# Enhancing the Performance of Global Model by Improving the Adaptability of Local Models in Federated Learning

**Wujun Zhou**, **Shu Ding**, **Zelin Li** and **Wei Wang***

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{zhouwujun, dings, lizelin, wangw}@lamda.nju.edu.cn

## Abstract

Federated learning enables the clients to collaboratively train a global model, which is aggregated from local models. Due to the heterogeneous data distributions over clients and data privacy in federated learning, it is difficult to train local models to achieve a well-performed global model. In this paper, we introduce the adaptability of local models, i.e., the average performance of local models on data distributions over clients, and enhance the performance of the global model by improving the adaptability of local models. Since each client does not know the data distributions over other clients, the adaptability of the local model cannot be directly optimized. First, we provide the property of an appropriate local model which has good adaptability on the data distributions over clients. Then, we formalize the property into the local training objective with a constraint and propose a feasible solution to train the local model. Extensive experiments on federated learning benchmarks demonstrate that our method significantly improves the adaptability of local models and achieves a well-performed global model that consistently outperforms the baseline methods.

## 1 Introduction

Owing to the accessibility of large-scale datasets, deep neural networks have achieved great success over the years [He *et al.*, 2016]. However, a vast amount of training data may be distributed across plenty of clients, posing challenges on how to effectively utilize the client data [Li *et al.*, 2019; Kairouz *et al.*, 2019]. To handle these challenges, federated learning [McMahan *et al.*, 2017] has emerged as a distributed learning paradigm with privacy-preserving property. Federated learning enables the clients to collaboratively learn a well-performed global model by sharing local updates. It demonstrates the strength in fully utilizing data from the clients without requiring the upload of private data to the server. Due to its communication efficiency and privacy preservation, federated learning has been widely used in multiple domains [Hard *et al.*, 2020; Kang *et al.*, 2020; Jiang *et al.*, 2021; Zheng *et al.*, 2020; Khan *et al.*, 2021; Adnan *et al.*, 2022].

However, the generalization performance of federated learning relies on the assumption that the client data are independent and identically distributed (IID) [McMahan *et al.*, 2017]. In real scenarios, federated learning often encounters data heterogeneity, where the clients hold Non-IID data. As mentioned in earlier studies, data heterogeneity affects the effectiveness of federated learning [Zhao *et al.*, 2018; Li *et al.*, 2020]. Recent studies have proposed numerous methods to address the issue of degradation of model generalization performance in Non-IID settings. A line of work relies on clients uploading extra auxiliary variables during the model upload phase, such as gradients [Karimireddy *et al.*, 2020; Dai *et al.*, 2023], statistical distribution information [Duan *et al.*, 2021], etc., to modify server-side model updates. However, these methods introduce additional communication costs and the reliability of the uploaded information may impact the effectiveness of federated learning [Li *et al.*, 2022].

Another line of work focuses on aligning the local models with the global model. [Li *et al.*, 2020; Acar *et al.*, 2021] propose adding penalty terms to local objectives to prevent local models from deviating from the global model during local training. [Chen and Chao, 2022; Zhang *et al.*, 2022] propose replacing the local loss with a balanced loss to improve the local model's performance on classes with few samples. These methods aim to make local models converge near the global optimal stationary point. However, since each local model is trained with its own data and does not know the data distributions over other clients, these methods still struggle to achieve effective alignment between the global and local models.

In this paper, we introduce the adaptability of local models, i.e., the average performance of local models on data distributions over clients, and enhance the performance of the global model by improving the adaptability of local models. Since each client does not know the data distributions of other clients, the adaptability of the local model cannot be directly optimized. First, we provide the property of an appropriate local model which has good adaptability on the data distributions over clients. Then, we formalize the property into the local training objective with a constraint and propose a feasible solution to train the local model. During the model aggregation phase, we further pro-

*Wei Wang is the corresponding author.

pose a model aggregation method that allows local models with good adaptability to have large aggregation weights. We call this method **Fed**erated Learning with **A**daptability over **C**lient **D**istributions (FedACD). Extensive experiments on federated learning benchmarks demonstrate that our method significantly improves the adaptability of local models and leads to a well-performed global model that consistently outperforms the baseline methods.

## 2 Related Works

Federated learning [Li *et al.*, 2019; Kairouz *et al.*, 2019] enables clients to collaboratively learn a global model by sharing local updates conducted on the local data. The general federated learning method, FedAvg[McMahan *et al.*, 2017], introduces an iterative model averaging approach. However, previous studies demonstrate that the robust generalization performance of federated learning relies on the assumption that the client data are independent and identically distributed (IID) [McMahan *et al.*, 2017]. In Non-IID scenarios, the learned model may experience degradation in the generalization performance [Zhao *et al.*, 2018].

Many methods based on FedAvg have been proposed to tackle data heterogeneity. One line of work relies on clients to upload extra auxiliary variables during the model upload phase to modify the server-side model updates. Scaffold [Karimireddy *et al.*, 2020] requires clients to upload extra gradient information besides the updated model, utilizing control variates (variance reduction) to correct client drift in local updates, thereby improving the convergence rate of FL. CReFF [Shang *et al.*, 2022] uses the gradient information uploaded by the clients to train the virtual features [Luo *et al.*, 2021] on the server to retrain the classifier of the global model. FedNH [Dai *et al.*, 2023] initializes uniform class prototypes on the server and then sends the fixed prototypes to clients to guide training. However, these methods introduce additional communication costs, and the model's performance heavily relies on the reliability of the auxiliary variables. In scenarios with severe data heterogeneity or partial client sampling, the reliability and update frequency of auxiliary variables decrease, leading to undecent generalization performance [Li *et al.*, 2022].

Another line of work focuses on that data heterogeneity leads to a misalignment between local models and the global model and attempts to adjust local training loss to align local and global models. FedProx [Li *et al.*, 2020] proposes adding a proximal term into local objectives and penalizing gradient updates far from the global model. FedNova [Wang *et al.*, 2020] employs a normalized averaging approach to eliminate objective inconsistency while preserving rapid error convergence. FedROD [Chen and Chao, 2022] proposes to view each client's local training as an independent class imbalance problem and utilizes the balanced loss to replace the cross-entropy loss to adjust local objectives. FedLC [Zhang *et al.*, 2022] introduces a deviation bound to measure the gradient deviation after local updates and then calibrates the logit of each class before softmax cross-entropy based on local label distribution to alleviate the deviation. However, the alignment of local and global models remains an ongoing exploration.

## 3 Methods

We consider the following Federated Learning (FL) scenario. There are $M$ clients with distribution $\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^M$ and each client $m$ has data $\mathcal{S}^m = \{(\boldsymbol{x}_k^m, y_k^m)\}_{k=1}^{N^m}$ drawn from distribution $\mathcal{D}^m$, where $\boldsymbol{x}_k^m \in \mathbb{R}^d$ is the $d$ dimension sample , $y_k^m \in [1, 2, ..., C]$ is the label of $\boldsymbol{x}_k^m$, and $N^m = |\mathcal{S}^m|$ is the local sample size, $m \in [M]$. The goal of FL is to learn a global model $\phi_w$ with parameters $w$ over all training data $\mathcal{S} \triangleq \bigcup_m \mathcal{S}^m$ without data transmission. For a sample $\boldsymbol{x}$, $\boldsymbol{f}_w(\boldsymbol{x})$ is the logit vector of the global model $\phi_w$ on $\boldsymbol{x}$. The probability output, denoted $\boldsymbol{p}(\boldsymbol{x}) = (p_1, \ldots, p_C)$, is derived from $\boldsymbol{f}_w(\boldsymbol{x})$, where $p_i$ is the $i$-th element of $\boldsymbol{p}(\boldsymbol{x})$ and $p_i = \frac{e^{f_w^i(\boldsymbol{x})}}{\sum_{k=1}^{C} e^{f_w^k(\boldsymbol{x})}}$, where $f_w^k(\boldsymbol{x})$ is the $k$-th element of $\boldsymbol{f}_w(\boldsymbol{x})$. The predicted label is given by $\phi_w(\boldsymbol{x}) = \text{argmax}(p_1, \ldots, p_C)$. In federated learning, the global model $\phi_w$ is to minimize the following risk over client distributions:

$$\mathcal{R}(\phi_w) = \sum_{m=1}^{M} \frac{1}{M} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}^m} [P(\phi_w(\boldsymbol{x}) \neq y)]. \tag{1}$$

Let $P(\phi_w(\boldsymbol{x}) \neq y \mid y = i)$ represent the error rate of the global model $\phi_w$ on class $i$ and $P(y = i)$ represent the prior probability of class $i$, $i \in [C]$. We can get that $P(\phi_w(\boldsymbol{x}) \neq y) = \sum_{i=1}^{C} P(\phi_w(\boldsymbol{x}) \neq y \mid y = i)P(y = i)$. For simplicity of notation, let $\boldsymbol{\epsilon}^{\phi_w} = (\epsilon_1^{\phi_w}, ..., \epsilon_C^{\phi_w})$ denote the error rate of the global model $\phi_w$, where $\epsilon_i^{\phi_w} \geq 0$ represents the error rate of the global model on class $i$, i.e., $\epsilon_i^{\phi_w} = P(\phi_w(\boldsymbol{x}) \neq y | y = i)$. Let $\boldsymbol{\pi}^m = (\pi_1^m, \ldots, \pi_C^m)$ denote the prior distribution of client $m$, where $\pi_i^m$ represents the prior distribution of class $i$ on client $m$. From Equation (1), we get that

$$\begin{aligned}
&\mathcal{R}(\phi_w) = \sum_{m=1}^{M} \frac{1}{M} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}^m} [P(\phi_w(\boldsymbol{x}) \neq y)] \\
&= \sum_{m=1}^{M} \frac{1}{M} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}^m} \left[ \sum_{i=1}^{C} P(\phi_w(\boldsymbol{x}) \neq y \mid y = i)P(y = i) \right] \\
&= \sum_{m=1}^{M} \frac{1}{M} \sum_{i=1}^{C} \epsilon_i^{\phi_w} \cdot \pi_i^m = \sum_{m=1}^{M} \frac{1}{M} \boldsymbol{\epsilon}^{\phi_w} \cdot \boldsymbol{\pi}^m.
\end{aligned} \tag{2}$$

In federated learning, the global model is aggregated from local models. Prior works attempt to enhance global model performance by aligning local models with the global model, e.g., introducing regularization terms [Li *et al.*, 2020; Acar *et al.*, 2021] or using balanced losses [Chen and Chao, 2022; Zhang *et al.*, 2022]. However, since clients have heterogeneous data distributions, and each client does not know the data distributions of other clients, these methods have struggled to achieve effective alignment. The global model $\phi_w$ is to minimize the risk on the data distributions $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^M$ over clients in Equation (2). Let $\phi_{w_m}$ denote the local model of client $m$ with parameters $w_m$, $m \in [M]$. This motivates us that the local model $\phi_{w_m}$ should also achieve small risk on the data distributions $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^M$ over clients:

$$\mathcal{R}(\phi_{w_m}) = \sum_{n=1}^{M} \frac{1}{M} \boldsymbol{\epsilon}^{\phi_{w_m}} \cdot \boldsymbol{\pi}^n. \tag{3}$$

$\mathcal{R}(\phi_{w_m})$ implies the adaptability of local model $\phi_{w_m}$. If $\mathcal{R}(\phi_{w_m})$ is small, i.e., $\phi_{w_m}$ has good average performance on the data distributions $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^M$, we say that $\phi_{w_m}$ has good adaptability. Unfortunately, since client $m$ does not know the data distributions $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^{m-1}, \boldsymbol{\pi}^{m+1}, \ldots, \boldsymbol{\pi}^M$, it is impossible to directly minimize Equation (3) to guide the training process of $\phi_{w_m}$. Here, we discuss how to find a surrogate loss for Equation (3). For client $m$, we consider the specific model $\phi_{w^*}$ whose error rate $\boldsymbol{\epsilon}^{\phi_{w^*}}$ satisfies $\epsilon_i^{\phi_{w^*}} = \epsilon_j^{\phi_{w^*}}$ for $i \neq j$ and $i, j \in [C]$. The risk $\mathcal{R}(\phi_{w^*})$ over the client distributions for $\phi_{w^*}$ is:

$$\begin{aligned} \mathcal{R}(\phi_{w^*}) &= \sum_{n=1}^{M} \frac{1}{M} \sum_{k=1}^{C} \frac{\sum_{i=1}^{C} \epsilon_i^{\phi_{w^*}}}{C} \cdot \pi_k^n \\ &= \sum_{n=1}^{M} \frac{\sum_{i=1}^{C} \epsilon_i^{\phi_{w^*}}}{MC} \cdot \sum_{k=1}^{C} \pi_k^n = \frac{\sum_{i=1}^{C} \epsilon_i^{\phi_{w^*}}}{C} = \frac{\|\boldsymbol{\epsilon}^{\phi_{w^*}}\|_1}{C}. \end{aligned} \tag{4}$$

From Equation (4), we can find that $\mathcal{R}(\phi_{w^*}) = \frac{\|\boldsymbol{\epsilon}^{\phi_{w^*}}\|_1}{C}$ for any client distributions $\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^M$. This implies that $\phi_{w^*}$ may be a good choice for serving as the local model. Now, we provide the following theorem:

**Theorem 1.** *For client $m$, let $\boldsymbol{\pi}^m$ denote the data distribution of client $m$, and $\phi_{w_m}$ is the local model on client $m$ with error rate $\boldsymbol{\epsilon}^{\phi_{w_m}}$. Suppose $\|\boldsymbol{\epsilon}^{\phi_{w_m}}\|_1 = \|\boldsymbol{\epsilon}^{\phi_{w^*}}\|_1 = \epsilon^*$. If $\boldsymbol{\epsilon}^{\phi_{w_m}}$ does not satisfy the condition that $\epsilon_i^{\phi_{w_m}} = \epsilon_j^{\phi_{w_m}}$ for $i \neq j$ and $i, j \in [C]$, there exist sets of client distributions $\widehat{\boldsymbol{\pi}}^1, \ldots, \widehat{\boldsymbol{\pi}}^{m-1}, \boldsymbol{\pi}^m, \widehat{\boldsymbol{\pi}}^{m+1}, \ldots, \widehat{\boldsymbol{\pi}}^M$ on which the risk $\mathcal{R}(\phi_{w_m}) > \mathcal{R}(\phi_{w^*})$.*

*Proof.* Without loss of generality, we assume that $\boldsymbol{\pi}^m = (\frac{1}{C} + \delta_1^m, \ldots, \frac{1}{C} + \delta_C^m)$ where $-\frac{1}{C} \leq \delta_1^m, \ldots, \delta_C^m \leq 1 - \frac{1}{C}$ and $\sum_{k=1}^{C} \delta_k^m = 0$. Let $r = \arg\max_{k \in [C]} \epsilon_k^{\phi_{w_m}}$ be the class index with maximal error rate, $s = \arg\min_{k \in [C]} \epsilon_k^{\phi_{w_m}}$ is the class index with minimal error rate. Since $\boldsymbol{\epsilon}^{\phi_{w_m}}$ does not satisfy the condition that $\epsilon_i^{\phi_{w_m}} = \epsilon_j^{\phi_{w_m}}$ for $i \neq j$ and $i, j \in [C]$, we have $\epsilon_r^{\phi_{w_m}} > \epsilon_s^{\phi_{w_m}}$. Now we construct distribution $\widehat{\boldsymbol{\pi}}^n = (\widehat{\pi}_1^n, \ldots, \widehat{\pi}_C^n)$ where $\widehat{\pi}_k^n = \frac{1}{C} + \delta_k^n$ for $k \in [C], n \in [M] \backslash \{m\}$ with $-\frac{1}{C} \leq \delta_k^n \leq 1 - \frac{1}{C}$ and $\sum_{k=1}^{C} \delta_k^n = 0$. Let $0 < \theta < \frac{1}{C}$ be a constant. If $\delta_k^n, k \in [C], n \in [M] \backslash \{m\}$ satisfy the following condition [1]:

$$\sum_{t \neq m} \delta_k^t = \begin{cases} -\delta_k^m, & \text{if } k \in [M] \backslash \{r, s\} \\ \theta - \delta_k^m, & \text{if } k = r \\ -\theta - \delta_k^m, & \text{if } k = s \end{cases} \tag{5}$$

then the risk $\mathcal{R}(\phi_{w_m})$ over client distributions is

$$\begin{aligned} \mathcal{R}(\phi_{w_m}) &= \boldsymbol{\epsilon}^{\phi_{w_m}} \cdot \frac{\boldsymbol{\pi}^m + \sum_{n \in [M], n \neq m} \widehat{\boldsymbol{\pi}}^n}{M} \\ &= \frac{\epsilon^*}{C} + \frac{\theta}{M} \left( \epsilon_r^{\phi_{w_m}} - \epsilon_s^{\phi_{w_m}} \right) > \frac{\epsilon^*}{C}. \end{aligned}$$

[1] Here, we consider the non-trivial case that the number of clients $M$ is larger than the number of class $C$.

It is easy to find that the condition in Equation (5) can be satisfied, e.g.,

$$\delta_k^t = \begin{cases} \frac{-\delta_k^m}{M-1}, & \text{if } k \in [M] \backslash \{r, s\} \\ \frac{\theta - \delta_k^m}{M-1}, & \text{if } k = r \\ \frac{-\theta - \delta_k^m}{M-1}, & \text{if } k = s \end{cases}$$

for $k \in [C]$ and $t \in [M] \backslash \{m\}$. Since $\|\boldsymbol{\epsilon}^{\phi_{w^*}}\|_1 = \epsilon^*$, with Equation (4) we get $\mathcal{R}(\phi_{w^*}) = \frac{\epsilon^*}{C}$. Thus, we have $\mathcal{R}(\phi_{w_m}) > \mathcal{R}(\phi_{w^*})$. □

Theorem 1 indicates that for the model $\phi_{w_m}$ with the same generalization ability as $\phi_{w^*}$, i.e., $\|\boldsymbol{\epsilon}^{\phi_{w_m}}\|_1 = \|\boldsymbol{\epsilon}^{\phi_{w^*}}\|_1 = \epsilon^*$, if the model $\phi_{w_m}$ does not satisfy the condition that $\epsilon_i^{\phi_{w_m}} = \epsilon_j^{\phi_{w_m}}$ for $i \neq j$ and $i, j \in [C]$, $\phi_{w^*}$ is a better choice than $\phi_{w_m}$. The reason is that there exist sets of client distributions $\widehat{\boldsymbol{\pi}}^1, \ldots, \widehat{\boldsymbol{\pi}}^{m-1}, \boldsymbol{\pi}^m, \widehat{\boldsymbol{\pi}}^{m+1}, \ldots, \widehat{\boldsymbol{\pi}}^M$ on which $\mathcal{R}(\phi_{w_m}) > \frac{\epsilon^*}{C}$, while $\mathcal{R}(\phi_{w^*})$ is always $\frac{\epsilon^*}{C}$ for any distributions. In federated learning, the clients have heterogeneous data distributions, and each client does not know the distributions of other clients. In this way, we can train the local model according to the optimization in Equation (6) for each client m, $m \in [M]$:

$$\begin{aligned} \min \quad & \|\boldsymbol{\epsilon}^{\phi_{w_m}}\|_1, \\ \text{s.t.} \quad & \epsilon_i^{\phi_{w_m}} = \epsilon_j^{\phi_{w_m}}, \quad \forall i \neq j. \end{aligned} \tag{6}$$

Minimizing $\|\boldsymbol{\epsilon}^{\phi_{w_m}}\|_1$ can be achieved by minimizing the cross-entropy (CE) loss over samples of local data $\mathcal{S}^m$. However, ensuring that the error rate $\boldsymbol{\epsilon}^{\phi_{w_m}}$ satisfies the constraints $\epsilon_i^{\phi_{w_m}} = \epsilon_j^{\phi_{w_m}}$ for $i \neq j$ is challenging. In the following section, we propose a feasible solution for this. As defined, $\epsilon_i^{\phi_{w_m}} = P(\phi_{w_m}(\boldsymbol{x}) \neq y \mid y = i) = \sum_{j \neq i} P(\phi_{w_m}(\boldsymbol{x}) = j \mid y = i)$, where $P(\phi_{w_m}(\boldsymbol{x}) = j \mid y = i)$ represents the probability that local model $\phi_{w_m}$ predicts a sample of class $i$ as class $j$ and can be computed by averaging the $j$-th element of the probability outputs of the samples of class $i$. Therefore, it is possible to optimize the probability outputs of local model $\phi_{w_m}$ to force the error rate $\boldsymbol{\epsilon}^{\phi_{w_m}}$ to meet the constraints in Equation (6). First, we introduce the probability matrix $P^m$ for client $m$:

$$P^m = \begin{bmatrix} P_{11}^m & P_{12}^m & \ldots & P_{1C}^m \\ P_{21}^m & P_{22}^m & \ldots & P_{2C}^m \\ \vdots & \vdots & \ddots & \vdots \\ P_{C1}^m & P_{C2}^m & \ldots & P_{CC}^m \end{bmatrix},$$

where $P_{ij}^m = P(\phi_{w_m}(\boldsymbol{x}) = j \mid y = i)$. $P_{ij}^m$ can be evaluated as: $P_{ij}^m = \frac{1}{N_i^m} \sum_{(\boldsymbol{x}, y) \sim \mathcal{D}_i^m} p_j, i, j \in [C]$, where $p_j$ is the $j$-th element of the probability output $\boldsymbol{p}(\boldsymbol{x})$ of sample $\boldsymbol{x}$. Specifically, $p_j = \frac{e^{f_{w_m}^j(\boldsymbol{x})}}{\sum_{k=1}^{C} e^{f_{w_m}^k(\boldsymbol{x})}}$, where $f_{w_m}^k(\boldsymbol{x})$ is the $k$-th element of the logit vector $\boldsymbol{f}_{w_m}(\boldsymbol{x})$ of sample $\boldsymbol{x}$. It is easy to find that $\epsilon_i^{\phi_{w_m}} = 1 - P_{ii}^m$, $i \in [C]$. In this way, the constraints $\epsilon_i^{\phi_{w_m}} = \epsilon_j^{\phi_{w_m}}, i \neq j$, in Equation (6) become the probability constraints: $P_{ii}^m = P_{jj}^m$, $i \neq j$. We make the constraints hold during the local training process in the following way:

| Dataset | CIFAR-10 | | | CIFAR-100 | | | Tiny-ImageNet | | |
|---|---|---|---|---|---|---|---|---|---|
| NonIID ($\beta$) | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 |
| FedAvg | $75.63_{\pm0.75}$ | $68.35_{\pm2.43}$ | $60.47_{\pm3.74}$ | $41.97_{\pm0.24}$ | $39.57_{\pm0.60}$ | $38.12_{\pm0.11}$ | $45.76_{\pm0.21}$ | $40.24_{\pm0.31}$ | $36.09_{\pm0.33}$ |
| FedProx | $75.54_{\pm0.91}$ | $68.80_{\pm2.60}$ | $62.18_{\pm0.20}$ | $41.70_{\pm0.16}$ | $39.33_{\pm0.05}$ | $38.15_{\pm0.07}$ | $45.47_{\pm0.09}$ | $40.35_{\pm0.35}$ | $35.64_{\pm0.19}$ |
| FedNova | $75.19_{\pm1.15}$ | $67.02_{\pm2.90}$ | $56.63_{\pm1.88}$ | $41.63_{\pm0.12}$ | $39.38_{\pm0.46}$ | $37.88_{\pm0.39}$ | $45.72_{\pm0.12}$ | $40.36_{\pm0.24}$ | $35.47_{\pm0.43}$ |
| CReFF | $76.07_{\pm0.85}$ | $69.40_{\pm2.17}$ | $61.71_{\pm3.57}$ | $37.60_{\pm0.27}$ | $37.71_{\pm0.59}$ | $38.03_{\pm0.17}$ | $44.75_{\pm0.23}$ | $39.74_{\pm0.49}$ | $35.26_{\pm0.55}$ |
| FedROD | $\underline{77.53_{\pm0.86}}$ | $71.12_{\pm1.33}$ | $\underline{62.46_{\pm3.29}}$ | $42.02_{\pm0.15}$ | $40.15_{\pm0.44}$ | $38.37_{\pm0.18}$ | $46.18_{\pm0.26}$ | $42.02_{\pm0.14}$ | $37.81_{\pm0.39}$ |
| FedNTD | $76.01_{\pm0.47}$ | $70.41_{\pm0.76}$ | $60.48_{\pm1.41}$ | $43.05_{\pm0.22}$ | $39.90_{\pm0.31}$ | $37.70_{\pm0.29}$ | $\underline{46.86_{\pm0.18}}$ | $41.89_{\pm0.30}$ | $36.86_{\pm0.28}$ |
| FedDecorr | $74.51_{\pm0.37}$ | $\underline{71.80_{\pm1.81}}$ | $61.19_{\pm1.79}$ | $38.85_{\pm0.34}$ | $38.89_{\pm0.19}$ | $37.50_{\pm0.18}$ | $45.89_{\pm0.12}$ | $40.69_{\pm0.38}$ | $35.64_{\pm0.25}$ |
| FedLC | $76.76_{\pm0.56}$ | $69.40_{\pm1.40}$ | $52.71_{\pm4.09}$ | $41.92_{\pm0.39}$ | $39.85_{\pm0.52}$ | $35.27_{\pm0.16}$ | $46.66_{\pm0.12}$ | $41.37_{\pm0.14}$ | $36.63_{\pm0.38}$ |
| FedNH | $75.30_{\pm0.84}$ | $68.11_{\pm2.15}$ | $60.54_{\pm2.61}$ | $\underline{44.74_{\pm0.14}}$ | $\underline{41.74_{\pm0.24}}$ | $\underline{39.61_{\pm0.43}}$ | $45.09_{\pm1.95}$ | $\underline{42.31_{\pm0.85}}$ | $\underline{38.87_{\pm0.21}}$ |
| FedACD | $\mathbf{79.57_{\pm0.02}}$ | $\mathbf{73.13_{\pm0.52}}$ | $\mathbf{63.57_{\pm0.46}}$ | $\mathbf{49.08_{\pm0.08}}$ | $\mathbf{46.24_{\pm0.17}}$ | $\mathbf{43.22_{\pm0.46}}$ | $\mathbf{49.62_{\pm0.32}}$ | $\mathbf{45.29_{\pm0.14}}$ | $\mathbf{41.44_{\pm0.85}}$ |

Table 1: Performance(%) of the global models on test sets with uniform data distribution. The best in each setting is highlighted in **bold**, and the second best is highlighted in underline.

first, we force the probability matrix $P^m$ to have the form of $P^m_{ij} = \frac{1-P^m_{ii}}{C-1}, i \neq j$; then, we make $P^m_{ii} = P^m_{jj}$ for $i \neq j$ by adjusting $P^m_{ij} = P^m_{ji}$, i.e., $\frac{1-P^m_{ii}}{C-1} = \frac{1-P^m_{jj}}{C-1}$.

The local probability matrix $P^m$ is evaluated by averaging the probability outputs of each class's samples of the local data. Therefore, to force the local probability matrix $P^m$ to have the form of $P^m_{ij} = \frac{1-P^m_{ii}}{C-1}, i \neq j$, we flatten the misclassification probability for each sample . For a sample $\boldsymbol{x}$ with label $y$, its probability output $\boldsymbol{p}(\boldsymbol{x}) = (p_1, ..., p_C)$, where $p_y$ is the correct classification probability, and $p_k$, $k \neq y$, is the misclassification probability. Based on the probability output $\boldsymbol{p}(\boldsymbol{x})$, we construct the target vector $\boldsymbol{q}(\boldsymbol{x}) = (q_1, ..., q_C)$, where the misclassification probability $q_k$, $k \neq y$, is flattened. Specifically, $q_y = p_y$ and $q_k = \frac{1-p_y}{C-1}$ for $k \neq y$. Thus, we minimize the Kullback-Leibler (KL) divergence between $\boldsymbol{p}(\boldsymbol{x})$ and $\boldsymbol{q}(\boldsymbol{x})$ shown in Equation (7) to flatten the misclassification probability for the sample $\boldsymbol{x}$.

$$\begin{aligned}\mathcal{L}_1 &= \frac{1}{|\mathcal{S}^m|} \sum_{(\boldsymbol{x},y)\sim\mathcal{D}^m} \mathrm{KL}(\boldsymbol{p}(\boldsymbol{x}) \| \boldsymbol{q}(\boldsymbol{x})) \\ &= \frac{1}{|\mathcal{S}^m|} \sum_{(\boldsymbol{x},y)\sim\mathcal{D}^m} \sum_{i=1}^{C} p_i \log\left(\frac{p_i}{q_i}\right), \\ q_k &= \begin{cases} p_y, & \text{if } k = y, \\ \frac{1-p_y}{C-1}, & \text{if } k \neq y. \end{cases}\end{aligned} \tag{7}$$

In this way, by adjusting the probability output $\boldsymbol{p}(\boldsymbol{x})$ of each sample $\boldsymbol{x}$ with label $y$ to $(\frac{1-p_y}{C-1}, ..., p_y, ..., \frac{1-p_y}{C-1})$, the probability matrix $P^m$ has the form of $P^m_{ij} = \frac{1-P^m_{ii}}{C-1}$ for $i \neq j$.

To ensure that the local probability matrix $P^m$ meets the probability constraints $P^m_{ii} = P^m_{jj}$ for $i \neq j$, for the pair of classes $(i, j)$, the local probability matrix $P^m$ should further satisfy $P^m_{ij} = P^m_{ji}$. With the condition $P^m_{ij} = \frac{1-P^m_{ii}}{C-1}$ for $i \neq j$, $P^m_{ij} = P^m_{ji}$ is equivalent to $\frac{1-P^m_{ii}}{C-1} = \frac{1-P^m_{jj}}{C-1}$, i.e., $P^m_{ii} = P^m_{jj}, i \neq j$. To make $P^m_{ij} = P^m_{ji}$, if $P^m_{ij} > P^m_{ji}$, $P^m_{ij}$ should be decreased; otherwise, $P^m_{ij}$ should be increased. Directly adjusting $P^m_{ij}$ and $P^m_{ji}$ during the local training process is difficult. Since $P^m$ is evaluated based on the sample's logit vector, we can implicitly adjust $P^m_{ij}$ and $P^m_{ji}$ by modifying the logit margin between $f^i_{w_m}(\boldsymbol{x})$ and $f^j_{w_m}(\boldsymbol{x})$ for sample $\boldsymbol{x}$. Based on previous work [Menon *et al.*, 2021], we dynamically adjust the cross-entropy (CE) loss by modifying the logit margin term $\{f^i_{w_m}(\boldsymbol{x}) - f^y_{w_m}(\boldsymbol{x})\}$ for each sample $\boldsymbol{x}$ with label $y$ with the following loss function:

$$\mathcal{L}_2 = \frac{1}{|\mathcal{S}^m|} \sum_{(\boldsymbol{x},y)\sim\mathcal{D}^m} \log\left[1 + \sum_{i\neq y} e^{f^i_{w_m}(\boldsymbol{x}) - f^y_{w_m}(\boldsymbol{x}) + \log\left(\frac{P^m_{yi}}{P^m_{iy}}\right)}\right]. \tag{8}$$

For simplicity, we define $\Delta_{yi} = \frac{P^m_{yi}}{P^m_{iy}}$. If $\Delta_{yi} < 1$, the logit margin term $\{f^i_{w_m}(\boldsymbol{x}) - f^y_{w_m}(\boldsymbol{x})\}$ will be reduced to suppress class $y$'s relative margin towards class $i$, causing an increase of $\Delta_{yi}$. Conversely, if $\Delta_{yi} > 1$, the logit margin term $\{f^i_{w_m}(\boldsymbol{x}) - f^y_{w_m}(\boldsymbol{x})\}$ will be increased to relax class $y$'s relative margin towards class $i$, leading to a decrease of $\Delta_{yi}$. Finally, $\Delta_{yi} = 1$ implies that $P^m_{ii} = P^m_{jj}$ for $i \neq j$. Thus, the loss function to guide the local training process of $\phi_{w_m}$ is:

$$\mathcal{L} = \mathcal{L}_1 + \lambda\mathcal{L}_2, \tag{9}$$

where the parameter $\lambda$ controls the contribution of $\mathcal{L}_1$ and $\mathcal{L}_2$.

When confronted with severe data heterogeneity, certain classes of some clients may hold limited samples or even no samples. If class $k$ is a missing class with $N^m_k = 0$, $\Delta_{yk}$ in Equation (8) is unknown due to the unknown value of $P^m_{ky}$. To deal with missing classes, previous work [Zhang *et al.*, 2022] suggests that the gradient updates of missing classes should be constrained, which inspires us to set $\Delta_{yk}$ as a small value. The evaluation of the local probability matrix relies on averaging the probability outputs of samples, while the means of the probability outputs of the samples belonging to the classes with limited samples may deviate from the true means. To address this issue, we incorporate Input Mixup [Zhang *et al.*, 2018] into the training process. Specifically, within each local training epoch's minibatch, the data mixup technique is employed to increase the occurrence frequency of samples for the classes with limited samples.

During the model aggregation phase, local models with good adaptability should be assigned with large aggregation weights. For client $m$, $m \in [M]$, if the local model $\phi_{w_m}$

| Dataset | CIFAR-10 | | | CIFAR-100 | | | Tiny-ImageNet | | |
|---|---|---|---|---|---|---|---|---|---|
| NonIID ($\beta$) | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 |
| FedAvg | $76.29_{\pm 2.19}$ | $70.75_{\pm 1.93}$ | $62.67_{\pm 2.62}$ | $41.68_{\pm 0.26}$ | $39.72_{\pm 0.67}$ | $38.58_{\pm 0.16}$ | $45.66_{\pm 0.51}$ | $40.30_{\pm 0.60}$ | $36.52_{\pm 0.86}$ |
| FedProx | $75.99_{\pm 1.95}$ | $70.52_{\pm 2.10}$ | $62.27_{\pm 2.53}$ | $41.57_{\pm 0.20}$ | $39.67_{\pm 0.13}$ | $38.48_{\pm 0.22}$ | $45.39_{\pm 0.20}$ | $40.09_{\pm 0.14}$ | $36.18_{\pm 0.74}$ |
| FedNova | $75.89_{\pm 2.09}$ | $67.27_{\pm 5.83}$ | $55.44_{\pm 2.52}$ | $41.84_{\pm 0.09}$ | $39.67_{\pm 0.54}$ | $38.23_{\pm 0.17}$ | $45.71_{\pm 0.37}$ | $40.14_{\pm 0.40}$ | $36.19_{\pm 0.98}$ |
| CReFF | $76.27_{\pm 1.56}$ | $69.08_{\pm 2.74}$ | $60.49_{\pm 5.37}$ | $37.85_{\pm 0.16}$ | $37.96_{\pm 0.64}$ | $38.17_{\pm 0.45}$ | $44.77_{\pm 0.38}$ | $39.77_{\pm 0.89}$ | $35.65_{\pm 0.74}$ |
| FedROD | $77.71_{\pm 0.98}$ | $71.53_{\pm 1.86}$ | $62.84_{\pm 3.16}$ | $42.09_{\pm 0.23}$ | $40.12_{\pm 0.68}$ | $38.36_{\pm 0.33}$ | $46.02_{\pm 0.30}$ | $42.27_{\pm 0.58}$ | $38.00_{\pm 0.78}$ |
| FedNTD | $76.69_{\pm 1.26}$ | $67.92_{\pm 2.10}$ | $59.26_{\pm 4.13}$ | $43.60_{\pm 0.12}$ | $40.33_{\pm 0.78}$ | $38.70_{\pm 1.03}$ | $47.19_{\pm 0.12}$ | $41.56_{\pm 0.04}$ | $37.19_{\pm 0.48}$ |
| FedDecorr | $75.84_{\pm 2.04}$ | $70.74_{\pm 2.21}$ | $62.48_{\pm 2.52}$ | $42.15_{\pm 0.29}$ | $39.63_{\pm 0.62}$ | $38.26_{\pm 0.66}$ | $45.47_{\pm 0.26}$ | $40.13_{\pm 0.37}$ | $36.19_{\pm 0.52}$ |
| FedLC | $77.08_{\pm 1.47}$ | $70.61_{\pm 1.99}$ | $56.52_{\pm 5.13}$ | $41.67_{\pm 0.16}$ | $39.49_{\pm 0.79}$ | $35.58_{\pm 0.15}$ | $45.90_{\pm 0.21}$ | $41.32_{\pm 0.24}$ | $36.21_{\pm 0.25}$ |
| FedNH | $75.89_{\pm 1.57}$ | $69.70_{\pm 1.38}$ | $61.73_{\pm 3.12}$ | $44.54_{\pm 0.30}$ | $41.78_{\pm 0.60}$ | $39.70_{\pm 0.57}$ | $46.25_{\pm 0.19}$ | $43.06_{\pm 0.11}$ | $39.66_{\pm 0.27}$ |
| FedACD | $\mathbf{79.39_{\pm 0.85}}$ | $\mathbf{74.05_{\pm 3.00}}$ | $\mathbf{63.56_{\pm 2.88}}$ | $\mathbf{48.94_{\pm 0.16}}$ | $\mathbf{46.30_{\pm 0.38}}$ | $\mathbf{43.96_{\pm 0.67}}$ | $\mathbf{48.01_{\pm 0.23}}$ | $\mathbf{45.07_{\pm 0.54}}$ | $\mathbf{42.17_{\pm 0.69}}$ |

Table 2: Performance(%) of the global models on the test sets constructed based on the data distribution of each client.

has good adaptability, the constraints in Equation (6) should hold, implying that the probability matrix $P^m$ satisfies that $P^m_{ii} = P^m_{jj}$ for $i \neq j$. In this way, we can assign the aggregation weights by measuring the degree to which the constraints are satisfied. We set a template matrix $Q$, where $Q_{ii} = Q_{jj}$ for $i \neq j$, and measure the degree to which the constraints are satisfied by calculating the KL divergence between the local probability matrix $P^m$ and the template matrix $Q$. Intuitively, the ideal template matrix is the identity matrix $\mathbb{I}$, but we can not calculate the KL divergence between $P^m$ and $\mathbb{I}$. Hence, we define a template matrix $Q$ as that shown in Equation (10). For client $m$, we compute its aggregation score $V_m$ by calculating the KL divergence between the local probability matrix $P^m$ and the template matrix $Q$ as follows:

$$
\begin{aligned}
V_m &= \text{Sigmoid}\left(\frac{1}{\text{KL}(P^m || Q)}\right) \\
&= \text{Sigmoid}\left(\frac{1}{\sum_{i=1}^{C}\sum_{j=1}^{C} P^m_{ij}\log\left(\frac{P^m_{ij}}{Q_{ij}}\right)}\right), \\
Q &= \begin{bmatrix} \tau & \frac{1-\tau}{C-1} & \cdots & \frac{1-\tau}{C-1} \\ \frac{1-\tau}{C-1} & \tau & \cdots & \frac{1-\tau}{C-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-\tau}{C-1} & \frac{1-\tau}{C-1} & \cdots & \tau \end{bmatrix},
\end{aligned}
\tag{10}
$$

where $0 < \tau < 1$ is a parameter. Our **Fed**erated Learning with **A**daptability over **C**lient **D**istributions (FedACD) follows the rules below:

$$
\begin{aligned}
&\textbf{Local: } w^\star_m \leftarrow argmin_w\ \mathcal{L}_1 + \lambda\mathcal{L}_2, \text{initialized with } \overline{w}; \\
&\textbf{Global: } \overline{w} \leftarrow \sum_{m\in[M]} \frac{V_m}{\sum_{k\in[M]} V_k} w^\star_m.
\end{aligned}
\tag{11}
$$

After local training, client $m$ uploads the aggregation score $V_m$ to the server. It is noteworthy that $V_m$ is a scalar, and the server cannot infer the local probability matrix or any privacy-sensitive information from $V_m$, which effectively protects the privacy of the client data.

# 4 Experiments

## 4.1 Settings

**Datasets and Models.** We perform extensive experiments on three benchmark datasets: CIFAR-10, CIFAR-100 [Krizhevsky, 2009], and Tiny-ImageNet [Deng *et al.*, 2009; Le and Yang, 2015]. Tiny-ImageNet is a subset of ImageNet with 100k samples of 200 classes. Following [Chen and Chao, 2022], we adopt a simple Convolutional neural network for CIFAR-10 and CIFAR-100, while using Resnet18 [He *et al.*, 2016] for the Tiny-ImageNet. We implement all compared federated learning methods with the same model for a fair comparison.

**Client settings.** We employ Dirichlet sampling to generate Non-IID data for each client. Dirichlet sampling is a common technique used in FL for creating Non-IID data [Chen and Chao, 2022; Zhang *et al.*, 2022; Dai *et al.*, 2023]. It yields distinct label distributions for each client, with the degree of data heterogeneity controlled by $\beta$. Smaller $\beta$ refers to severer heterogeneity and when $\beta < 1$, some clients may lack samples of certain classes. In our experiments, we adopt $\beta \in \{0.3, 0.1, 0.05\}$.

**Hyper-parameters.** For the local training process, SGD optimizer is used with a 0.01 initial learning and 0.9 momentum. We employ a weight decay of $10^{-5}$ for CIFAR-10 and CIFAR-100, and $10^{-3}$ for TinyImageNet to mitigate overfitting. The batch size is set as 64. The number of clients is set as 20 and the participation ratio is set as $40\%$. The local training epoch is set as 5 and the total communication round is set as 200. $\lambda$ is set as 1 and $\tau$ is set as $1 - 10^{-5}$.

**Baselines.** We select three types of FL methods as the baselines. *1) Generic FL:* FedAvg [McMahan *et al.*, 2017]; *2) Classical FL with Non-IID data:* FedProx [Li *et al.*, 2020], FedNova [Wang *et al.*, 2020]; *3) FL methods most related to us:* FedROD [Chen and Chao, 2022], CReFF[Shang *et al.*, 2022], FedNTD[Lee *et al.*, 2022], FedLC [Zhang *et al.*, 2022], FedDecorr[Shi *et al.*, 2023], FedNH [Dai *et al.*, 2023].

## 4.2 Main results

We evaluate the performance of the global model on the test set with uniform data distribution (shown in Table 1) and the test sets constructed based on the data distribution of each

| Dataset | CIFAR-10 | | | CIFAR-100 | | | Tiny-ImageNet | | |
|---|---|---|---|---|---|---|---|---|---|
| NonIID ($\beta$) | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 |
| FedAvg | $0.41_{\pm0.033}$ | $0.57_{\pm0.019}$ | $0.70_{\pm0.020}$ | $0.69_{\pm0.006}$ | $0.77_{\pm0.006}$ | $0.82_{\pm0.001}$ | $0.68_{\pm0.004}$ | $0.79_{\pm0.006}$ | $0.86_{\pm0.004}$ |
| FedProx | $0.41_{\pm0.034}$ | $0.58_{\pm0.019}$ | $0.70_{\pm0.020}$ | $0.69_{\pm0.005}$ | $0.77_{\pm0.006}$ | $0.82_{\pm0.001}$ | $0.68_{\pm0.003}$ | $0.79_{\pm0.006}$ | $0.86_{\pm0.004}$ |
| FedNova | $0.41_{\pm0.032}$ | $0.56_{\pm0.030}$ | $0.69_{\pm0.030}$ | $0.69_{\pm0.005}$ | $0.77_{\pm0.007}$ | $0.82_{\pm0.002}$ | $0.68_{\pm0.005}$ | $0.79_{\pm0.006}$ | $0.86_{\pm0.003}$ |
| CReFF | $0.41_{\pm0.033}$ | $0.57_{\pm0.019}$ | $0.70_{\pm0.020}$ | $0.69_{\pm0.006}$ | $0.77_{\pm0.006}$ | $0.82_{\pm0.001}$ | $0.68_{\pm0.004}$ | $0.79_{\pm0.006}$ | $0.86_{\pm0.004}$ |
| FedROD | $0.30_{\pm0.014}$ | $0.38_{\pm0.020}$ | $0.47_{\pm0.031}$ | $0.64_{\pm0.001}$ | $0.70_{\pm0.005}$ | $0.78_{\pm0.003}$ | $0.66_{\pm0.002}$ | $0.76_{\pm0.002}$ | $0.83_{\pm0.004}$ |
| FedNTD | $0.31_{\pm0.030}$ | $0.46_{\pm0.028}$ | $0.59_{\pm0.030}$ | $0.65_{\pm0.001}$ | $0.71_{\pm0.004}$ | $0.79_{\pm0.010}$ | $0.66_{\pm0.002}$ | $0.77_{\pm0.001}$ | $0.84_{\pm0.003}$ |
| FedDecorr | $0.41_{\pm0.033}$ | $0.58_{\pm0.019}$ | $0.70_{\pm0.021}$ | $0.69_{\pm0.003}$ | $0.77_{\pm0.006}$ | $0.82_{\pm0.001}$ | $0.66_{\pm0.003}$ | $0.79_{\pm0.003}$ | $0.86_{\pm0.003}$ |
| FedLC | $0.30_{\pm0.026}$ | $0.39_{\pm0.023}$ | $0.49_{\pm0.062}$ | $0.64_{\pm0.002}$ | $0.72_{\pm0.004}$ | $0.83_{\pm0.001}$ | $0.67_{\pm0.004}$ | $0.77_{\pm0.005}$ | $0.87_{\pm0.002}$ |
| FedNH | $0.40_{\pm0.031}$ | $0.55_{\pm0.020}$ | $0.69_{\pm0.021}$ | $0.66_{\pm0.004}$ | $0.74_{\pm0.007}$ | $0.79_{\pm0.002}$ | $0.67_{\pm0.001}$ | $0.76_{\pm0.003}$ | $0.82_{\pm0.004}$ |
| FedACD | $\mathbf{0.25}_{\pm\mathbf{0.017}}$ | $\mathbf{0.31}_{\pm\mathbf{0.012}}$ | $\mathbf{0.39}_{\pm\mathbf{0.031}}$ | $\mathbf{0.60}_{\pm\mathbf{0.003}}$ | $\mathbf{0.67}_{\pm\mathbf{0.005}}$ | $\mathbf{0.74}_{\pm\mathbf{0.010}}$ | $\mathbf{0.64}_{\pm\mathbf{0.001}}$ | $\mathbf{0.73}_{\pm\mathbf{0.002}}$ | $\mathbf{0.80}_{\pm\mathbf{0.004}}$ |

Table 3: The average risk on the distributions over clients of local models.

| Client Num K | 20 | | | 40 | 60 | 80 | 100 |
|---|---|---|---|---|---|---|---|
| Participation Ratio $\gamma$ | 0.2 | 0.4 | 0.6 | 0.4 | 0.4 | 0.4 | 0.4 |
| FedAvg | $36.87_{\pm0.38}$ | $39.57_{\pm0.60}$ | $40.05_{\pm0.84}$ | $39.64_{\pm0.46}$ | $38.47_{\pm0.13}$ | $37.77_{\pm0.25}$ | $35.76_{\pm0.20}$ |
| CReFF | $38.60_{\pm0.52}$ | $37.71_{\pm0.59}$ | $36.40_{\pm0.45}$ | $31.28_{\pm0.50}$ | $26.50_{\pm0.23}$ | $23.26_{\pm0.18}$ | $20.86_{\pm0.31}$ |
| FedROD | $39.85_{\pm0.29}$ | $40.15_{\pm0.44}$ | $40.56_{\pm0.35}$ | $38.50_{\pm0.19}$ | $36.97_{\pm0.10}$ | $34.78_{\pm0.16}$ | $33.01_{\pm0.11}$ |
| FedNTD | $39.11_{\pm0.25}$ | $39.90_{\pm0.31}$ | $40.55_{\pm0.15}$ | $39.34_{\pm0.15}$ | $37.69_{\pm0.19}$ | $37.48_{\pm0.14}$ | $36.35_{\pm0.04}$ |
| FedDecorr | $37.80_{\pm0.61}$ | $38.89_{\pm0.19}$ | $39.87_{\pm0.15}$ | $39.09_{\pm0.18}$ | $38.34_{\pm0.25}$ | $37.82_{\pm0.21}$ | $35.62_{\pm0.19}$ |
| FedNH | $39.99_{\pm0.12}$ | $41.74_{\pm0.24}$ | $42.10_{\pm0.61}$ | $40.76_{\pm0.46}$ | $39.48_{\pm0.15}$ | $38.21_{\pm0.17}$ | $37.43_{\pm0.12}$ |
| FedACD | $\mathbf{44.42}_{\pm\mathbf{0.33}}$ | $\mathbf{46.24}_{\pm\mathbf{0.17}}$ | $\mathbf{46.70}_{\pm\mathbf{0.03}}$ | $\mathbf{44.27}_{\pm\mathbf{0.46}}$ | $\mathbf{42.49}_{\pm\mathbf{0.17}}$ | $\mathbf{41.30}_{\pm\mathbf{0.12}}$ | $\mathbf{40.33}_{\pm\mathbf{0.04}}$ |

Table 4: Ablation study for number of clients and participation ratio. All experiments are conducted on CIFAR-100 with Non-IID $\beta = 0.1$.

client (shown in Table 2). Our method achieves the best performance across three datasets with diverse data heterogeneity. To further demonstrate that our method improves the adaptability of local models, we also calculate the average risk on the distributions over clients of local models (shown in Section 4.1). The results show that the risk over client distributions of the local models in our method is consistently lower than that in the baseline methods with diverse data heterogeneity. This confirms that our method enhances the performance of the global model by improving the adaptability of local models. Classical FL methods with Non-IID data, such as FedProx and FedNova, show similar results to FedAvg, which indicates that it is hard to align the local models that are trained on data with different distributions. The classifier re-training method, CReFF, fails to demonstrate effectiveness under various settings, which could be attributed to the unreliability of synthesized features. The prototype-based method, FedNH, exhibits only marginal improvement in settings with severe data heterogeneity, which indicates that prototype learning is also impacted by data heterogeneity. FedDecorr focuses on representation learning with data heterogeneity, but also fails to achieve significant performance improvements. FedROD and FedLC demonstrate an improvement in accuracy compared to FedAvg on CIFAR-10, which indicates that the balanced loss helps alleviate the impact of data heterogeneity to some extent. However, the balanced loss fails to exhibit effectiveness when confronted with challenging scenarios, such as CIFAR-100 and TinyImageNet.

### 4.3 Ablation Study

**Different number of clients $K$ with various participation ratio $\gamma$.** We select the best baselines in Table 1 and conduct experiments on skewed CIFAR-100 under different number of clients $K$ with various participation ratio $\gamma$. As shown in Table 4, to validate the impact of client number, we set $\gamma = 0.4$ and $K \in \{20, 40, 60, 80, 100\}$. As the number of clients gradually increases, achieving convergence in FL becomes harder. Most methods experience a decline in accuracy, while our method consistently outperforms the baseline methods. To validate the impact of the participation ratio, we set $K = 20$ and $\gamma \in \{0.2, 0.4, 0.6\}$. When $\gamma$ is small, the client data distributions among different rounds vary significantly, leading to divergent gradient directions. However, our method consistently achieves the best performance. All experiments demonstrate that our method performs well under different numbers of clients with partial participation ratios.

**Input Mixup.** We incorporate Input Mixup into local training. For a fair comparison, we select several methods compatible with Input Mixup in Table 1 and incorporate Input Mixup into the training process. As shown in Table 5, our method without Input Mixup outperforms the baseline methods without Input Mixup under all settings. After incorporating Input Mixup, our method consistently outperforms the baseline methods with Input Mixup by a large margin under various data heterogeneity. Unlike other methods where Input Mixup mainly augments data diversity, we leverage it to improve the reliability of the evaluation process of the local

| Dataset | CIFAR-10 | | | CIFAR-100 | | | Tiny-ImageNet | | |
|---|---|---|---|---|---|---|---|---|---|
| NonIID ($\beta$) | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 | 0.3 | 0.1 | 0.05 |
| FedAvg w/o Input Mixup | $75.63_{\pm 0.75}$ | $68.35_{\pm 2.43}$ | $60.47_{\pm 3.74}$ | $41.97_{\pm 0.24}$ | $39.57_{\pm 0.60}$ | $38.12_{\pm 0.11}$ | $45.76_{\pm 0.21}$ | $40.24_{\pm 0.31}$ | $36.09_{\pm 0.33}$ |
| FedAvg w/ Input Mixup | $73.44_{\pm 2.20}$ | $66.13_{\pm 2.62}$ | $59.25_{\pm 2.89}$ | $46.43_{\pm 0.49}$ | $41.41_{\pm 0.54}$ | $37.17_{\pm 1.13}$ | $47.36_{\pm 0.36}$ | $41.61_{\pm 1.13}$ | $34.99_{\pm 0.43}$ |
| FedROD w/o Input Mixup | $77.53_{\pm 0.86}$ | $71.12_{\pm 1.33}$ | $62.46_{\pm 3.29}$ | $42.02_{\pm 0.15}$ | $40.15_{\pm 0.44}$ | $38.37_{\pm 0.18}$ | $46.18_{\pm 0.26}$ | $42.02_{\pm 0.14}$ | $37.81_{\pm 0.39}$ |
| FedROD w/ Input Mixup | $76.71_{\pm 0.18}$ | $71.70_{\pm 0.92}$ | $61.76_{\pm 0.14}$ | $42.38_{\pm 0.15}$ | $40.06_{\pm 0.64}$ | $38.13_{\pm 0.23}$ | $46.19_{\pm 0.17}$ | $41.83_{\pm 0.16}$ | $37.08_{\pm 0.11}$ |
| FedLC w/o Input Mixup | $76.76_{\pm 0.56}$ | $69.40_{\pm 1.40}$ | $52.71_{\pm 4.09}$ | $41.92_{\pm 0.39}$ | $39.85_{\pm 0.52}$ | $35.27_{\pm 0.16}$ | $46.66_{\pm 0.12}$ | $41.37_{\pm 0.14}$ | $36.63_{\pm 0.38}$ |
| FedLC w/ Input Mixup | $76.22_{\pm 0.41}$ | $69.82_{\pm 0.31}$ | $55.39_{\pm 0.58}$ | $45.37_{\pm 0.19}$ | $42.38_{\pm 0.42}$ | $38.01_{\pm 0.25}$ | $47.33_{\pm 0.13}$ | $43.22_{\pm 0.14}$ | $37.51_{\pm 0.31}$ |
| FedNH w/o Input Mixup | $75.30_{\pm 0.84}$ | $68.11_{\pm 2.15}$ | $60.54_{\pm 2.61}$ | $44.74_{\pm 0.14}$ | $41.74_{\pm 0.24}$ | $39.61_{\pm 0.43}$ | $45.09_{\pm 1.95}$ | $42.31_{\pm 0.85}$ | $38.87_{\pm 0.21}$ |
| FedNH w/ Input Mixup | $74.16_{\pm 1.73}$ | $66.87_{\pm 3.33}$ | $58.16_{\pm 2.50}$ | $46.80_{\pm 0.47}$ | $43.11_{\pm 0.29}$ | $40.58_{\pm 0.70}$ | $47.47_{\pm 0.44}$ | $42.62_{\pm 0.71}$ | $39.85_{\pm 0.37}$ |
| FedACD w/o Input Mixup | $77.97_{\pm 0.41}$ | $72.59_{\pm 0.38}$ | $62.64_{\pm 0.32}$ | $45.53_{\pm 0.30}$ | $42.61_{\pm 0.26}$ | $40.76_{\pm 0.24}$ | $46.94_{\pm 0.36}$ | $43.40_{\pm 0.38}$ | $39.67_{\pm 0.25}$ |
| FedACD w/ Input Mixup | $\mathbf{79.57_{\pm 0.02}}$ | $\mathbf{73.13_{\pm 0.52}}$ | $\mathbf{63.57_{\pm 0.46}}$ | $\mathbf{49.08_{\pm 0.08}}$ | $\mathbf{46.24_{\pm 0.17}}$ | $\mathbf{43.22_{\pm 0.46}}$ | $\mathbf{49.62_{\pm 0.32}}$ | $\mathbf{45.29_{\pm 0.14}}$ | $\mathbf{41.44_{\pm 0.85}}$ |

Table 5: Ablation study for Input Mixup.

| Local Epochs E | 3 | 5 | 7 | 9 | 11 | 13 | 15 |
|---|---|---|---|---|---|---|---|
| FedAvg | $40.76_{\pm 0.68}$ | $39.57_{\pm 0.60}$ | $38.89_{\pm 0.40}$ | $38.51_{\pm 0.13}$ | $37.81_{\pm 0.24}$ | $37.41_{\pm 0.21}$ | $37.32_{\pm 0.36}$ |
| CReFF | $36.93_{\pm 0.52}$ | $37.71_{\pm 0.59}$ | $38.38_{\pm 0.37}$ | $38.48_{\pm 0.28}$ | $38.23_{\pm 0.15}$ | $38.78_{\pm 0.22}$ | $38.24_{\pm 0.10}$ |
| FedROD | $41.72_{\pm 0.38}$ | $40.15_{\pm 0.44}$ | $39.23_{\pm 0.28}$ | $38.84_{\pm 0.28}$ | $37.28_{\pm 0.07}$ | $37.47_{\pm 0.09}$ | $38.00_{\pm 0.24}$ |
| FedNTD | $41.61_{\pm 0.33}$ | $39.90_{\pm 0.31}$ | $39.25_{\pm 0.22}$ | $39.42_{\pm 0.21}$ | $39.30_{\pm 0.33}$ | $39.03_{\pm 0.29}$ | $38.51_{\pm 0.18}$ |
| FedDecorr | $40.22_{\pm 0.44}$ | $38.89_{\pm 0.19}$ | $38.33_{\pm 0.28}$ | $38.18_{\pm 0.28}$ | $37.84_{\pm 0.29}$ | $36.87_{\pm 0.25}$ | $36.92_{\pm 0.33}$ |
| FedLC | $41.40_{\pm 0.15}$ | $39.85_{\pm 0.52}$ | $39.53_{\pm 0.60}$ | $38.68_{\pm 0.15}$ | $37.20_{\pm 0.23}$ | $37.63_{\pm 0.40}$ | $36.74_{\pm 0.17}$ |
| FedNH | $42.17_{\pm 0.55}$ | $41.74_{\pm 0.24}$ | $41.34_{\pm 0.42}$ | $39.62_{\pm 0.31}$ | $39.55_{\pm 0.30}$ | $39.93_{\pm 0.24}$ | $39.36_{\pm 0.32}$ |
| FedACD | $\mathbf{46.12_{\pm 0.39}}$ | $\mathbf{46.24_{\pm 0.17}}$ | $\mathbf{45.46_{\pm 0.01}}$ | $\mathbf{45.36_{\pm 0.25}}$ | $\mathbf{44.73_{\pm 0.35}}$ | $\mathbf{44.68_{\pm 0.29}}$ | $\mathbf{44.85_{\pm 0.14}}$ |

Table 6: Ablation study for different local epochs. All experiments are conducted on CIFAR-100 with Non-IID $\beta = 0.1$.

| $\lambda$ | 0.5 | 1 | 1.5 | 2 |
|---|---|---|---|---|
| FedACD | $46.05_{\pm 0.21}$ | $\mathbf{46.24_{\pm 0.17}}$ | $45.79_{\pm 0.18}$ | $45.35_{\pm 0.28}$ |

Table 7: Ablation study for parameter $\lambda$. All experiments are conducted on CIFAR-100 with Non-IID $\beta = 0.1$.

| NonIID($\beta$) | 0.3 | 0.1 | 0.05 |
|---|---|---|---|
| $\text{FedACD}_{\text{base}}$ | $48.63_{\pm 0.41}$ | $45.51_{\pm 0.21}$ | $42.76_{\pm 0.45}$ |
| FedACD | $\mathbf{49.08_{\pm 0.08}}$ | $\mathbf{46.24_{\pm 0.17}}$ | $\mathbf{43.22_{\pm 0.46}}$ |

Table 8: Ablation study for the aggregation method. All experiments are conducted on CIFAR-100 with Non-IID $\beta \in \{0.3, 0.1, 0.05\}$.

probability matrix by ensuring that the means of the probability outputs of samples belonging to classes with limited samples are closer to the true means, thus incorporating Input Mixup can improve the performance of our method.

**Different local epoch $E$.** We select the best baselines in Table 1 and vary the number of local training epoch $E \in \{3, 5, 7, 9, 11, 13, 15\}$ for each client in every round. As shown in Table 6, our method consistently achieves the best performance across different settings. Meanwhile, it is noteworthy that as the local training epoch $E$ increases, the performance of most methods exhibits a declining trend. However, our method maintains relatively stable performance for different local epochs, especially with a large $E \in \{9, 11, 13, 15\}$. The reason is that with the increase of local training epochs, the local models tend to overfit the local data distributions. This increases the risk of local models on heterogeneous data distributions over clients, i.e., the adaptability of local models becomes worse.

**Parameter $\lambda$.** Parameter $\lambda$ in Equation (9) controls the contribution of $\mathcal{L}_1$ and $\mathcal{L}_2$. We conduct experiments with $\lambda \in \{0.5, 1, 1, 5, 2.0\}$. As shown in Table 7, our method performs best with $\lambda$ = 1 and remains robust to variations of $\lambda$. Therefore, we set $\lambda = 1$ in all experiments.

**Aggregation method.** We propose an aggregation method that allows local models with good adaptability over client distributions to have large aggregation weights. We conduct experiments by comparing our aggregation method with the aggregation method with uniform weights (denoted as $\text{FedACD}_{\text{base}}$). As shown in Table 8, our aggregation method consistently outperforms $\text{FedACD}_{\text{base}}$ under various data heterogeneity, demonstrating its effectiveness.

## 5 Conclusion

In this paper, we introduce the adaptability of local models, i.e., the average performance of local models on data distributions over clients, and focus on improving the adaptability of local models to enhance the performance of the global model. Extensive experiments on federated learning benchmarks demonstrate that our method achieves the well-performed global model that outperforms the baseline methods.

## Acknowledgements

This work is supported by the National Science Foundation of China (62276125) and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

## References

[Acar *et al.*, 2021] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*, pages 790–826, 2021.

[Adnan *et al.*, 2022] Mohammed Adnan, Shivam Kalra, Jesse C Cresswell, Graham W Taylor, and Hamid R Tizhoosh. Federated learning and differential privacy for medical image analysis. *Scientific Reports*, 12(1):1953, 2022.

[Chen and Chao, 2022] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, pages 21579–21611, 2022.

[Dai *et al.*, 2023] Yutong Dai, Zeyuan Johnson Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI'23)*, pages 7314–7322, 2023.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 248–255, 2009.

[Duan *et al.*, 2021] Moming Duan, Duo Liu, Xianzhang Chen, Renping Liu, Yujuan Tan, and Liang Liang. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems*, 32(1):59–71, 2021.

[Hard *et al.*, 2020] Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, Ignacio Lopez Moreno, and Rajiv Mathews. Training keyword spotting models on non-iid data with federated learning. *arXiv*, abs/2005.10406, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pages 770–778, 2016.

[Jiang *et al.*, 2021] Ziyue Jiang, Yi Ren, Ming Lei, and Zhou Zhao. Fedspeech: Federated text-to-speech with continual learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence(IJCAI'21)*, pages 3829–3835, 8 2021.

[Kairouz *et al.*, 2019] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, and et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14:1–210, 2019.

[Kang *et al.*, 2020] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2):72–80, 2020.

[Karimireddy *et al.*, 2020] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, pages 5132–5143, 2020.

[Khan *et al.*, 2021] Latif U Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799, 2021.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, pages 32–33, 2009.

[Le and Yang, 2015] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[Lee *et al.*, 2022] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS'22)*, pages 38461–38474, 2022.

[Li *et al.*, 2019] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35:3347–3366, 2019.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of the Machine Learning and Systems (MLSys'20)*, pages 429–450, 2020.

[Li *et al.*, 2022] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *Proceedings of the IEEE 38th International Conference on Data Engineering (ICDE'22)*, pages 965–978, 2022.

[Luo *et al.*, 2021] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, pages 5972–5984, 2021.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS'17)*, pages 1273–1282, 2017.

[Menon *et al.*, 2021] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *Proceedings of the 9th International Con-*

*ference on Learning Representations (ICLR'21)*, pages 3360–3384, 2021.

[Shang *et al.*, 2022] Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. In *Proceedings of the 31th International Joint Conference on Artificial Intelligence (IJCAI'22)*, pages 2218–2224, 2022.

[Shi *et al.*, 2023] Yujun Shi, Jian Liang, Wenqing Zhang, Vincent Tan, and Song Bai. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*, pages 9820–9844, 2023.

[Wang *et al.*, 2020] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pages 7611–7623, 2020.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*, pages 2866–2879, 2018.

[Zhang *et al.*, 2022] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pages 26311–26329, 2022.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv*, abs/1806.00582, 2018.

[Zheng *et al.*, 2020] Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. Federated meta-learning for fraudulent credit card detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence(IJCAI'20)*, pages 4654–4660, 7 2020.