CalFAT: Calibrated Federated Adversarial Training with Label Skewness

Chen Chen[*]
Zhejiang University

Yuchen Liu
Zhejiang University

Xingjun Ma
Fudan University

Lingjuan Lyu[†]
Sony AI

Abstract

Recent studies have shown that, like traditional machine learning, federated learning (FL) is also vulnerable to adversarial attacks. To improve the adversarial robustness of FL, federated adversarial training (FAT) methods have been proposed to apply adversarial training locally before global aggregation. Although these methods demonstrate promising results on independent identically distributed (IID) data, they suffer from training instability on non-IID data with label skewness, resulting in degraded natural accuracy. This tends to hinder the application of FAT in real-world applications where the label distribution across the clients is often skewed. In this paper, we study the problem of FAT under label skewness, and reveal one root cause of the training instability and natural accuracy degradation issues: skewed labels lead to non-identical class probabilities and heterogeneous local models. We then propose a Calibrated FAT (CalFAT) approach to tackle the instability issue by calibrating the logits adaptively to balance the classes. We show both theoretically and empirically that the optimization of CalFAT leads to homogeneous local models across the clients and better convergence points.

1 Introduction

Federated learning (FL) is a privacy-aware learning paradigm that allows multiple participants (clients) to collaboratively train a global model without sharing their private data [23, 28, 43, 29]. In FL, each client follows the conventional machine learning procedure to train a local model on its own data and periodically uploads the local model updates to a central server for global aggregation. However, recent studies have shown that, like conventional machine learning, FL is also vulnerable to well-crafted adversarial examples [20, 47, 9, 45], i.e., at inference time, attackers can add small, human-perceptible adversarial perturbations to the test examples to fool the global model with high success rates. This raises security and reliability concerns on the implementation of FL in real-world scenarios where such a vulnerability could cause heavy losses [38]. For example, for cross-silo FL in the biomedical domain, a vulnerable global model may cause misdiagnosis, wrong medical treatments, or even the loss of lives. Similarly, in financial-based cross-silo FL, the lack of adversarial robustness may lead to huge financial losses. It is thus imperative to develop a robust FL method that can train adversarially robust global models resistant to different types of adversarial attacks.

In conventional machine learning, adversarial training (AT) has been shown to be one of the most effective defenses against adversarial attacks [22, 40, 4]. Since the local training in FL is the same as conventional machine learning, recent works [47, 9, 45] proposed to perform local AT to

[*]Work done during internship at Sony AI.
[†]Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

improve the adversarial robustness of the global model. These methods in general are known as Federated Adversarial Training (FAT). AT has been found to be *more* challenging than standard training [3, 44, 42, 41, 4], as it generally requires more training data and larger-capacity models. Moreover, adversarial robustness may even be at odds with accuracy [30], meaning that the increase of robustness may inevitably decrease the natural accuracy (i.e., accuracy on natural test data). As a result, the natural accuracy of AT is much lower than standard training [5]. This phenomenon also exists in FL, i.e., FAT exhibits slower convergence and lower natural accuracy than standard FL, as mentioned by recent studies [47, 9].

Arguably, FAT will become more challenging if the data are non-independent and identically distributed (non-IID) across the clients. One typical non-IID setting that commonly exists in real-world applications is skewed label distribution [17], where different clients have different label distributions. In this paper, we study the problem of FAT on non-IID data with a particular focus on the challenging skewed label distribution setting (formally defined in Section 3.1). Under conventional training, Xu et al. [37] have shown that adversarially trained models introduce severe performance disparity across different classes. And such a disparity will be exacerbated under label skewness, ending up with much worse performance on the minority classes [33].

By far, only a few works have studied non-IID FAT in the current literature. Zizzo et al. [47] propose to perform AT on only part of the local data for better convergence, while standard training is applied to the rest of the local data. We term this method as MixFAT. Another relevant work called FedRBN [9] tackles a different problem: how to propagate federated robustness to low-resource clients. Although MixFAT and FedRBN have demonstrated promising results, they suffer from training instability and low natural accuracy issues when compared to standard FL, as we show in Figure 1. We also compare with the other four FAT baselines adapted from existing AT methods to FL, i.e., FedPGD, FedTRADES, FedMART, and FedGAIRAT. Unfortunately, these methods also exhibit slow convergence and much degraded final accuracy (details can be found in Section 4.1). This motivates us to propose a novel method called *Calibrated Federated Adversarial Training* (CalFAT) for effective FAT on non-IID data with skewed label distribution. CalFAT tackles the training instability issue by calibrating the logits to give higher scores to the minority classes.

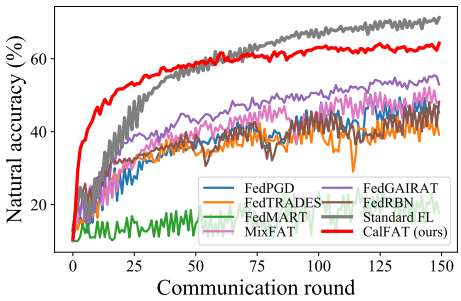

Figure 1: Natural accuracy and convergence of standard FL, our CalFAT, and 6 FAT baselines (FedPGD, FedTRADES, FedMART, MixFAT [47], FedGAIRAT and FedRBN [9]) under skewed label distribution with $\beta = 0.1$ (see Section 4).

In summary, our main contributions are:

- *New insight:* We study the problem of FAT on non-IID data with skewed label distribution, and reveal one root cause of the training instability and natural accuracy degradation: skewed labels lead to non-identical class probabilities and heterogeneous local models.
- *Novel method:* We propose a novel method called CalFAT for FAT with label skewness, and show that the optimization of CalFAT can lead to homogeneous local models, and consequently, stable training, faster convergence, and better final performance.
- *High effectiveness*: Extensive experiments on 4 benchmark vision datasets across various settings prove the effectiveness of our CalFAT and its superiority over existing FAT methods.

2 Notation and Preliminaries

2.1 Notation

Suppose there are m clients in FL with i denoting the i-th client, e.g., $\mathcal{D}_i$ denotes the local data of client i and θ_i denotes the parameters of its local model. We use $\hat{\theta}$ to denote the parameters of the global model. Subscript j is the sample index, e.g., (x_{ij}, y_{ij}) denotes the j-th sample of client i and its corresponding label with $y_{ij} \in \{1, \cdots, C\}$. Let $f_\theta(\cdot)$ be the local model $f(\cdot)$ (before softmax)

with parameter θ. Superscript l is the class index, e.g., $f^l(\cdot)$ denotes the logit output for class l. We denote the adversarial example of clean sample x by $\widetilde{x}$. $[m]$ denotes the integer set $\{1, \cdots, m\}$. $p_i(x, y)$ denotes the joint distribution of input x and label y at client i, and accordingly, $p_i(y)$ is the marginal distribution of label y, $p_i(y \mid x)$ is the conditional distribution of label y given input x and $p_i(x \mid y)$ is the conditional distribution of input x given label y.

2.2 Centralized Adversarial Training

Let $\mathcal{D} = \{x_j, y_j\}_{j=1}^n$ be the training dataset with n samples. The cross-entropy loss $\ell_{ce}(f_\theta(x), y)$ for an input-label pair (x, y) is defined as $\ell_{ce}(f_\theta(x), y) = -\log \sigma^y(f_\theta(x))$, where $\sigma^y(f) = \exp(f^y)/\sum_{l=1}^C \exp(f^l)$ is the softmax function, C is the number of classes, and f^l is the model output for class l. The objective function of the centralized adversarial training (AT) [22] can then be defined as $\min_\theta \sum_{j=1}^n \ell_{ce}(f_\theta(\widetilde{x}_j), y_j)/n$, where the adversarial example $\widetilde{x}_j$ can be generated by

$$\widetilde{x}_j = \underset{x'_j \in \mathcal{B}_\epsilon(x_j)}{\arg\max}\, \ell_{ce}(f_\theta(x'_j), y_j), \tag{1}$$

where $\mathcal{B}_\epsilon(x_j) = \{x' \mid \|x' - x_j\|_\infty < \epsilon\}$ is the closed ball of radius $\epsilon > 0$ centered at x_j, $\|\cdot\|_\infty$ is the L_∞ norm, and $\widetilde{x}_j$ is the most adversarial sample within the ϵ-ball.

A standard centralized AT method uses Projected Gradient Decent (PGD) to generate adversarial examples [22]. In particular, PGD iteratively generates an adversarial example $\widetilde{x}_j$ as follows:

$$x_j^{(k+1)} = \Pi_{\mathcal{B}_\epsilon(x_j^{(0)})}\left(x_j^{(k)} + \alpha\, \mathrm{sign}(\nabla_x \ell_{ce}(f_\theta(x_j^{(k)}), y_j))\right), \quad k = 0, \cdots, K-1, \tag{2}$$

where k is the step number, K is the total number of steps (i.e., $\widetilde{x}_j = x_j^{(K)}$), $\alpha > 0$ is the step size, $x_j^{(0)}$ is the natural sample, $x_j^{(k)}$ is the adversarial example generated at step k, $\Pi_{\mathcal{B}_\epsilon(x_j^{(0)})}$ is the projection function that projects the adversarial data onto the ϵ-ball centered at $x_j^{(0)}$, and $\mathrm{sign}(\cdot)$ is the sign function.

By optimizing the model parameters on adversarial examples generated by PGD, centralized AT is able to train a model that is robust against adversarial attacks.

2.3 Federated Adversarial Training

The concept of federated adversarial training (FAT) was first introduced in [47] (i.e., MixFAT) to deal with the adversarial vulnerability of FL. MixFAT applies AT locally to improve the robustness of the global model. Suppose there are m clients and each client i has its local data $\mathcal{D}_i = \{x_{ij}, y_{ij}\}_{j=1}^{n_i}$ sampled from distribution $p_i(x, y)$ with $n_i = |\mathcal{D}_i|$ being the size of the local data. In MixFAT, each client i optimizes its local model by minimizing the following objective:

$$\min_{\theta_i} \frac{1}{n_i}\Big(\sum_{j=1}^{n'_i} \ell_{ce}(f_{\theta_i}(\widetilde{x}_{ij}), y_{ij}) + \sum_{j=n'_i+1}^{n_i} \ell_{ce}(f_{\theta_i}(x_{ij}), y_{ij})\Big), \tag{3}$$

where $\widetilde{x}_{ij}$ is the PGD adversarial example of x_{ij}, n'_i is a hyperparameter that controls the ratio of data for AT, and θ_i are the local model parameters. After training the local model for certain epochs, client i uploads its local model parameters θ_i to the central server for aggregation. Note that MixFAT only applies AT to a proportion of the local data, mainly for convergence and stability considerations.

3 Calibrated Federated Adversarial Training (CalFAT)

3.1 Skewed Label Distribution Leads to Non-identical Class Probabilities

In this paper, we focus on one representative non-IID setting: *skewed label distribution* [19, 10], which is defined as follows.

Definition 1 (Skewed label distribution)**.** The label distribution across the clients is skewed, if for all $i \neq u$ and $i, u \in [m]$:

(a) there exists $y \in [C]$ such that $p_i(y) \neq p_u(y)$ and (b) $p_i(x \mid y) = p_u(x \mid y)$ for all x, y.

Condition (b) is to assume that, given a class y, x is sampled with equal probability at different clients. Note that there exist different types of non-IID: label skew, non-identical class conditional, quantity skew, to name a few (Appendix K in [10]). The class conditional is often assumed to be identical (i.e., condition (b)) when studying the label skewness problem, which is the main focus of this work. When condition (b) does not hold, it becomes the non-identical class conditional problem.

Lemma 1 (Non-identical class probabilities)**.** *If the label distribution across the clients is skewed and the class conditionals have the same support, then the class probabilities $\{p_i(y \mid x) \mid i \in [m]\}$ are non-identical, i.e., for all $i \neq u$ and $i, u \in [m]$, there exist x, y such that $p_i(y \mid x) \neq p_u(y \mid x)$.*

Lemma 1 implies that skewed label distribution gives rise to non-identical class probabilities $\{p_i(y \mid x) \mid i \in [m]\}$. The proof of Lemma 1 is given in Appendix A.

3.2 Standard Cross-entropy Leads to Heterogeneity

From a statistical point of view, each client i in previous FAT methods estimates its local class probability $p_i(y \mid x)$ during local training [7]. More specifically, they assume that $p_i(y \mid x)$ can be parameterized by θ_i^* as:

$$p_i(y \mid x) = \hat{p}(y \mid x; \theta_i^*) = \sigma^y(f_{\theta_i^*}(x)), \tag{4}$$

where θ_i^* is the ground-truth parameters of the local class probability $p_i(y \mid x)$. According to Lemma 1, the class probabilities $\{p_i(y \mid x)\}$ are non-identical when there is a skewed label distribution. Therefore, the ground-truth parameters $\{\theta_i^* \mid i \in [m]\}$ are heterogeneous. We use the sample variance of the ground-truth parameters to measure such heterogeneity as follows:

$$(s^*)^2 = V(\theta_1^*, \ldots, \theta_m^*) = \frac{1}{m-1} \sum_{i=1}^{m} \|\theta_i^* - \frac{1}{m} \sum_{j=1}^{m} \theta_j^*\|^2. \tag{5}$$

Each client i updates its local model parameters θ_i by optimizing the standard cross-entropy (CE) loss. The updated θ_i is the maximum likelihood estimate [1] of the ground-truth parameter θ_i^* [7]. Similarly, we use the sample variance [1] of the local model parameters to measure the heterogeneity of the local models:

$$s^2 = V(\theta_1, \ldots, \theta_m). \tag{6}$$

Larger sample variance implies higher model heterogeneity.

The following proposition suggests that the heterogeneity of local models originates from the heterogeneity of the local class probabilities.

Proposition 1 (Heterogeneous local models)**.** *Assume the label distribution across the clients is skewed. Let θ_i be the maximum likelihood estimate of θ_i^* in Eq. (4) given local data at client i. Then s^2 converges almost surely to a nonzero constant:*

$$s^2 \xrightarrow{a.s.} (s^*)^2 \neq 0, \tag{7}$$

where $\xrightarrow{a.s.}$ represents the almost sure convergence.

The proof of Proposition 1 is provided in Appendix B. $(s^*)^2$ measures the heterogeneity of the ground-truth parameters $\{\theta_i^* \mid i \in [m]\}$, which reflects the class probability difference across the clients as shown in Eq. (4).

Proposition 1 implies that the local models in previous FAT methods are heterogeneous when the label distribution across the clients is skewed. Since the local models are heterogeneous, aggregating these models tends to hurt the convergence and cause the divergence of the global model [17]. As shown in Figure 1, the training process of existing FAT methods is unstable and has much lower natural accuracy than the standard FL.

3.3 Learning Homogeneous Local Models by Calibration

Motivated by [24], we propose to re-parameterize the class probabilities. According to Bayes' formula [12],

$$p_i(y \mid x) = \frac{p_i(x \mid y) p_i(y)}{\sum_{l=1}^{C} p_i(x \mid l) p_i(l)}. \tag{8}$$

Algorithm 1 Local training of CalFAT

Input: Client i, global model parameters $\hat{\theta}$, local dataset $\mathcal{D}_i$, and local epoch number E

procedure CLIENTUPDATE
  $\theta_i \leftarrow \hat{\theta}$
  Compute π_i with $\mathcal{D}_i$ by Eq. (10)
  for local epoch=$1, \cdots, E$ **do**
    for $j = 1, \cdots, n_i$ **do**
      Sample (x_{ij}, y_{ij}) from $\mathcal{D}_i$
      Generate adversarial example $\widetilde{x}_{ij}$ by Eq. (14)
    end for
    $\theta_i \leftarrow \theta_i - \eta \frac{1}{n_i} \sum_{j=1}^{n_i} \nabla_{\theta_i} \ell_{cce}(f_{\theta_i}(\widetilde{x}_{ij}), y_{ij}, \pi_i)$
  end for
  return θ_i
end procedure

On the right-hand side of the above equation: (1) the class priors can be easily computed by the relative frequencies [1]; and (2) more importantly, the class conditionals $\{p_i(x \mid y) \mid i \in [m]\}$ are *identical* across different clients (see Definition 1).

Inspired by the above observation, we propose an alternative parameterization of $p_i(y \mid x)$. Assume that for all $i \in [m]$, the class conditional $p_i(x \mid y)$ can be parameterized by θ^* as $p_i(x \mid y) = \hat{q}(x \mid y; \theta^*)$, where $\hat{q}(x \mid y; \theta^*)$ can be an arbitrary conditional probability function. Then, $p_i(y \mid x)$ can be re-parameterized by θ^* as follows:

$$p_i(y \mid x) = \hat{q}_i(y \mid x; \theta^*) = \frac{\hat{q}(x \mid y; \theta^*)\pi_i^y}{\sum_{l=1}^{C} \hat{q}(x \mid l; \theta^*)\pi_i^l}. \tag{9}$$

where

$$\pi_i^y = n_i^y / n_i + \delta, y \in [C]. \tag{10}$$

Here π_i^y approximates the class prior $p_i(y)$, n_i^y is the sample size of class y on client i and $\delta > 0$ is a small constant added for numerical stability purpose. During local updates, client i uses its local data to update θ_i, which makes θ_i the maximum likelihood estimate of θ^*. The entire training procedure of our method is described in Section 3.4.

The following proposition suggests that the local models are homogeneous when trained with the above re-parameterization. The proof of Proposition 2 is provided in Appendix C.

Proposition 2 (Homogeneous local models)**.** *Assume the label distribution across the clients is skewed. Let θ_i be the maximum likelihood estimate of θ^* in Eq. (9) given local data at client i. Then s^2 converges almost surely to zero:*

$$s^2 \xrightarrow{a.s.} 0. \tag{11}$$

3.4 Details of CalFAT

The local training procedure of our proposed CalFAT is described in Algorithm 1. Specifically, we define $\hat{q}(x \mid y; \theta^*) = \exp\left(f_{\theta^*}^y(x)\right)$. Then, we maximize the likelihood of $\hat{q}_i(y \mid x; \theta^*)$ for each client i, which is equivalent to minimizing the following objective:

$$\min_{\theta_i} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell_{cce}(f_{\theta_i}(\widetilde{x}_{ij}), y_{ij}, \pi_i), \tag{12}$$

where $\ell_{cce}(\cdot, \cdot, \cdot)$ is the calibrated cross-entropy (CCE) loss and $\widetilde{x}_{ij}$ is the adversarial example of x_{ij}. The CCE loss is defined as:

$$\ell_{cce}(f_{\theta_i}(\widetilde{x}_{ij}), y_{ij}, \pi_i) = -\log \sigma^{y_{ij}}(f_{\theta_i}(\widetilde{x}_{ij}) + \log \pi_i). \tag{13}$$

As discussed in Section 3.3, minimizing the above CCE loss mitigates the heterogeneity of the local models, which can lead to improved convergence and performance of the global model.

Table 1: Natural and robust accuracy (%) on different datasets. The best results are in **bold**.

Dataset	CIFAR10						CIFAR100					
Metric	Natural	FGSM	BIM	CW	PGD-20	AA	Natural	FGSM	BIM	CW	PGD-20	AA
MixFAT	53.35 ± 0.11	29.14 ± 0.10	26.31 ± 0.17	22.79 ± 0.12	26.27 ± 0.11	21.89 ± 0.13	34.43 ± 0.13	15.69 ± 0.13	14.60 ± 0.14	11.31 ± 0.17	14.36 ± 0.20	9.06 ± 0.11
FedPGD	46.96 ± 0.16	28.70 ± 0.19	26.59 ± 0.18	24.38 ± 0.17	26.74 ± 0.18	22.47 ± 0.11	33.96 ± 0.14	16.07 ± 0.08	14.68 ± 0.10	11.67 ± 0.10	14.67 ± 0.15	10.87 ± 0.12
FedTRADES	46.06 ± 0.12	27.75 ± 0.17	26.32 ± 0.09	22.86 ± 0.10	26.31 ± 0.12	21.70 ± 0.09	29.55 ± 0.10	15.01 ± 0.06	14.11 ± 0.11	10.58 ± 0.03	14.30 ± 0.13	9.53 ± 0.09
FedMART	25.67 ± 0.21	18.50 ± 0.18	18.21 ± 0.22	15.22 ± 0.17	18.10 ± 0.22	14.41 ± 0.20	19.96 ± 0.17	13.00 ± 0.19	12.91 ± 0.14	9.92 ± 0.21	12.83 ± 0.18	8.57 ± 0.14
FedGAIRAT	48.42 ± 0.08	29.30 ± 0.09	26.55 ± 0.07	22.78 ± 0.12	27.20 ± 0.08	21.96 ± 0.07	34.92 ± 0.05	16.18 ± 0.06	15.37 ± 0.10	11.80 ± 0.05	14.90 ± 0.03	9.41 ± 0.05
FedRBN	47.80 ± 0.06	26.87 ± 0.07	26.25 ± 0.03	22.00 ± 0.01	26.30 ± 0.09	21.33 ± 0.09	28.55 ± 0.07	14.69 ± 0.04	13.41 ± 0.08	9.71 ± 0.08	14.15 ± 0.12	8.83 ± 0.08
CalFAT (ours)	**64.69** ± 0.08	**35.03** ± 0.12	**31.50** ± 0.07	**24.69** ± 0.11	**31.12** ± 0.11	**22.91** ± 0.08	**44.57** ± 0.10	**17.63** ± 0.10	**15.60** ± 0.11	**12.01** ± 0.11	**15.21** ± 0.07	**11.49** ± 0.08
Dataset	SVHN						ImageNet subset					
Metric	Natural	FGSM	BIM	CW	PGD-20	AA	Natural	FGSM	BIM	CW	PGD-20	AA
MixFAT	19.57 ± 0.10	19.61 ± 0.12	19.66 ± 0.12	19.66 ± 0.11	19.75 ± 0.11	14.80 ± 0.07	33.53 ± 0.06	19.47 ± 0.02	18.48 ± 0.10	16.15 ± 0.07	18.39 ± 0.02	11.98 ± 0.06
FedPGD	19.55 ± 0.08	19.33 ± 0.09	19.37 ± 0.08	19.68 ± 0.05	19.52 ± 0.09	13.64 ± 0.10	30.87 ± 0.12	18.88 ± 0.13	17.95 ± 0.10	16.07 ± 0.11	18.40 ± 0.16	11.34 ± 0.08
FedTRADES	56.96 ± 0.13	36.92 ± 0.13	35.15 ± 0.05	31.08 ± 0.14	34.90 ± 0.15	30.37 ± 0.11	30.22 ± 0.14	18.67 ± 0.13	17.99 ± 0.21	16.23 ± 0.13	17.82 ± 0.12	11.81 ± 0.10
FedMART	19.85 ± 0.16	19.94 ± 0.16	19.71 ± 0.16	19.85 ± 0.15	19.79 ± 0.17	14.64 ± 0.14	26.47 ± 0.18	16.40 ± 0.18	15.53 ± 0.17	14.43 ± 0.13	15.40 ± 0.21	9.34 ± 0.14
FedGAIRAT	58.41 ± 0.11	38.30 ± 0.12	36.52 ± 0.09	31.24 ± 0.15	36.69 ± 0.13	31.63 ± 0.06	34.25 ± 0.12	19.62 ± 0.10	19.28 ± 0.14	16.78 ± 0.14	19.18 ± 0.12	11.80 ± 0.09
FedRBN	53.88 ± 0.04	34.48 ± 0.08	32.52 ± 0.02	27.99 ± 0.02	32.32 ± 0.03	28.35 ± 0.05	29.35 ± 0.09	18.76 ± 0.03	17.25 ± 0.12	15.07 ± 0.10	18.05 ± 0.09	11.42 ± 0.12
CalFAT (ours)	**84.15** ± 0.07	**48.38** ± 0.11	**42.04** ± 0.07	**31.66** ± 0.04	**41.68** ± 0.11	**32.57** ± 0.10	**49.89** ± 0.11	**22.31** ± 0.17	**19.99** ± 0.09	**17.42** ± 0.12	**19.97** ± 0.14	**12.30** ± 0.07

In previous FAT methods, heterogeneous local models tend to give higher scores to the majority classes while lower scores to the minority classes. By contrast, our CalFAT encourages local models to give higher scores to the minority classes by adding a class-wise prior $\log \pi_i^l$ to the logits. Also different from MixFAT that trains the local models on both natural and adversarial data, our CalFAT trains the local models only on adversarial examples. Extensive empirical experiments are conducted in Section 4.1 to show the impact of using only adversarial data for optimization.

Adversarial example generation. Inspired by [40], we generate the adversarial examples by maximizing the following calibrated Kullback–Leibler (CKL) divergence loss:

$$\widetilde{x}_{ij} = \underset{x'_{ij} \in \mathcal{B}_\epsilon(x_{ij})}{\arg\max} \ \ell_{ckl}(f_{\theta_i}(x'_{ij}), f_{\theta_i}(x_{ij}), \pi_i), \tag{14}$$

where $\ell_{ckl}(\cdot,\cdot,\cdot)$ is the CKL loss defined as:

$$\ell_{ckl}(f_{\theta_i}(x'_{ij}), f_{\theta_i}(x_{ij}), \pi_i) = -\sum_{y=1}^{C} \sigma^y(f_{\theta_i}(x_{ij}) + \log \pi_i) \log \sigma^y(f_{\theta_i}(x'_{ij}) + \log \pi_i), \tag{15}$$

where $\log \pi_i$ is the same as in our CCE loss. Following centralized AT [22], we also use PGD to solve Eq. (14).

After training the local model for certain epochs following the above procedure, each client i uploads the model parameters θ_i to the server for aggregation. To be consistent with the most recent FAT methods [47, 9], we adopt the most widely used FedAvg [23] as the default aggregation framework. Our method is compatible with other FL frameworks (e.g., FedProx [16] and Scaffold [11]), as we will show in Section 4.1.

4 Experiments

Data configurations. Our experiments are conducted on 4 real-world datasets: CIFAR10 [13], CIFAR100 [13], SVHN [25], and ImageNet subset [6]. To simulate label skewness, we sample $p_i^l \sim Dir(\beta)$ and allocate a p_i^l proportion of the data of label l to client i, where $Dir(\beta)$ is the Dirichlet distribution with a concentration parameter β [39]. By default, we set $\beta = 0.1$ to simulate a highly skewed label distribution that widely exists in reality.

Baselines. We compare our proposed CalFAT with two state-of-the-art FAT methods: MixFAT [47] and FedRBN [9]. We also investigate the combination of the state-of-the-art centralized AT methods with FL, i.e., we apply standard PGD [22], TRADES [40], MART [34]), and GAIRAT [44] to FL, and term them as FedPGD, FedTRADES, FedMART, and FedGAIRAT, respectively.

Evaluation metrics. We report the natural test accuracy (Natural) and robust test accuracy under the most representative attacks, i.e., FGSM [35], BIM [15], PGD-20 [22], CW [2], and AA [5]. We run the experiment for 5 times and report the mean and standard deviation. More detailed experimental setup is provided in Appendix D.1.

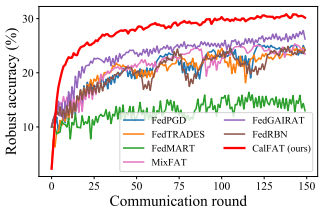

Figure 2: Robust accuracy (against PGD-20 attack) of different methods on CIFAR10 dataset.

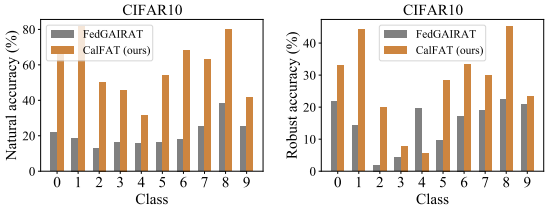

Figure 3: Per-class natural accuracy and robust accuracy (against PGD-20 attack) of CalFAT and the best baseline (FedGAIRAT) on CIFAR10 dataset.

4.1 Main Results

Evaluation on different datasets. Table 1 shows the results of all methods on CIFAR10, CIFAR100, SVHN, and ImageNet subset. From the table, we can observe that:

(1) Our CalFAT achieves the best robustness on all datasets, validating the efficacy of our CalFAT. For example, CalFAT outperforms the best baseline method (FedGAIRAT) by 10.20% on SVHN dataset under FGSM attack.

(2) Our CalFAT shows a significant improvement in natural accuracy compared to other baselines. For example, CalFAT can improve the natural accuracy of the best baseline method (FedGAIRAT) by 25.63% on SVHN dataset. We hypothesise that the reason lies in the homogeneity of local models in our CalFAT, which leads to better convergence and higher clean accuracy.

(3) All methods demonstrate the worst performance on CIFAR100 and ImageNet subset datasets. We conjecture that this is because there are more classes in these two datasets, making federated training substantially harder. Nevertheless, our CalFAT still achieves the best performance.

Learning curves of different methods. To visually compare our CalFAT with all the baselines, we plot the learning curves (i.e., performance across different communication rounds) of all methods in Figure 1 and Figure 2. As can be observed, CalFAT achieves the best natural accuracy and robust accuracy across almost the entire training process, which indicates that the design of our CalFAT is profitable for different federated learning stages. Moreover, our CalFAT is fairly stable during the whole training process while the accuracy curves of other baselines oscillate strongly. Such oscillations lead to bad convergence and low performance. We hypothesize that the heterogeneity of local models in the baseline methods is the main cause of the unstable training.

Table 2: Combining FAT methods with different losses.

Metric	Natural	PGD-20
MixFAT	53.35 ± 0.11	26.27 ± 0.11
MixFAT + LogitAdj	57.53 ± 0.21	27.65 ± 0.16
MixFAT + RoBal	58.25 ± 0.13	27.86 ± 0.10
MixFAT + Calibration (ours)	**60.23** ± 0.19	**28.67** ± 0.14
FedPGD	46.96 ± 0.16	26.74 ± 0.18
FedPGD + LogitAdj	59.79 ± 0.15	28.84 ± 0.12
FedPGD + RoBal	61.48 ± 0.07	29.51 ± 0.07
FedPGD + Calibration (ours)	**63.91** ± 0.13	**30.72** ± 0.16
FedTRADES	46.06 ± 0.12	26.31 ± 0.12
FedTRADES + LogitAdj	58.26 ± 0.20	27.92 ± 0.19
FedTRADES + RoBal	59.25 ± 0.23	28.63 ± 0.08
FedTRADES + Calibration (ours)	**63.12** ± 0.10	**30.27** ± 0.23
FedMART	25.67 ± 0.21	18.10 ± 0.22
FedMART + LogitAdj	42.01 ± 0.10	24.92 ± 0.02
FedMART + RoBal	44.26 ± 0.22	25.57 ± 0.17
FedMART + Calibration (ours)	**48.85** ± 0.08	**27.19** ± 0.11
CalFAT (ours)	**64.69** ± 0.08	**31.12** ± 0.11

Combining calibration loss with other FAT methods. To further show the effectiveness of our calibration loss, we combine it with four FAT methods (MixFAT, FedPGD, FedTRADES, and FedMART) and name them MixFAT + Calibration, FedPGD + Calibration, FedTRADES + Calibration, and FedMART + Calibration, respectively. We compare these calibration loss-based FAT methods with their original versions in Table 2. It is evident that, by introducing our calibration loss into their objectives, all FAT methods can be improved. These results confirm the importance of class calibration for FAT with label skewness.

Table 3: Natural and robust accuracy (%) across different FL frameworks on CIFAR10 dataset.

FL framework	FedProx			Scaffold		
Metric	Natural	PGD-20	AA	Natural	PGD-20	AA
MixFAT	53.75 ± 0.16	29.61 ± 0.19	21.59 ± 0.27	55.27 ± 0.20	28.78 ± 0.15	21.26 ± 0.11
FedPGD	49.57 ± 0.18	28.48 ± 0.17	21.31 ± 0.18	49.52 ± 0.14	27.46 ± 0.21	20.27 ± 0.15
FedTRADES	48.14 ± 0.20	27.75 ± 0.17	21.13 ± 0.21	47.78 ± 0.23	27.31 ± 0.16	20.04 ± 0.16
FedMART	28.32 ± 0.22	19.32 ± 0.23	15.91 ± 0.25	27.80 ± 0.17	20.03 ± 0.26	16.85 ± 0.15
FedGAIRAT	49.61 ± 0.20	29.34 ± 0.11	21.33 ± 0.18	49.54 ± 0.21	27.23 ± 0.25	20.16 ± 0.09
FedRBN	47.26 ± 0.13	26.63 ± 0.15	20.46 ± 0.06	49.77 ± 0.09	28.37 ± 0.12	20.32 ± 0.06
CalFAT	**66.32** ± 0.08	**32.79** ± 0.13	**22.83** ± 0.11	**67.16** ± 0.06	**32.94** ± 0.06	**21.94** ± 0.05

Comparison with state-of-the-art long-tail learning methods. We also compare our calibration loss with the losses used by long-tail learning methods LogitAdj [24] and RoBal [36]. In particular, we combine four FAT methods (MixFAT, FedPGD, FedTRADES, and FedMART) with the above three losses (LogitAdj loss, RoBal loss, and our calibration loss), and train the models following the same default setting. As shown in Table 2, both LogitAdj-based methods and RoBal-based methods have lower natural and robust accuracies than calibration loss-based methods. This indicates that our calibration loss is more suitable for FAT than other long-tail learning losses.

4.2 Performance on Different Classes

We further compare the per-class performance of our CalFAT with the best baseline FedGAIRAT. First, we use a well-trained model to initialize a global model. Second, the global model distributes the model parameter to all clients. Third, the local clients train their local models with their local data for 1 epoch. Then, we report the per-class average performance of all clients for each class. For fair comparison, we use the same well-trained model for initialization and the same data partition on each client for CalFAT and FedGAIRAT.

In Figure 3, we report the per-class natural and robust accuracies of CalFAT and FedGAIRAT on CIFAR10. As shown in these figures, the average performance of most classes of CalFAT is much higher than FedGAIRAT. We also report the per-class performance of each client on CIFAR10 in Appendix D.2. In FedGAIRAT, due to the highly skewed label distribution, the prediction of each client is highly biased to the majority classes, which leads to high performance on the majority classes and low performance (even 0% accuracy) on the minority classes. By contrast, in CalFAT, each client has higher performance on most classes. This verifies that the calibrated cross-entropy loss can indeed improve the performance on the minority classes, and further improve the overall performance of the model. Moreover, we report the per-class average performance on SVHN in Appendix D.3. Our CalFAT also outperforms the best baseline across most of the classes on SVHN.

4.3 Results on Different FL Frameworks and Network Architectures

Evaluation on different FL frameworks. Besides FedAvg [23], we also conduct experiments on other FL frameworks, i.e., FedProx [16] and Scaffold [11]. The results for all methods on FedProx and Scaffold are given in Table 3. It shows that our CalFAT exhibits better natural and robust accuracies than all baseline methods on all FL frameworks, which indicates the high comparability of our CalFAT with different aggregation algorithms.

Evaluation on different network architectures. We also compare CalFAT with baselines on different network architectures, i.e., CNN [23], VGG-8 [27], and ResNet-18 [8]. For CNN, we use the same architecture as [23]. VGG-8 and ResNet-18 are two widely used architectures in deep learning. The results on CIFAR10 dataset are shown in Appendix D.4. CalFAT outperforms all baselines, which further validates the superiority of CalFAT with different network architectures.

4.4 Feature Visualization

To better understand the efficacy of CalFAT, we visualize the learned features extracted from the second last layer of FedTRADES (the best baseline) and CalFAT trained on SVHN dataset in Appendix D.5. The features are projected into a 2-dimensional space via t-SNE [31]. It shows that

samples from different classes are mixed together in FedTRADES, indicating its low performance. For instance, Class 6 (pink) and Class 8 (khaki) are hard to separate in FedTRADES while these 2 classes can be well separated by CalFAT. This illustration verifies that the server cannot learn a global model with good inter-class separability if the local models are heterogeneous. By contrast, CalFAT can well separate different classes thus can achieve better overall performance.

4.5 Results under IID settings

Besides the non-IID setting, we also conduct an experiment under the IID setting. The results are shown in Appendix D.6 where it shows that our CalFAT achieves the best robustness (under PGD-20 attack). Compared to the non-IID setting, all FAT methods demonstrate much better performance under the IID setting. This indicates that existing FAT methods can easily handle IID data yet face substantial challenges when the data is non-IID.

4.6 Ablation Studies

Impact of the number of clients. To show the generality of CalFAT, we train CalFAT with different numbers of clients m. Table 7 in Appendix D.7 reports the results for $m = \{20, 50, 100\}$. As expected, CalFAT achieves the best performance across all m. As m increases, the performance of all methods decreases. We conjecture that this is because more clients in FAT makes the training harder to converge. However, our CalFAT can still achieve 41.23% natural accuracy when there are 100 clients, outperforming other baselines by a large margin.

Impact of skewed label distribution. We observe that the performance of FAT defense is closely related to label skewness. We thus investigate the impact of skewed label distribution by varying the Dirichlet parameter $\beta = \{0.05, 0.2, 0.3\}$ and report the results on CIFAR10 in Table 8 in Appendix D.8. Not surprisingly, our CalFAT outperforms all baselines under all β's. This further verifies the consistent effectiveness of CalFAT under different levels of label skewness.

Note that as β decreases (i.e., the labels on each client are more imbalanced), the performance of all methods drop rapidly. For example, the natural accuracy of FedMART drops from 38.38% to 29.84% as β decreases from 0.2 to 0.05. This indicates that all methods are hard to train a good model in extremely skewed label distribution scenarios. However, our CalFAT still achieves 61.00% natural accuracy and 32.40% robust accuracy (against FGSM attack) when $\beta = 0.05$, which are much higher than all the baselines.

Contribution of the calibrated loss functions. As shown in Eq. (13) and Eq. (15), for each client i, we have two new loss functions: a CCE loss $\ell_{cce}(\cdot,\cdot,\cdot)$ for optimization and a CKL loss $\ell_{ckl}(\cdot,\cdot,\cdot)$ for generating the adversarial examples. This naturally raises a question: how do these two loss functions contribute to CalFAT? To answer this question, we conduct leave-one-out tests by removing the CCE loss (w/o $\ell_{cce}(\cdot,\cdot,\cdot)$) or removing the CKL loss (w/o $\ell_{ckl}(\cdot,\cdot,\cdot)$) from the overall optimization objective. As illustrated in Appendix D.9, w/o $\ell_{cce}(\cdot,\cdot,\cdot)$ leads to poor performance, which implies that CCE loss plays an important role in CalFAT. Besides, if we only use the CCE loss (i.e., w/o $\ell_{ckl}(\cdot,\cdot,\cdot)$), we can obtain a much better performance, but it still underperforms CalFAT. All these results indicate that the CCE loss is the most important part of CalFAT, whilst the CKL loss can further increase the performance of CalFAT. The combination of both loss functions leads to the best performance.

Impact of the ratio of adversarial data. Here, we conduct experiments with different ratios of adversarial data used in CalFAT and report the robust accuracy (against PGD-20 attack) in Appendix D.10. Ratios r=0 and r=1 stand for training the model on only natural data and only adversarial data, respectively. Overall, r=1 produces the best robustness, meaning that training on only adversarial data can better enhance the adversarial robustness of our CalFAT.

5 Conclusion

In this paper, we studied the challenging problem of Federated Adversarial Training (FAT) with label skewness and proposed a novel Calibrated Federated Adversarial Training (CalFAT) to simultaneously achieve stable training, better convergence, and natural accuracy and robustness in FL. CalFAT

calibrates the model prediction and trains homogeneous local models across different clients by automatically assigning higher scores to the minority classes. Extensive experiments on multiple datasets under various settings validate the effectiveness of CalFAT. Our work can serve as a simple but strong baseline for accurate and robust FAT. For future work, we will continue to improve FAT under other non-IID settings such as feature skewness and quantity skewness [10, 46].

Acknowledgement

This work is funded by Sony AI. This work is also supported by the National Key R&D Program of China (Grant No. 2021ZD0112804) and the National Natural Science Foundation of China (Grant No. 62276067).

References

[1] Peter J Bickel and Kjell A Doksum. *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. CRC Press, 2015.

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[3] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 11192–11203, 2019.

[4] Chen Chen, Jingfeng Zhang, Xilie Xu, Lingjuan Lyu, Chaochao Chen, Tianlei Hu, and Gang Chen. Decision boundary-aware data augmentation for adversarial training. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216, 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[8] 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*, pages 770–778, 2016.

[9] Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. Federated robustness propagation: Sharing adversarial robustness in federated learning. *arXiv preprint arXiv:2106.10196*, 2021.

[10] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020.

[11] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143, 2020.

[12] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. In *Technical report*, 2009.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[16] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[18] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34, 2021.

[19] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. Real-world image datasets for federated learning. *arXiv preprint arXiv:1910.11089*, 2019.

[20] Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S Yu. Privacy and robustness in federated learning: Attacks and defenses. *arXiv preprint arXiv:2012.06337*, 2020.

[21] Lingjuan Lyu, Han Yu, Jun Zhao, and Qiang Yang. Threats to federated learning. In *Federated Learning*, pages 3–16. Springer, 2020.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

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

[24] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[25] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[26] Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[29] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*.

[30] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2018.

[31] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[32] Abraham Wald. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

[33] Wentao Wang, Han Xu, Xiaorui Liu, Yaxin Li, Bhavani Thuraisingham, and Jiliang Tang. Imbalanced adversarial training with reweighting. *arXiv preprint arXiv:2107.13639*, 2021.

[34] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

[35] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[36] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. Adversarial robustness under long-tailed distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8659–8668, 2021.

[37] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*, pages 11492–11501, 2021.

[38] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[39] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261, 2019.

[40] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.

[41] Jie Zhang, Chen Chen, Jiahua Dong, Ruoxi Jia, and Lingjuan Lyu. Qekd: Query-efficient and data-free knowledge distillation from black-box models. *arXiv preprint arXiv:2205.11158*, 2022.

[42] Jie Zhang, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, Lei Zhang, and Chao Wu. Towards efficient data free black-box adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15115–15125, 2022.

[43] 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 *International Conference on Machine Learning*, pages 26311–26329, 2022.

[44] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

[45] Yao Zhou, Jun Wu, and Jingrui He. Adversarially robust federated learning for neural networks, 2021.

[46] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[47] Giulio Zizzo, Ambrish Rawat, Mathieu Sinn, and Beat Buesser. Fat: Federated adversarial training. *arXiv preprint arXiv:2012.01791*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]

(c) Did you discuss any potential negative societal impacts of your work? [No]
(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]