idea of our FedCross approach, where the training process is gradually optimized towards the flat solution areas. In this example, two middleware models are trained by the two clients (i.e., green circles for middleware model 1 and purple circles for middleware model 2). Note that, during FedCross training, the middleware models are sufficiently trained on different devices, indicated by interleaved arrow lines with different colors. This way, the conflicting parameters of these two middleware models are gradually revised by continuous local training. Eventually, their optimization directions will converge towards the intersection of flat optimal solution areas.

Unlike one-to-multi style FedAvg, FedCross conducts the local training in a multi-to-multi manner, which uses multiple middleware models to resolve the conflicts among local models on the cloud server. Rather than eliminating the conflicts immediately through FedAvg-like coarse-grained aggregation, FedCross effectively solves them by consecutive local training on different clients. Specifically, in each training round of FedCross, the cloud server dispatches multiple homogeneous middleware models to the selected clients for local training. After receiving all the locally trained models, FedCross applies our multi-model cross-aggregation strategy, which updates each middleware model on the cloud server by aggregating it with its collaborative model trained on some selected client. With our multi-to-multi training scheme, each middleware model in FedCross is updated with data from different clients without privacy leaking. The conflicting weights of each middleware model can be revised by fine-grained local training rather than coarse-grained averaging aggregation. Thus, FedCross can generally achieve better classification performance than FedAvg-based FL methods. Due to the same set of host clients and our proposed cross-aggregation strategy that restricts the weight differences between middleware models, the trained middleware models will eventually become similar. Note that, at the end of FL training, FedCross only performs the federated averaging operation once on all the trained middleware models so as to form a unified "global" model to benefit all the clients.

This paper makes the following four major contributions:

- We establish a novel multi-to-multi FL framework named FedCross, which adopts only-for-training middleware models to generate a well-generalized global model.
- We design a multi-model cross-aggregation scheme, which supports the fine-grained training of local models to wisely resolve the conflicts among their parameters.
- We prove the convergence of FedCross and propose two optimization methods to accelerate the FedCross training.
- We conduct extensive experiments to evaluate the performance and pervasiveness of our FedCross approach.

The rest of this paper is organized as follows. Section II introduces the preliminaries and related works on FL. Section III presents the details of our proposed FedCross approach. Section IV empirically studies the performance of our FedCross approach, compared with state-of-the-art FL methods. Finally, Section V concludes the paper.

II. Preliminaries and Related Work

A. Preliminaries

Consider learning a predictive model that maps an input space X to an output space Y. Assume that there are two entities involved in an FL system: a cloud server S and N distributed clients with indices of $\{1, 2, \cdots, N\}$. Let each client i possess a local dataset $D_i = \{z_{i,1}, z_{i,2}, \cdots, z_{i,n_i}\}$, where $z_{i,j} = (x_{i,j}, y_{i,j}) \in X \times Y$. Under the coordination of the cloud server, all participant clients collaboratively train a global model $\hat{w}$ by sharing their local models trained on their private datasets. The goal of a standard FL optimization problem is formulated as follows:

$$\min_{w} F(w) = \frac{1}{N}\sum_{i=1}^{N} f_i(w), \ s.t., \ f_i(w) = \frac{1}{n_i}\sum_{j=1}^{n_i} l(z_{i,j}; w),$$

where l and f_i denote the loss functions of an individual sample (e.g., the cross-entropy loss) and all the samples of client i, respectively. F represents the loss function of the global model. The traditional one-to-multi FL system solves this problem based on iterative stochastic optimization, where each training iteration t involves four major steps: i) model dispatching, where the cloud server selects a subset of clients and dispatches the current model w_t to them; ii) local updating, where each selected client i independently trains a local model based on $w_{t+1}^i = w_t^i - \eta \nabla F(w_t)$; iii) model uploading, where each client i uploads the updated local model w_{t+1}^i to the cloud server; and iv) model aggregation, where the server aggregates all the received models and conducts the model aggregation to obtain a new global model w_{t+1} by FedAvg [5].

B. Related Work on FL Optimization

To support efficient FL in the design of AIoT applications, various framework- and workflow-level optimization techniques have been extensively studied, including cloud-client collaboration [7], [21]–[23], resource allocation and task scheduling [24]–[27], heterogeneity management [28]–[32], fault tolerance [9], [33]–[35], and personalized service [36]. Although these methods are promising, they can only deal with specific AIoT scenarios. So far, to improve the classification performance of general-purpose FL methods, especially for non-IID scenarios, existing optimization methods for FL can be mainly classified into the following three categories.

The *global control variable-based methods* [37], [38] attempt to use a global variable to guide the training direction of local training, thereby alleviating gradient divergence. For example, SCAFFOLD [37] dispatches global control variables to clients to correct the "client-drift" problem in the local training process. FedProx [39] regularizes local loss functions with a proximal term to stabilize the model convergence, where such a proximal term is the squared distance between local and global models. The *client grouping-based methods* [40], [41] group clients based on the similarity of their data distributions and select clients to participate in FL training by group. Since it is hard to directly obtain the data distributions of clients, most existing methods conduct the client grouping only based on simple information such as model similarity. For example, FedCluster [40] groups the clients into multiple clusters that perform federated learning cyclically in each learning round. CluSamp [41] uses either the sample size or model similarity to group clients, which can reduce the variance of client stochastic aggregation parameters in FL. Unlike the former two categories, the *Knowledge Distillation (KD)-based methods* [42]–[44] adopt a "teacher model" to guide the training of "student models". Specifically, the "student models" use soft labels of the teacher model to perform model training, thus learning the knowledge of the teacher model. For example, FedAUX [45] performs data-dependent distillation by using an auxiliary dataset to initialize the server model. FedGen [46] performs data-free distillation and leverages a proxy dataset to address the heterogeneous FL problem using a built-in generator. FedDF [43] uses ensemble distillation to accelerate FL by training the global model through unlabeled data on the outputs of local models.

Although various optimization methods have been proposed to improve FL performance, due to the usage of the same global models for local training, most of them suffer from the problem of getting stuck in sharp ravines during the exploration of loss landscapes. As an alternative, our FedCross approach adopts multiple intermediate models for local training. In this case, intermediate models can quickly escape from sharp ravines based on our proposed cross-aggregation mechanism. To the best of our knowledge, FedCross is the first attempt that uses a novel multi-to-multi training scheme based on our proposed multi-model cross-aggregation. By using a more fine-grained FL training strategy, FedCross fully respects the convergence characteristics of clients during the training, thus achieving much better classification performance than state-of-the-art FL methods.

III. Our FedCross Approach

A. Overview of FedCross

The architecture of FedCross consists of a central cloud server and multiple local devices, which is the same as conventional one-to-multi FL frameworks. The main difference is that FedCross uses a multi-to-multi training and aggregation mechanism. Specifically, FedCross uses multiple homogeneous middleware models for local training and updates these middleware models with a cross-aggregation strategy. FedCross still generates a global model, but this global model is only for deployment rather than model training.

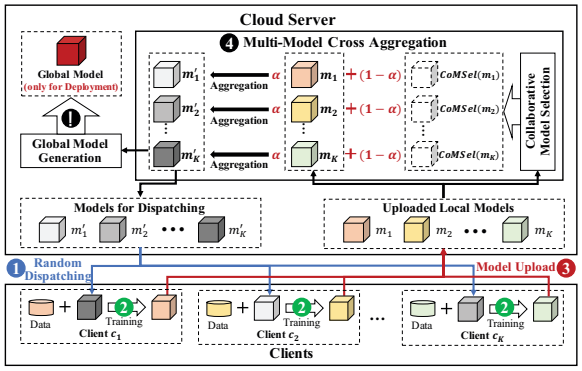


Fig. 2. The FedCross Framework.

Figure 2 presents the framework for FedCross, which shows the two processes above, i.e., model training and global model generation. Assume that there are a total of N clients. In each FL round, there are K clients participating in local training, where $K \leq N$. The model training process trains middleware models, which consists of 4 steps:

- **Step 1 (Middleware Model Dispatching):** The cloud server randomly dispatches K middleware models to K local clients, where each client receives one middleware model.
- **Step 2 (Middleware Model Training):** Clients train their received middleware models independently with local data.
- **Step 3 (Model Uploading):** All the clients upload their trained middleware models to the cloud server.

Algorithm 1: The FedCross Algorithm

Input: **i)** $round$, # of training rounds; **ii)** C, the set of clients; **iii)** K, # of clients participating in each FL round.

Output: w_g, the global model.

1 **FedCross**($round$,C,K) **begin**
2 $W \leftarrow [w_0^1, w_0^2, ..., w_0^K]$ // initialize the model list;
3 **for** r = *0, ..., $round-1$* **do**
4 $L_c \leftarrow$ Random select K clients from C;
5 $L_c \leftarrow Shuffle(L_c)$;
6 /*parallel for block*/
7 **for** i = *1, ..., K* **do**
8 $v_{r+1}^i \leftarrow$*LocalTraining*$(w_r^i, L_c[i])$;
9 $W[i] \leftarrow v_{r+1}^i$;
10 **end**
11 **for** i = *1, ..., K* **do**
12 $v_{co}^i \leftarrow$*CoModelSel*(v_{r+1}^i, W);
13 $w_{r+1}^i \leftarrow$*CrossAggr*(v_{r+1}^i, v_{co}^i);
14 **end**
15 $W \leftarrow [w_{r+1}^1, w_{r+1}^2, ..., w_{r+1}^K]$;
16 **end**
17 $w_g \leftarrow$*GlobalModelGen*(W);
18 **return** w_g;
19 **end**

- **Step 4 (Multi-Model Cross-Aggregation):** For each middleware model m_i ($1 \leq i \leq K$), FedCross chooses another middleware model m_j ($j \neq i$) as the collaborative model. By aggregating each middleware model and its collaborative model with weights of α and $1-\alpha$, respectively, the cloud server generates K new middleware models (i.e., $m'_1, ..., m'_K$) for the next-round training.

The global model generation process aggregates multiple trained middleware models to generate a global model. Since in FedCross the global model is only used for the model deployment, the global model generation does not need to be performed in every FL round.

B. The FedCross Algorithm

Algorithm 1 presents the pseudo-code of our FedCross approach. Line 2 initializes a list, W, of K dispatched models. Lines 3-16 present the model training process. Line 4 randomly selects K clients for each round's model training, where L_c is the list of selected clients. Line 5 shuffles the order of the selected models, with which each dispatched model is given an equal chance to be trained by the client. Note that, without shuffling, each middleware model will be dispatched to the clients encountered in the previous training rounds with a high probability. Lines 7-10 dispatch models to the corresponding clients and conduct local training process. In Line 8, each client trains the received model using local data and uploads the retrained local model to the cloud server. In Line 9, the cloud server updates the model list W using the received trained model. In Line 12, the function *CoModelSel* selects a collaborative model for each uploaded model. In Line 13, the function *CrossAggr* aggregates each uploaded model with its collaborative model to generate K models. Line 15 updates the dispatched model list W using these generated models. In Line 17, the function *GlobalModelGen* generates a global model for the deployment by aggregating all the models in W. Since the global model does not participate in the model training, the global model generation can be performed asynchronously at any time. The following will detail the key parts of FedCross and analyze its convergence.

1) Collaborative Model Selection (CoModelSel): To facilitate knowledge exchange between models, FedCross selects a collaborative model for each in the uploaded model list for cross-aggregation. According to model characteristics, we design three following model selection criteria to accommodate different purposes: i) *adequacy-and-diversity of participation*, ii) *minimizing gradient divergence*, and iii) *maximizing the knowledge acquisition*.

Since each middleware model is trained on a client, the knowledge acquired by each model is different. To fully exploit the information in the uploaded models, the *adequacy-and-diversity criterion* encourages each model to update other models as much as possible. This way, each middleware model can acquire diverse knowledge. Based on this criterion, we ordinally select a collaborative model from the middleware model list for the target model.

Since middleware models trained on different clients inevitably have differences, the *gradient divergence minimization criterion* encourages each model to find a similar collaborative model for the cross-aggregation to minimize gradient divergence in each cross-aggregation. Based on this criterion, we present *the highest similarity* strategy, which selects the most similar model to the target model.

The *knowledge maximization criterion* encourages each model to obtain more knowledge at each training round. Since models with high similarity have similar knowledge, contrary to the gradient divergence minimization criteria, the knowledge maximization criteria prefer to select a model with low similarity to the target model. Based on this criterion, we present *the lowest similarity* strategy, which selects the least similar model for the target model. The details of the three model selection strategies (i.e., in-order, highest similarity, lowest similarity) are as follows:

In-order strategy: For the i^{th} model, the cloud server selects the $((i + (r\%(K-1)+1))\%K)^{th}$ model as the collaborative model in the r^{th} training round. The in-order strategy is as follows:

$$CoModelSel(v_r^i, W) = W[(i + (r\%(K-1)+1))\%K],$$

where W is the list of uploaded local model parameters, and K is the number of uploaded models. With this strategy, all the upload models are chosen as collaborative models in each round. Note that, in every $(K-1)$ rounds of training, each middleware model collaborates with all the other $(K-1)$ models once.

The highest similarity strategy: By calculating the model similarity between the uploaded models, each middleware

model aggregates the model with the highest similarity as follows:

$$CoModelSel(v_r^i, W) = \underset{v \in W \setminus \{v_r^i\}}{\arg\max} \ Similarity(v_r^i, v),$$

where W is a list of uploaded local model parameters and *Similarity(·)* is a function to calculate the model similarity. Note that a higher *Similarity(·)* value means a higher similarity between the two models.

The lowest similarity strategy: According to the definition of the highest similarity strategy, the lowest similarity strategy encourages each model to select the model with the least similarity as the collaborative model:

$$CoModelSel(v_r^i, W) = \underset{v \in W \setminus \{v_r^i\}}{\arg\min} \ Similarity(v_r^i, v).$$

In this paper, since the classic cosine similarity can accurately reflect the angles of gradients, we adopt it as the measure as follows:

$$Similarity(X, Y) = \frac{\sum_{i=1}^{n} X_i \times Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} + \sqrt{\sum_{i=1}^{n} Y_i^2}},$$

where X and Y are two models, n indicates the number of parameters, and X_i indicates the i^{th} parameter in X. We would like to leave interesting topics of using other measures (e.g., Euclidean Distance) as our future work.

Compared with both in-order and the lowest similarity strategies, there are obvious flaws in the highest similarity strategy. Since the goal of FedCross is still to train a high-performance global model, the collaborative model selection strategy should guide all middleware models to be optimized in a similar direction. Although the *the highest similarity* strategy makes the lowest gradient divergence in each cross-aggregation, from a global perspective, such strategy makes models with high similarity increasingly similar, and it is more and more difficult for dissimilar models to share knowledge. At the end of FL training, middleware models are clustered into several groups, and the optimization directions of such groups are different. Finally, in the deployment phase, more serious gradient conflicts than ever will occur in the aggregation of the global model.

2) Cross-Aggregation (CrossAggr): The cross-aggregation is a novel multi-to-multi aggregation method, which fuses each upload model with its collaborative model with the weight α. Suppose that v_r^i is an uploaded model and v_{co}^i is its collaborative model. The cross-aggregation process is as follows:

$$CrossAggr(v_r^i, v_{co}^i) = \alpha \times v_r^i + (1 - \alpha) \times v_{co}^i,$$

where $\alpha \in [0.5, 1.0)$ is a hyperparameter used to determine the weight of the aggregation. The adjustment of α is important and difficult. If α is small, the gradient conflict will become serious. If α is large, it is difficult for the model to learn the knowledge of the collaborative model. Thus, we conduct an ablation study to confirm the reasonable value space of α by evaluating the performance of FedCross with different α values in Section IV-E1.

3) Global Model Generation: The global model generation phase is the same as the traditional FL methods. In FedCross, the global model does not participate in model training and is only used for model deployment. Thus, the global model can be performed asynchronously with model training. The global model is obtained by the following formula:

$$w_g = \frac{1}{K} \sum_{i=1}^{K} w_r^i$$

where w_r^i is the parameters of the i^{th} model in the dispatched model list, and r is the number of the current training round.

C. Convergence Analysis

Inspired by the proof of the convergence of traditional one-to-multi FL approach [37], [47], we prove the convergence of FedCross as follows.

1) Notations: Assume that all clients adopt Stochastic Gradient Descent (SGD) as the optimizer. Let t be the number of rounds of the current SGD iteration on clients, and w_t^i be the parameters of the i^{th} middleware model. After exactly one SGD iteration, we can get the parameters of some local model, i.e., v_{t+1}^i, by using the following model update formula:

$$v_{t+1}^i = w_t^i - \eta_t \nabla f_i(w_t^i, \xi_t^i),$$

Assuming that each local model is uploaded to the cloud server in every E iterations and $i' = (i + (t\%E)\%(N-1) + 1)\%N$, we have

$$w_{t+1}^i = \begin{cases} v_{t+1}^i, & if (t+1)\%E \neq 0 \\ \alpha v_{t+1}^i + (1-\alpha) v_{t+1}^{i'}, & if (t+1)\%E = 0 \end{cases},$$

Since FedCross generates a global model by aggregating all the middleware models, we use two variables $\overline{v}_t$ and $\overline{w}_t$ to represent the aggregated model of all middleware models:

$$\overline{v}_t = \frac{1}{N} \sum_{i=1}^{N} v_t^i, \ \overline{w}_t = \frac{1}{N} \sum_{i=1}^{N} w_t^i.$$

We define g_t^i to denote the gradients of the model in the i^{th} client after training with a data batch ξ_t^i:

$$g_t^i = \nabla f_i(w_t^i; \xi_t^i).$$

2) Proofs of Key Lemmas: We analyze the convergence of FedCross based on three assumptions for the loss function of each client (i.e., $f_1, f_2, ...,$ or f_N), including L-smooth assumption (Assumption 3.1), μ-convex assumption (Assumption 3.2), and variance/mean bound assumption for stochastic gradients (Assumption 3.3), which have been used in prior works [47]–[49].

Assumption 3.1: f_i is L-smooth satisfying $||\nabla f_i(w) - \nabla f_i(w')|| \leq L||w - w'||$, where $i \in \{1, 2, \cdots, N\}$.

Assumption 3.2: f_i is μ-convex satisfying $||\nabla f_i(w) - \nabla f_i(w')|| \geq \mu||w - w'||$, where $i \in \{1, 2, \cdots, N\}$ and $\mu \geq 0$.

Assumption 3.3: The variance of stochastic gradients is upper bounded by σ^2 and the expectation of squared norm of stochastic gradients is upper bounded by G^2, i.e., $\mathbb{E}||\nabla f_i(w; \xi) - \nabla f_i(w)||^2 \leq \sigma^2$, $\mathbb{E}||\nabla f_i(w; \xi)||^2 \leq G^2$, where ξ is a data batch of the i^{th} client in the t^{th} FL round.

Assume that in FedCross all the N clients are participating in every FL training round, and we employ the in-order selection strategy. Let $\{v_r^1, v_r^2, .., v_r^N\}$ be the set of uploaded local model parameters in the $(r-1)^{th}$ round, $\{w_r^1, w_r^2, .., w_r^N\}$ be the set of cross-aggregated model parameters, and $i' = (i + r\%(N-1) + 1)\%N$ be the index of collaborative model of the i^{th} middleware model. Based on the implementation of our in-order strategy, we have

$$w_r^i = \alpha v_r^i + (1-\alpha) v_r^{i'}. \tag{1}$$

Since in the in-order strategy each uploaded model is selected as a collaborative model for cross-aggregation, we have

$$\sum_{i=1}^{N} w_r^i = \sum_{i=1}^{N} (\alpha v_r^i + (1-\alpha) v_r^{i'}) = \sum_{i=1}^{N} v_r^i \tag{2}$$

According to Equations 1-2, we have Lemma 3.4 as follows.

Lemma 3.4: Let $w_r^i = \alpha v_r^i + (1-\alpha) v_r^{i'}$, $\alpha \in [0,1]$, and $\overline{w}_r = \sum_{i=1}^{N} w_r^i$. We have

$$||\overline{w}_r - w^\star||^2 \leq \frac{1}{N}\sum_{i=1}^{N} ||w_r^i - w^\star||^2 \leq \frac{1}{N}\sum_{i=1}^{N} ||v_r^i - w^\star||^2,$$

where $w^\star$ is the optimal parameters for the global loss function $F(\cdot)$. In other words, $\forall w, F^\star \leq F(w)$, where $F^\star$ denotes $F(w^\star)$.

Proof: We can derive the following inequality:

$$\begin{aligned}\sum_{i=1}^{N} ||w_r^i - w^\star||^2 &= \sum_{i=1}^{N} ||\alpha v_r^i + (1-\alpha) v_r^{i'} - w^\star||^2 \\ &= \sum_{i=1}^{N} (||v_r^i - w^\star||^2 - \alpha(1-\alpha)||v_r^i - v_r^{i'}||^2) \\ &\leq \sum_{i=1}^{N} ||v_r^i - w^\star||^2.\end{aligned}$$

Since $\overline{v}_r = \overline{w}_r = \frac{1}{N}\sum_{i=1}^{N} w_r^i$ holds, by using the AM–GM inequality, we can obtain:

$$||\overline{v}_r - w^\star||^2 \leq \frac{1}{N}\sum_{i=1}^{N} ||w_r^i - w^\star||^2.$$

. □

To facilitate the convergence analysis of FedCross, we present Lemmas 3.5-3.6.

Lemma 3.5: (Results of one step SGD). If $\eta_t \leq \frac{1}{4L}$ holds, we have:

$$\begin{aligned}\mathbb{E}||\overline{v}_{t+1} - w^\star||^2 \leq &\frac{1}{N}\sum_{i=1}^{N} (1-\mu\eta_t)||w_t^i - w^\star||^2 \\ &+ \frac{1}{N}\sum_{i=1}^{N} ||w_t^i - w_{t_0}^i||^2 + 10\eta_t^2 L\Gamma.\end{aligned}$$

Proof: By using the AM–GM inequality, it holds that:

$$\begin{aligned}||\overline{v}_{t+1} - w^\star||^2 &\leq \frac{1}{N}\sum_{i=1}^{N} ||v_{t+1}^i - w^\star||^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} (||v_t^i - w^\star||^2 - 2\eta_t\langle v_t^i - w^\star, g_t^i\rangle + \eta_t^2 ||g_t^i||^2).\end{aligned}$$

Let $P_1 = -2\eta_t\langle w_t^i - w^\star, g_t^i\rangle$ and $P_2 = \eta_t^2 \sum_{i=1}^{N} ||g_t^i||^2$. By using μ-convex (Assumption 3.2), we have:

$$P_1 \leq -2\eta_t(f_i(v_t^i) - f_i(w^\star)) - \mu\eta_t ||w_t^i - w^\star||^2. \tag{3}$$

By using L-smooth (Assumption 3.1), we obtain:

$$P_2 \leq 2\eta_t^2 L(f_i(w_t^i) - f_i^\star). \tag{4}$$

When $(t+1)\%E \neq 0$ and $v_t^i = w_t^i$ hold, according to Equations 3-4, we have:

$$\begin{aligned}||\overline{v}_{t+1} - w^\star||^2 \leq &\frac{1}{N}\sum_{i=1}^{N} [(1-\mu\eta_t)||v_t^i - w^\star||^2 - 2\eta_t(f_i(w_t^i) - f_i(w^\star)) \\ &+ 2\eta_t^2 L(f_i(w_t^i) - f_i^\star)].\end{aligned}$$

Let $P_3 = \frac{1}{N}\sum_{i=1}^{N} [-2\eta_t(f_i(w_t^i) - f_i(w^\star)) + 2\eta_t^2 L(f_i(w_t^i) - f_i^\star)]$. It holds that:

$$P_3 = -\frac{2\eta_t(1-\eta_t L)}{N}\sum_{i=1}^{N}(f_i(w_t^i) - F^\star) + \frac{2\eta_t^2 L}{N}\sum_{i=1}^{N}(F^\star - f_i^\star).$$

Let $\Gamma = F^\star - \frac{1}{N}\sum_{i=1}^{N} f_i^\star$ and $\phi = 2\eta_t(1 - L\eta_t)$. We have:

$$P_3 = -\frac{\phi}{N}\sum_{i=1}^{N}(f_i(w_t^i) - F^\star) + 2\eta_t^2 L\Gamma.$$

Let $P_4 = -\frac{1}{N}\sum_{i=1}^{N}(f_i(w_t^i) - F^\star)$, $t_0\%E = 0$ and $t - t_0 \leq E$. It holds that:

$$P_4 = -\frac{1}{N}\sum_{i=1}^{N}(f_i(w_t^i) - f_i(w_{t_0}^i) + f_i(w_{t_0}^i) - F^\star).$$

Based on the Cauchy–Schwarz inequality, we can derive that:

$$\begin{aligned}P_4 \leq &\frac{1}{2N}\sum_{i=1}^{N}(\eta_t ||\nabla f_i(w_{t_0}^i)||^2 + \frac{1}{\eta_t}||w_t^i - w_{t_0}^i||^2) \\ &- \frac{1}{N}\sum_{i=1}^{N}(f_i(w_{t_0}^i) - F^\star) \\ \leq &\frac{1}{2N}\sum_{i=1}^{N}\left[2\eta_t L(f_i(w_{t_0}^i) - f_i^\star) + \frac{1}{\eta_t}||w_t^i - w_{t_0}^i||^2\right] \\ &- \frac{1}{N}\sum_{i=1}^{N}(f_i(w_{t_0}^i) - F^\star).\end{aligned} \tag{5}$$

Note that, since $\eta \leq \frac{1}{4L}$, $\eta_t \leq \phi \leq 2\eta_t$ and $\eta_t L \leq \frac{1}{4}$, according to Equation 5, we have:

$$\begin{aligned}P_3 \leq &\frac{\phi}{2N}\sum_{i=1}^{N}\left[2\eta_t L(f_i(w_{t_0}^i) - f_i^\star) + \frac{1}{\eta_t}||w_t^i - w_{t_0}^i||^2\right] \\ &- \frac{\phi}{N}\sum_{i=1}^{N}(f_i(w_{t_0}^i) - F^\star) + \eta_t^2 L\Gamma \\ \leq &\frac{\phi}{2\eta_t N}\sum_{i=1}^{N}||w_t^i - w_{t_0}^i||^2 + (\phi\eta_t L + 2\eta_t^2 L)\Gamma + \frac{\phi}{N}\sum_{i=1}^{N}(F^\star - f_i^\star) \\ \leq &\frac{1}{N}\sum_{i=1}^{N}||w_t^i - w_{t_0}^i||^2 + 10\eta_t^2 L\Gamma.\end{aligned}$$

□

Lemma 3.6: In FedCross, the cross-aggregation occurs every E iteration. For arbitrary t, there always exists $t_0 \leq t$ while t_0 is the nearest cross-aggregation to t. As a result, $t - t_0 \leq E - 1$

holds. Given the constraint on learning rate from [47], we know that $\eta_t \leq \eta_{t_0} \leq 2\eta_t$. It follows that:

$$\frac{1}{N}\sum_{i=1}^{N}||w_t^i - w_{t_0}^i||^2 \leq 4\eta_t^2(E-1)^2G^2.$$

Proof: Let $t_0 \% E = 0$ and $t - t_0 \leq E$. We have:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}||w_t^i - w_{t_0}^i||^2 &= \frac{1}{N}\sum_{i=1}^{N}\left|\left|\sum_{t=t_0}^{t_0+E-1}\eta_t \nabla f_{a_1}(w_t^{a_1};\xi_t^{a_1})\right|\right|^2 \\ &\leq (E-1)\sum_{t=t_0}^{t_0+E-1}\eta_t^2 G^2 \\ &\leq 4\eta_t^2(E-1)^2G^2.\end{aligned}$$

□

Based on Lemmas 3.4-3.6, we prove Theorem 1 as follows.

Theorem 1: Let E be the number of SGD iterations conducted within one FL round, and the whole training consists of r FL rounds. Let $t = r \times E$ be the total number of SGD iterations conducted so far, and $\eta_t = \frac{2}{\mu(t+\lambda)}$ be the learning rate. We have:

$$\mathbb{E}[F(\overline{w}_t)] - F^\star \leq \frac{L}{2\mu(t+\lambda)}\left[\frac{4B}{\mu} + \frac{\mu(\lambda+1)}{2}\Delta_1\right], \tag{6}$$

where $B = 10L\Gamma + 4(E-1)^2G^2$.

Proof: Let $\Delta_t = ||\overline{w}_t - w^\star||^2$ and $\Delta_t^{glb} = \frac{1}{N}\sum_{i=1}^{N}||w_t^i - w^\star||^2$. According to Lemma 3.4, 3.5, and 3.6, we have:

$$\Delta_{t+1} \leq \Delta_{t+1}^{glb} \leq (1-\mu\eta_t)\Delta_t^{glb} + \eta_t^2 B.$$

When the step size becomes smaller, we have $\eta_t = \frac{\beta}{t+\lambda}$ for some $\beta > \frac{1}{\mu}$, $\lambda > 0$ such that $\eta_t \leq min\left\{\frac{1}{\mu}, \frac{1}{4L}\right\} = \frac{1}{4L}$ and $\eta_t \leq 2\eta_{t+E}$.

Let $\theta = max\left\{\frac{\beta^2 B}{\mu\beta - 1}, (\lambda+1)\Delta_1\right\}$. We firstly prove $\Delta_t \leq \frac{\theta}{t+\lambda}$ by induction. When $t = 1$,

$$\Delta_1 = \Delta_1^{glb} = \frac{\lambda+1}{\lambda+1}\Delta_1 \leq \frac{\theta}{\lambda+1}. \tag{7}$$

Assuming that $\Delta_t \leq \Delta_t^{glb} \leq \frac{\theta}{\lambda+1}$, we have:

$$\begin{aligned}\Delta_{t+1} &\leq \Delta_{t+1}^{glb} \\ &\leq (1-\mu\eta_t)\Delta_t^{glb} + \eta_t^2 B \\ &\leq \frac{t+\lambda-1}{(t+\lambda)^2}\theta + \left[\frac{\beta^2 B}{(t+\lambda)^2} - \frac{\mu\beta-1}{(t+\lambda)^2}\theta\right] \\ &\leq \frac{\theta}{t+1+\lambda}.\end{aligned} \tag{8}$$

According to Equations 7-8, we have:

$$\Delta_t \leq \frac{\theta}{t+\lambda}. \tag{9}$$

From Assumption 3.1 and Equation 9, we obtain:

$$\mathbb{E}[f(\overline{w}_t)] - F^\star \leq \frac{L}{2}\Delta_t \leq \frac{\theta L}{2(t+\lambda)}. \tag{10}$$

If we set $\beta = \frac{2}{\mu}$ and $\lambda = max\{\frac{10L}{\mu}, E\} - 1$, we have $\eta_t = \frac{2}{\mu(t+\lambda)}$ and $\eta_t \leq 2\eta_{t+E}$ for $t \geq 1$. Then, it holds that:

$$\begin{aligned}\theta &= max\left\{\frac{\beta^2 B}{\mu\beta - 1}, (\lambda+1)\Delta_1\right\} \\ &\leq \frac{\beta^2 B}{\mu\beta - 1} + (\lambda+1)\Delta_1 \\ &\leq \frac{4B}{\mu^2} + (\lambda+1)\Delta_1.\end{aligned} \tag{11}$$

Based on Equations 10-11, we have:

$$\begin{aligned}\mathbb{E}[F(\overline{w}_t)] - F^\star &\leq \frac{L}{2(t+\lambda)}\left[\frac{4B}{\mu^2} + (\lambda+1)\Delta_1\right] \\ &= \frac{L}{2\mu(t+\lambda)}\left[\frac{4B}{\mu} + \frac{\mu(\lambda+1)}{2}\Delta_1\right].\end{aligned}$$

□

Theorem 1 indicates that the difference between the current loss $F(\overline{w}_t)$ and the optimal loss $F^\star$ is inversely related to t. From Theorem 1, we observe that as the value of t increases, the right side of Equation 6 in Theorem 1 will approach 0, indicating that FedCross will eventually converge. In addition, we can also find that the convergence rate of FedCross is similar to that of FedAvg, which has been analyzed in [47].

D. Training Acceleration Methods for FedCross

Although the vanilla FedCross (i.e., FedCross without any training acceleration) can achieve the best accuracy performance compared with traditional aggregation methods (see Section IV-C1), due to our proposed fine-grained training strategy, it still suffers from the slow convergence during FL training. Especially in each FL training round at the early stage of training, due to significant knowledge differences among clients, the knowledge learned by each middleware model is limited, resulting in the low performance of aggregated global models. However, as the number of training rounds increases, each middleware model gradually becomes well-trained with fully exchanged knowledge, leading to a notable increase in the similarity among middleware models. Meanwhile, the classification performance of the global model improves significantly as well. Note that for the cross-aggregation, the value of α determines how much new knowledge a model can learn from its collaborative model. Specifically, a larger α indicates less knowledge can be learned from its collaborative model, leading to slow convergence.

Since the fusion weight (i.e., α) of a middleware model is much higher than that of its collaborative model in each cross-aggregation process, FedCross needs a large number of training rounds to unify all the middleware models. To accelerate the convergence of FedCross, we propose two optimization methods (i.e., *propeller models* and *dynamic* α) by dividing its training procedure into two stages, where the first stage allows middleware models to learn from each other in a coarse-grained manner, while the second stage adopts a fine-grained heuristic to fine-tune the middleware models. This way, we can balance the convergence rate and accuracy performance for a better training procedure. The following details the two training acceleration methods:

- **Propeller models-based acceleration:** To fully exploit the information of uploaded middleware models, we use propeller models that are selected by the in-order selection strategy from the middleware model list. For each middleware model, we use multiple propeller models rather than one collaborative model to provide more knowledge that

can be learned by middleware models, thus significantly accelerating the training procedure.

- **Dynamic α-based acceleration**: To accelerate the overall training convergence, we encourage middleware models to learn more knowledge from their collaborative models in earlier FL training rounds. Along with the process of FL training, since each middleware model can learn more knowledge with a smaller value of α, we gradually increase the value of α from 0.5 to a specific threshold (e.g., $\alpha = 0.99$ used in our experiments).

IV. Experimental Results

To evaluate the performance of FedCross, we conducted extensive experiments on well-known datasets and underlying DNN models. The subsequent subsections aim to answer the following four research questions (RQs).

RQ1: (Validation of Motivation): Compared with FedAvg-based methods, can FedCross converge into a flatter valley?

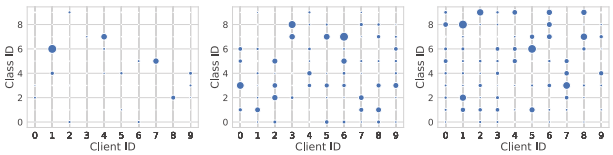
RQ2: (Superiority of FedCross): What are FedCross merits compared with state-of-the-art FedAvg-based methods?

RQ3: (Compatibility of FedCross): What is the performance of FedCross with different settings (e.g., client data distributions, DNN architectures, datasets)?

RQ4: (Benefits of FedCross Components): Can our proposed techniques improve classification performance?

A. Experimental Settings

We implemented FedCross on top of vanilla FedAVg by modifying its one-to-multi training scheme. Similar to the work in [5], in the experiments, we assumed that only 10% of clients are selected to participate in the training. To ensure comparison fairness, for all the involved FL methods, we set the local training batch size to 50 and performed five epochs for each local training round. For each client, we used SGD as the optimizer with a learning rate of 0.01 and a momentum of 0.5. For FedCross, we set $\alpha = 0.99$ and adopted the lowest similarity criterion to select collaborative models. We did not use other optimization methods (e.g., data augmentation) in all the following experiments. All the experimental results were obtained from an Ubuntu workstation with Intel i9 CPU, 32GB memory, and NVIDIA RTX 3080 GPU.



(a) $\beta = 0.1$ (b) $\beta = 0.5$ (c) $\beta = 1.0$

Fig. 3. Data distributions of selected clients with different non-IID settings.

1) Dataset Settings: We conducted experiments on five well-known datasets, i.e., CIFAR-10, CIFAR-100 [50], FEMNIST, Shakespeare, and Sent140 [51]. To evaluate the performance of FedCross within both IID and non-IID scenarios, we adopted the Dirichlet distribution [52] denoted by $Dir(\beta)$ to control the heterogeneity settings for datasets CIFAR-10 and CIFAR-100, where a smaller β indicates a higher data heterogeneity of clients. For these two datasets, we assumed that there are 100 clients involved in FL. To show the quantity differences of samples on clients within non-IID scenarios for the CIFAR-10 experiment, Figure 3 shows the data distributions of ten clients randomly selected from these 100 clients, where a larger blue dot indicates more samples on the corresponding device. Unlike CIFAR-10 and CIFAR-100, the other three datasets (i.e., FEMNIST, Shakespeare, and Sent140) are naturally non-IID in terms of data heterogeneity (i.e., number of samples and class imbalance). For FEMNIST, Shakespeare, and Sent140, we assumed that there are 180, 128, and 803 clients involved in FL, and each client has more than 100, 5700, and 40 samples, respectively.

TABLE I
Comparison between baseline methods and FedCross

Method	Category	Comm. Overhead
FedAvg	Classic	Low
FedProx	Global Control Variable	Low
SCAFFOLD	Global Control Variable	High
FedGen	Knowledge Distillation	Medium
CluSamp	Client Grouping	Low
FedCross	Multi-Model Guided	Low

2) Baseline Methods and Their Settings: We compared FedCross with five baseline methods, including the classic FedAvg and four state-of-the-art FL optimization methods (i.e., FedProx, SCAFFOLD, FedGen, and CluSamp). Table I compares FedCross with all the baseline methods from the perspectives of categories and communication overheads, where the baselines cover all the three FL optimization categories introduced in Section II-B. Note that, as a novel multi-model guided FL method, FedCross does not belong to any of the three existing categories. The following presents their settings.

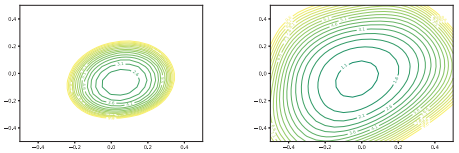
- **FedAvg** [5] is the most classic one-to-multi FL framework, wherein each FL training round the cloud server dispatches a global model to selected clients for FL training and aggregates their trained local models averagely to update the global model.
- **FedProx** [39] is a global control variable-based FL framework influenced by the hyper-parameter μ, where μ controls the weight of its proximal term. We set the best μ values for CIFAR-10, CIFAR-100, and FEMNIST to 0.01, 0.001, and 0.1, respectively. All these values are explored from the set $\{0.001, 0.01, 0.1, 1.0\}$.
- **SCAFFOLD** [37] is a global control variable-based FL framework, where the cloud server dispatches the variable with the same size as the model to guide local training in each training round.
- **FedGen** [46] is a KD-based method, which includes a built-in generator for proxy dataset generation. The subsequent experiments used the same settings as in [46].
- **CluSamp** [41] is a client grouping-based method. We select the model gradient similarity as the criteria for client grouping rather than the sample size. This is because directly exposing the distribution of data may increase the risk of privacy exposure. Furthermore, it may not be possible to directly obtain data distribution in real scenarios.

TABLE II
TEST ACCURACY COMPARISON FOR BOTH NON-IID AND IID SCENARIOS USING THREE DL MODELS

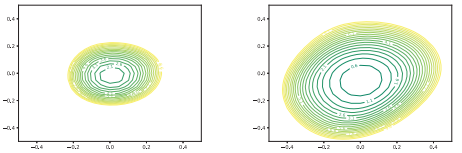
Model	Dataset	Heterogeneity Settings	Test Accuracy (%)					
			FedAvg	FedProx	SCAFFOLD	FedGen	CluSamp	FedCross
CNN	CIFAR-10	$\beta = 0.1$	46.12 ± 2.35	47.17 ± 1.65	49.12 ± 0.91	49.27 ± 0.85	47.09 ± 0.97	$\mathbf{55.70 \pm 0.74}$
		$\beta = 0.5$	52.82 ± 0.91	53.59 ± 0.88	54.50 ± 0.44	51.77 ± 0.73	54.00 ± 0.38	$\mathbf{58.74 \pm 0.67}$
		$\beta = 1.0$	54.78 ± 0.56	54.96 ± 0.60	56.75 ± 0.26	55.38 ± 0.66	55.82 ± 0.73	$\mathbf{62.16 \pm 0.42}$
		IID	57.64 ± 0.22	58.34 ± 0.15	59.98 ± 0.22	58.71 ± 0.19	57.32 ± 0.21	$\mathbf{62.97 \pm 0.22}$
	CIFAR-100	$\beta = 0.1$	28.37 ± 1.10	28.11 ± 1.03	30.32 ± 1.05	28.18 ± 0.58	28.63 ± 0.63	$\mathbf{32.53 \pm 0.45}$
		$\beta = 0.5$	30.01 ± 0.56	32.16 ± 0.50	33.49 ± 0.73	29.55 ± 0.41	33.04 ± 0.41	$\mathbf{36.87 \pm 0.24}$
		$\beta = 1.0$	32.34 ± 0.65	32.78 ± 0.13	34.95 ± 0.58	31.88 ± 0.65	32.92 ± 0.31	$\mathbf{37.65 \pm 0.36}$
		IID	32.98 ± 0.20	33.39 ± 0.25	35.11 ± 0.23	32.43 ± 0.20	34.97 ± 0.24	$\mathbf{38.42 \pm 0.18}$
	FEMNIST	–	81.67 ± 0.36	82.10 ± 0.61	81.65 ± 0.21	81.95 ± 0.36	80.80 ± 0.40	$\mathbf{83.49 \pm 0.18}$
ResNet-20	CIFAR-10	$\beta = 0.1$	45.11 ± 2.13	45.45 ± 3.42	50.46 ± 1.76	42.71 ± 3.48	44.87 ± 1.65	$\mathbf{53.79 \pm 2.91}$
		$\beta = 0.5$	60.56 ± 0.95	59.52 ± 0.74	58.85 ± 0.85	60.29 ± 0.68	59.55 ± 1.00	$\mathbf{69.38 \pm 0.30}$
		$\beta = 1.0$	62.99 ± 0.62	61.47 ± 0.66	61.63 ± 0.78	63.81 ± 0.33	63.32 ± 0.71	$\mathbf{71.59 \pm 0.31}$
		IID	67.12 ± 0.27	66.06 ± 0.22	65.20 ± 0.27	65.89 ± 0.17	65.62 ± 0.23	$\mathbf{75.01 \pm 0.09}$
	CIFAR-100	$\beta = 0.1$	31.90 ± 1.16	33.00 ± 1.21	35.71 ± 0.62	32.40 ± 1.45	34.34 ± 0.52	$\mathbf{39.40 \pm 1.43}$
		$\beta = 0.5$	42.45 ± 0.53	42.83 ± 0.54	42.33 ± 1.23	42.72 ± 0.32	42.07 ± 0.39	$\mathbf{50.39 \pm 0.24}$
		$\beta = 1.0$	44.22 ± 0.36	44.35 ± 0.36	43.28 ± 0.61	44.75 ± 0.57	43.29 ± 0.41	$\mathbf{53.09 \pm 0.29}$
		IID	44.42 ± 0.18	45.16 ± 0.24	44.37 ± 0.19	45.21 ± 0.19	43.59 ± 0.24	$\mathbf{54.07 \pm 0.19}$
	FEMNIST	–	78.47 ± 0.40	79.74 ± 0.54	76.14 ± 0.90	79.56 ± 0.34	79.28 ± 0.42	$\mathbf{80.93 \pm 0.52}$
VGG-16	CIFAR-10	$\beta = 0.1$	63.79 ± 3.90	63.35 ± 4.31	64.18 ± 3.86	66.52 ± 1.46	66.91 ± 1.83	$\mathbf{76.07 \pm 1.09}$
		$\beta = 0.5$	78.14 ± 0.67	77.70 ± 0.45	76.22 ± 1.37	78.9 ± 0.39	78.82 ± 0.40	$\mathbf{84.39 \pm 0.48}$
		$\beta = 1.0$	78.55 ± 0.21	79.10 ± 0.28	76.99 ± 1.01	79.75 ± 0.26	80.00 ± 0.37	$\mathbf{85.74 \pm 0.21}$
		IID	80.02 ± 0.05	80.77 ± 0.22	78.80 ± 0.07	80.00 ± 0.27	80.96 ± 0.12	$\mathbf{87.33 \pm 0.11}$
	CIFAR-100	$\beta = 0.1$	46.60 ± 1.45	45.88 ± 3.35	45.79 ± 1.77	49.04 ± 0.63	48.04 ± 1.76	$\mathbf{54.46 \pm 0.70}$
		$\beta = 0.5$	55.86 ± 0.64	55.79 ± 0.56	55.30 ± 0.61	56.40 ± 0.37	56.23 ± 0.34	$\mathbf{64.01 \pm 0.24}$
		$\beta = 1.0$	57.55 ± 0.51	57.40 ± 0.32	55.43 ± 0.45	57.15 ± 0.27	57.95 ± 0.35	$\mathbf{67.09 \pm 0.31}$
		IID	58.30 ± 0.23	58.49 ± 0.11	56.51 ± 0.08	57.62 ± 0.18	58.14 ± 0.20	$\mathbf{70.81 \pm 0.07}$
	FEMNIST	–	84.22 ± 0.46	83.98 ± 0.48	82.65 ± 0.74	84.69 ± 0.28	84.32 ± 0.36	$\mathbf{85.75 \pm 0.45}$
LSTM	Shakespeare	–	52.08 ± 0.29	52.53 ± 0.23	48.94 ± 0.18	53.87 ± 0.13	49.74 ± 0.74	$\mathbf{54.81 \pm 0.07}$
	Sent140	–	69.36 ± 0.20	68.63 ± 0.20	59.61 ± 0.06	69.32 ± 0.13	69.19 ± 0.14	$\mathbf{71.33 \pm 0.12}$

We implemented all FL methods on top of our own unified FL framework. For the baselines FedGen and CluSamp, we re-used the open source code from [53] and [54], respectively. For the baselines FedProx and SCAFFOLD, we re-implemented them according to their original papers [37], [39].

3) Model Settings: We investigated three well-known models, i.e., CNN, ResNet-20 [55], VGG-16 [56]. The CNN model was obtained from FedAvg [5], consisting of two convolutional and fully-connected layers. ResNet-20 and VGG-16 models were obtained from the official library [57].



(a) FedAvg with $\beta = 0.1$ (b) FedCross with $\beta = 0.1$



(c) FedAvg with IID (d) FedCross with IID

Fig. 4. Comparison between loss landscapes of FedAvg and FedCross.

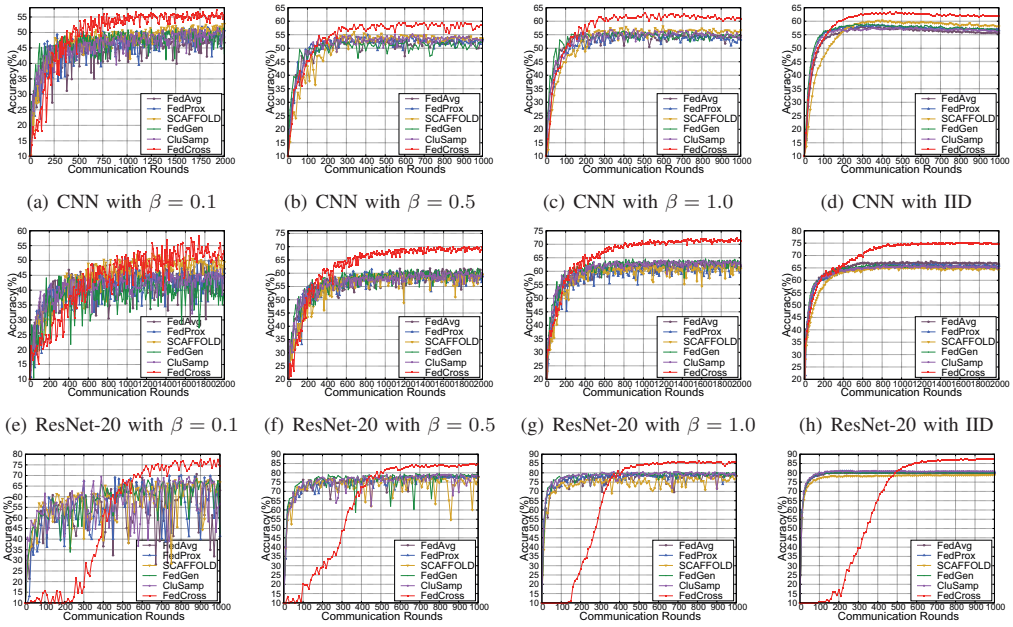
B. Motivation Validation (RQ1)

To validate whether a global model trained by FedCross can converge into a flatter valley than FedAvg, we checked four models for ResNet-20 that are trained using both FedAvg and FedCross on the CIFAR-10 dataset with $\beta = 0.1$ and IID scenarios, respectively. Since it is hard to draw the landscapes of all the involved clients together, Figure 4 only shows the loss landscapes of the obtained global models on top of their corresponding whole datasets. From this figure, we can observe that the global models trained by FedAvg are located in sharper areas than those obtained by FedCross. This implicitly reflects the fact that all the clients converge into nearby flat optimal solution areas, which is consistent with our observation in Figure 1. In other words, from the perspective of loss landscapes, FedCross can train a more generalized global model than that trained by FedAvg.

C. Performance Comparison (RQ2)

To show the superiority of FedCross, we compared it with the five baselines. For datasets CIFAR-10 and CIFAR-100, we considered one IID and three non-IID scenarios (with $\beta = 0.1, 0.5, 1.0$, respectively).

1) Comparison of Inference Accuracy: Table II presents the classification accuracy results for FedCross and all the five baselines on three datasets, where both IID and non-IID scenarios are all investigated. Note that, in the third column, we use β to control the heterogeneity settings for datasets CIFAR-10 and CIFAR-100 based on Diricht distribution Dir. Note that for all the baselines, we set the numbers of FL training rounds to 2000, 2000, and 1000 when using the CNN, ResNet-20, and VGG-16 models, respectively. We set the number of FL training rounds to 1000 for the ShakeSpeare dataset and 3000 for the Sent140 dataset. From this table, we can observe that FedCross achieves the highest accuracy for all different settings. For example, when using the VGG-16 model on CIFAR-10, FedCross outperforms the best baseline counterparts by 9.16% and 6.37% within IID and non-IID ($\beta = 0.1$) scenarios, respectively. Note that, by merely replacing the one-to-multi training scheme in the FedAvg framework with



(a) CNN with $\beta = 0.1$ (b) CNN with $\beta = 0.5$ (c) CNN with $\beta = 1.0$ (d) CNN with IID

(e) ResNet-20 with $\beta = 0.1$ (f) ResNet-20 with $\beta = 0.5$ (g) ResNet-20 with $\beta = 1.0$ (h) ResNet-20 with IID

(i) VGG-16 with $\beta = 0.1$ (j) VGG-16 with $\beta = 0.5$ (k) VGG-16 with $\beta = 1.0$ (l) VGG-16 with IID

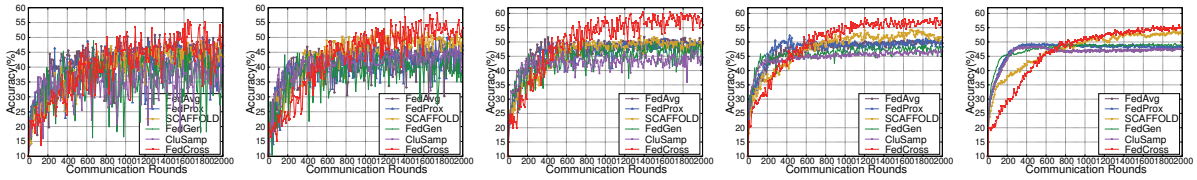
Fig. 5. Learning curves of different FL methods on CIFAR-10 dataset.

our proposed multi-to-multi training scheme, the classification performance of FedCross can be improved dramatically. One may argue that the classification performance improvements made by FedCross for FEMNIST are not as significant as the ones obtained for datasets CIFAR-10 and CIFAR-100. This is mainly because the data samples are simpler than the ones in datasets CIFAR-10 and CIFAR-100, where even FedAvg can achieve near-optimal classification performance. Moreover, we can observe that FedCross achieves the best performance on the two text datasets, i.e., ShakeSpeare and Sent140.

2) Comparison of Convergence Rate: Figure 5 shows the convergence trends of all the FL methods (including five baselines and FedCross) on the CIFAR-10 dataset, where Figures 5(a)-5(d) use CNN model, Figures 5(e)-5(h) use ResNet-20 model, and Figures 5(i)-5(l) use VGG-16 model. FedCross does not generate global models along with the FL training process. To enable the classification accuracy comparison between FedCross and the baselines, we additionally generated one pseudo-global model based on the middleware models in each round of FL training, and adopted this global model to derive the test accuracy information.

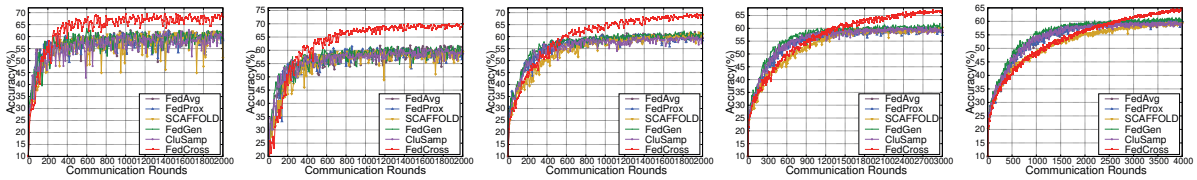
From Figure 5, we can find that FedCross consistently achieves the highest accuracy performance of the six FL methods in both non-IID and IID scenarios. Furthermore, we can observe that FedCross converges with much smaller fluctuations for all the investigated models and data settings. This is mainly because FedCross uses a multi-to-multi training scheme based on our proposed multi-model cross-aggregation, leading to the fine-grained training of the global model. Due to mitigated gradient divergence during local training and the available access of data across clients, FedCross can achieve the highest test accuracy results while lowering the risk of stuck-at-local-training. As shown in Figures 5(i)-5(l), at the beginning of FL training, FedCross lags behind the five baselines. This phenomenon is mainly because VGG-16 is a connection-intensive model with more than 130 million parameters, while ResNet-20 only has about 30 million parameters. Since VGG-16 is much larger than ResNet-20, it has a smaller performance acceleration than ResNet-20 at the early phase of FL training.

3) Comparison of Communication Overhead: For FedAvg, each training round involves the dispatching of K models and the upload of K models in total, where K is the number of selected clients. Although FedCross uses multiple models for FL training, it does not increase communication overhead than FedAvg. For FedCross, each participant client in local training receives only one model and uploads its trained version. Therefore, each training round of FedCross needs a communication of $2K$ models, which is the same as FedAvg. For FedProx and CluSamp, since their communication does not involve parameters other than models, their communication overhead is the same as FedAvg. For SCAFFOLD, it needs $2K$ models plus $2K$ global control variables in each FL training round, since the cloud server dispatches a global control variable to K clients and each client uploads global control variables to the cloud server in each round of FL training. For FedGen, since the cloud server dispatches an additional built-in generator to K clients in each FL training round, the communication overhead of FedGen is $2K$ models plus K generators. Based on the above analysis, we can find that FedCross requires the least communication overhead in each FL training round. Note that, as shown in Figure 5, although FedCross needs more rounds to achieve its best accuracy, for the highest accuracy that can be achieved by some FL methods, FedCross uses much fewer training rounds than the counterpart. This again shows the communication savings obtained by FedCross.



(a) $K = 5$ (b) $K = 10$ (c) $K = 20$ (d) $K = 50$ (e) $K = 100$

Fig. 6. Learning curves of different ResNet-20-based FL methods for different number of activated clients on CIFAR-10 dataset with $\alpha = 0.1$.



(a) $|C| = 50$ (b) $|C| = 100$ (c) $|C| = 200$ (d) $|C| = 500$ (e) $|C| = 1000$

Fig. 7. Learning curves of different ResNet-20-based FL methods for different number of clients on CIFAR-10 dataset with $\alpha = 0.5$.

D. Compatibility Analysis (RQ3)

1) Impacts of Client Data Distributions: For the same dataset, although FedCross can alleviate the performance degradation caused by various data heterogeneity factors, compared with their IID counterpart, the non-IID scenarios still lead to worse classification performance, especially when β is small. Furthermore, we find that in non-IID scenarios, FedCross requires more FL training rounds to converge. Note that all the above phenomena are also applicable to all the baselines. In other words, the training in non-IID scenarios is more difficult than the training in IID scenarios. From Table II and Figure 5, we can find that FedCross achieves the best performance for both IID and non-IID scenarios.

2) Impacts of Datasets: From Table II, we can observe that for all the three datasets, FedCross can significantly improve the classification performance compared with baselines, especially for complex datasets. As an example shown in Figure 5, we can observe that FedCross benefits CIFAR-10 and CIFAR-100 more than FEMNIST.

3) Impacts of Models: From Figure 5, we can observe that for the same dataset but different underlying DNN models, FedCross can still achieve the best classification performance. When adopting a model with a larger volume of parameters, although the convergence of FedCross may be slower than the baselines at the beginning of training, we can observe that FedCross can achieve much better classification accuracy at the end of training. Meanwhile, to achieve the best possible classification accuracy, FedCross uses much fewer training rounds. To accelerate the convergence of FedCross at the beginning of training, we proposed two training acceleration methods. Please refer to Section IV-E3 for more details.

4) Impacts of Activated Clients: Figure 6 compares FedCross with five baselines on the CIFAR-10 dataset using the ResNet-20 model within a non-IID scenario ($\beta = 0.1$), where the number of activated clients investigated in subfigures are 5, 10, 20, 50, and 100, respectively. From Figure 6, we can observe that FedCross can achieve the best results for all the cases. When $K < 20$, the maximal classification accuracy increases along with the increasing number of activated clients. However, when $K \geq 20$, the impact of the increasing number of activated clients is negligible. Moreover, we can find that the convergence becomes smoother when more activated clients are involved in the FL training.

5) Impacts of the Total Number of Clients: Figure 7 compares FedCross with five baselines on the CIFAR-10 dataset using the ResNet-20 model within a non-IID scenario ($\beta = 0.5$), where the total number of clients investigated in the subfigures is 50, 100, 200, 500, and 1,000, respectively. For each case, we selected 10% of clients to participate in local training. From Figure 7, we can observe that FedCross can achieve the best inference accuracy for all the cases. Note that in this experiment, since the total number of samples is fixed, the larger the total number of clients, the smaller the amount of data assigned to each client. As a result, we can find that when the number of clients increases, all the investigated FL methods need to use more training rounds for convergence.

TABLE III
TEST ACCURACY COMPARISON WITH DIFFERENT α SETTINGS

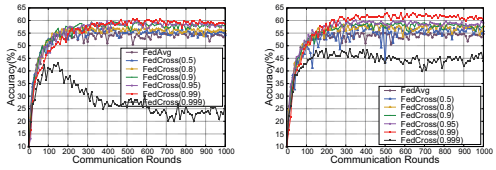
α	Selection Criteria		
	In-Order	Highest Similarity	Lowest Similarity
0.5	56.42 ± 0.54	56.33 ± 0.23	$\mathbf{56.81 \pm 0.91}$
0.8	56.66 ± 0.46	55.83 ± 0.85	$\mathbf{57.78 \pm 0.65}$
0.9	$\mathbf{58.69 \pm 0.46}$	46.91 ± 0.97	58.61 ± 0.48
0.95	59.12 ± 0.62	49.94 ± 0.94	$\mathbf{59.47 \pm 0.38}$
0.99	59.86 ± 0.40	49.70 ± 1.33	$\mathbf{62.16 \pm 0.42}$
0.999	40.85 ± 1.82	32.51 ± 3.39	$\mathbf{46.83 \pm 1.14}$

E. Ablation Studies (RQ4)

1) Evaluation of Model Selection Strategies: Table III presents the classification performance using three model selection strategies on the CIFAR-10 dataset within a non-IID scenario ($\beta = 1.0$). From Table III, we can observe that *the lowest similarity* strategy can achieve the best performance for five out of the given six α settings. Note that *the highest similarity* strategy achieves the worst performance for all the α settings. This is because the *the highest similarity* strategy makes middleware models with high similarity gradually get closer, while the models with low similarities become far away from each other, resulting in higher aggregation difficulty for

the global model. On the contrary, *the lowest similarity* reduces the distances between models with low similarities in each round of aggregation, which forces all the models to roughly optimize their local training towards similar directions. Regarding the in-order strategy, since every two models are aggregated within a finite number of rounds, the similarities between models will be limited to a certain range. However, its efficiency will be relatively lower compared with the one achieved by the highest similarity strategy. In summary, we recommend using either the lowest similarity strategy or the in-order strategy to select the collaboration model.

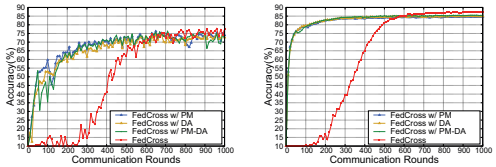
2) Evaluation of Aggregation Rate α: Figure 8 presents learning curves of both the in-order and lowest similarity strategies with six different settings of α. In Figure 8, FedCross performs best when $\alpha = 0.99$. We can observe that, as the value of α decreases, the performance of FedCross gradually decreases. However, when $\alpha = 0.999$, the performance of FedCross drops sharply. This is because the value of α is too large, which leads to less knowledge acquisition from the collaboration model. In other words, reducing the distance between models in each round of aggregation cannot offset the increase in model distance in each round of training. Therefore, the distances between models will gradually increase, resulting in a sharp decline in the performance of the global model. From this figure, we can find that a large α will improve the performance of FedCross since it supports the model aggregation in a more fine-grained way. Note that a large α may cause a sharp performance drop for the global model. In our experiments, FedCross achieves the best performance when $\alpha = 0.99$. We recommend using a $\alpha = 0.99$ in FedCross.



(a) In-Order (b) Lowest Similarity

Fig. 8. Learning curves of CNN-based FedCross with different α settings within a non-IID scenario ($\beta = 1.0$).

3) Evaluation of Training Acceleration Methods: We evaluated the performance of two training acceleration methods on the CIFAR-10 dataset using the VGG-16 model. Here, we considered three variants for FedCross. The first variant "FedCross w/ PM" uses propeller models to speed up training in the first 100 FL rounds. The second variant "FedCross w/ DA" uses dynamic α to speed up training for the first 100 FL rounds. The third variant "FedCross w/ PM-DA" uses propeller models for the first 50 rounds and dynamic α for the following 50 rounds to speed up training. Figure 9 presents the learning curves of FedCross in both non-IID ($\beta = 0.1$) and IID scenarios. From Figure 9, we can find that all the variants can significantly accelerate the training, but will slightly reduce the models' accuracy. In the non-IID scenario, the performance of the three variants is similar. In the IID scenario, the performance of "FedCross w/ PM-DA" is higher than the other two variants.



(a) non-IID ($\beta = 0.1$) (b) IID

Fig. 9. Learning curves of VGG-16-based FedCross with different training acceleration methods on CIFAR-10 dataset.

F. Discussion

1) Privacy Preserving: Similar to traditional one-to-multi FL methods, for FL training FedCross does not need any data distribution information for each local client. FedCross does not attempt to restore the user data by analyzing the model for each upload model. For each dispatched model, since it is aggregated with a collaborative model, and the model is dispatched randomly, clients cannot restore client data through the model and do not know the sources of received models. In addition, since the model dispatching, local training, and model update processes of FedCross are the same as the ones of FedAvg, FedCross can easily integrate existing privacy-preserving techniques [58]–[60] that are suitable for FedAvg to avoid privacy leaks.

2) Limitations: Although FedCross can achieve better performance than the baselines, its slow convergence on complex models is still a severe limitation that is worthy of further study. Although our proposed acceleration method can partially alleviate this problem, it may lead to slight performance degradation. Therefore, we need a more powerful acceleration method that does not affect the overall classification performance. Furthermore, at present, we only considered heterogeneous data for FedCross, where FedCross cannot deal with the training of heterogeneous models. These will be an interesting topic for our future work.

V. Conclusions

Due to the classic FedAvg-based local model aggregation scheme, traditional Federated Learning (FL) methods greatly suffer from the problems of slow convergence as well as low classification accuracy, especially for non-IID scenarios. To address this problem, this paper presents a novel FL framework named FedCross, which adopts our proposed multiple-to-multiple training scheme, i.e., multi-model cross aggregation. During the FL training, FedCross maintains a small set of intermediate models on the cloud server for the purpose of weighted fusion of similar local models. Since Fedcross fully respects the convergence characteristics of individual clients rather than simply averaging their local models, the local models can quickly converge to their local optimum counterparts. Comprehensive experimental results on well-known datasets show that FedCross outperforms state-of-the-art FL methods significantly in both IID and non-IID scenarios without causing extra communication overhead.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019), the Natural Science Foundation of China (62272170), the National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN), "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (22510750100), and Natural Science Foundation (NSF CCF-2217104). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, and Cyber Security Agency of Singapore. Mingsong Chen is the corresponding author (mschen@sei.ecnu.edu.cn).

REFERENCES

[1] G. Wang, H. Guo, A. Li, X. Liu, and Q. Yan, "Federated iot interaction vulnerability analysis," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 1517–1530.

[2] S. Liu, H. Su, Y. Zhao, K. Zeng, and K. Zheng, "Lane change scheduling for autonomous vehicle: A prediction-and-search framework," in *Proc. of ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 3343–3353.

[3] Z. Qin, J. Tang, and J. Ye, "Deep reinforcement learning with applications in transportation," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 3201–3202.

[4] Z. Jia, Z. Wang, F. Hong, L. Ping, Y. Shi, and J. Hu, "Personalized deep learning for ventricular arrhythmias detection on medical Iot systems," in *Proc. of International Conference on Computer-Aided Design (ICCAD)*, no. 38, 2020, pp. 1–9.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[7] X. Zhang, M. Hu, J. Xia, T. Wei, M. Chen, and S. Hu, "Efficient federated learning for cloud-based aiot applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 40, no. 11, pp. 2211–2223, 2020.

[8] M. Hu, E. Cao, H. Huang, M. Zhang, X. Chen, and M. Chen, "Aiotml: A unified modeling language for aiot-based cyber-physical systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 42, no. 11, pp. 3545–3558, 2023.

[9] A. Li, L. Zhang, J. Wang, J. Tan, F. Han, Y. Qin, N. M. Freris, and X.-Y. Li, "Efficient federated-learning model debugging," in *Proc. of International Conf. on Data Engineering (ICDE)*, 2021, pp. 372–383.

[10] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. of Conference on Computer Communications (INFOCOM)*, 2020, pp. 1698–1707.

[11] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *Proc. of International Conference on Data Engineering (ICDE)*, 2022, pp. 965–978.

[12] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data," *Proc. of Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 5972–5984, 2021.

[13] X.-C. Li and D.-C. Zhan, "Fedrs: Federated learning with restricted softmax for label distribution non-iid data," in *Proc. of ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 995–1005.

[14] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[15] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[16] W. Zhuang, Y. Wen, X. Zhang, X. Gan, D. Yin, D. Zhou, S. Zhang, and S. Yi, "Performance optimization of federated person re-identification via benchmark analysis," in *Proc. of ACM International Conference on Multimedia (MM)*, 2020, pp. 955–963.

[17] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural computation*, vol. 9, no. 1, pp. 1–42, 1997.

[18] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *International conference on machine learning*. PMLR, 2016, pp. 1225–1234.

[19] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9636–9647.

[20] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park, "Swad: Domain generalization by seeking flat minima," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 405–22 418, 2021.

[21] M. P. Uddin, Y. Xiang, X. Lu, J. Yearwood, and L. Gao, "Mutual information driven federated learning," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 32, no. 7, pp. 1526–1538, 2020.

[22] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in *Proc. of The Web Conference (WWW)*, 2021, pp. 947–956.

[23] M. Hu, Z. Xia, D. Yan, Z. Yue, J. Xia, Y. Huang, Y. Liu, and M. Chen, "Gitfl: Uncertainty-aware real-time asynchronous federated learning using version control," in *IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2023, pp. 145–157.

[24] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 39, no. 12, pp. 3606–3621, 2021.

[25] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. on Parallel and Distr. Sys. (TPDS)*, vol. 33, no. 3, pp. 536–550, 2021.

[26] Y. Cui, K. Cao, G. Cao, M. Qiu, and T. Wei, "Client scheduling and resource management for efficient training in heterogeneous iot-edge federated learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 41, no. 8, pp. 2407–2420, 2022.

[27] D. Yan, M. Hu, Z. Xia, Y. Yang, J. Xia, X. Xie, and M. Chen, "Have your cake and eat it too: Toward efficient and accurate split federated learning," *arXiv preprint arXiv:2311.13163*, 2023.

[28] J. Liu, Y. Xu, H. Xu, Y. Liao, Z. Wang, and H. Huang, "Enhancing federated learning with intelligent model migration in heterogeneous edge computing," in *Proc. of International Conference on Data Engineering (ICDE)*, 2022, pp. 1586–1597.

[29] C. Yang, Q. Wang, M. Xu, Z. Chen, K. Bian, Y. Liu, and X. Liu, "Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data," in *Proc. of The Web Conference (WWW)*, 2021, pp. 935–946.

[30] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: an efficient federated learning framework for heterogeneous mobile clients," in *Proc. of Int. Conf. on Mobile Computing and Networking (MobiCom)*, 2021, pp. 420–437.

[31] R. Liu, M. Hu, Z. Xia, J. Xia, P. Zhang, Y. Huang, Y. Liu, and M. Chen, "Adapterfl: Adaptive heterogeneous federated learning for resource-constrained mobile computing systems," *arXiv preprint arXiv:2311.14037*, 2023.

[32] C. Jia, M. Hu, Z. Chen, Y. Yang, X. Xie, Y. Liu, and M. Chen, "Adaptivefl: Adaptive heterogeneous federated learning for resource-constrained aiot systems," *arXiv preprint arXiv:2311.13166*, 2023.

[33] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in iot devices," *Computer Networks*, vol. 204, p. 108693, 2022.

[34] A. Li, Y. Cao, J. Guo, H. Peng, Q. Guo, and H. Yu, "Fedcss: Joint client-and-sample selection for hard sample-aware noise-robust federated learning," *Proceedings of the ACM on Management of Data*, vol. 1, no. 3, pp. 1–24, 2023.

[35] A. Li, L. Zhang, J. Wang, F. Han, and X.-Y. Li, "Privacy-preserving efficient federated-learning model debugging," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2291–2303, 2021.

[36] J. Wu, Q. Liu, Z. Huang, Y. Ning, H. Wang, E. Chen, J. Yi, and B. Zhou, "Hierarchical personalized federated learning for user modeling," in *Proc. of The Web Conference (WWW)*, 2021, pp. 957–968.
[37] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. of International Conference on Machine Learning (ICML)*, 2020, pp. 5132–5143.
[38] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data." in *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 7865–7873.
[39] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems (MLSys)*, pp. 429–450, 2020.
[40] C. Chen, Z. Chen, Y. Zhou, and B. Kailkhura, "Fedcluster: Boosting the convergence of federated learning via cluster-cycling," in *Proc. of International Conference on Big Data (Big Data)*, 2020, pp. 5017–5026.
[41] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered sampling: Low-variance and improved representativity for clients selection in federated learning," in *Proc. of International Conference on Machine Learning (ICML)*, 2021, pp. 3407–3416.
[42] 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. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 174–10 183.
[43] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Proc. of Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2351–2363, 2020.
[44] K. Ozkara, N. Singh, D. Data, and S. Diggavi, "Quped: Quantized personalization via distillation with applications to federated learning," *Proc. of Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 3622–3634, 2021.
[45] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "Fedaux: Leveraging unlabeled auxiliary data in federated learning," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021.
[46] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. of International Conference on Machine Learning (ICML)*, 2021, pp. 12 878–12 889.
[47] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. of International Conference on Learning Representations (ICLR)*, 2020.
[48] Y. Zhang, M. J. Wainwright, and J. C. Duchi, "Communication-efficient algorithms for statistical optimization," *Proc. of Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1511–1519, 2012.
[49] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.
[50] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[51] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.
[52] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.
[53] https://github.com/zhuangdizhu/FedGen.
[54] https://github.com/Accenture//Labs-Federated-Learning/tree/clustered_sampling.
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[57] Pytorch, "Models and pre-trained weight," https://pytorch.org/vision/stable/models.html, 2022.
[58] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *Proc. of Int. Conf. on Big Data (Big Data)*, 2019, pp. 2587–2596.
[59] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 15, pp. 3454–3469, 2020.
[60] L. Sun, J. Qian, and X. Chen, "Ldp-fl: Practical private aggregation in federated learning with local differential privacy," in *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, 2021, pp. 1571–1578.