GEAR: A Margin-based Federated Adversarial Training Approach

Chen Chen[1*], **Jie Zhang**[1*], **Lingjuan Lyu**[2†]

[1] Zhejiang University, China
[2]Sony AI, Japan

Abstract

Previous studies have shown that federated learning (FL) is vulnerable to well-crafted adversarial examples. Some recent efforts tried to combine adversarial training with FL, i.e., federated adversarial training (FAT), in order to achieve adversarial robustness in FL. However, most of the existing FAT works suffer from either low natural accuracy or low robust accuracy. Moreover, none of these works provide a more in-depth understanding of the challenges behind adversarial robustness in FL. To address these issues, we propose a novel marGin-based fEderated Adversarial tRaining Approach called GEAR. It encourages the minority classes to have larger margins by introducing a margin-based cross-entropy loss, and regularizes the decision boundary to be smooth by introducing a regularization loss, thus providing a better decision boundary for the global model. To the best of our knowledge, this work is the first to investigate the impact of decision boundary on FAT and delivers the best natural accuracy and robust accuracy in FL by far. Extensive experiments on multiple datasets across various settings all validate the effectiveness of our proposed method. For example, on SVHN dataset, GEAR can improve the natural accuracy and robust accuracy (against FGSM attack) of the best baseline method (FedTRADES) by 20.17% and 10.73%, respectively.

1 Introduction

Federated learning (FL) has emerged as a privacy-aware learning paradigm that allows multiple participants (clients) to collaboratively train a better global model (McMahan et al. 2017; Lyu and Chen 2021; Tan et al. 2022). In FL, each client follows the standard machine learning training procedure (i.e., standard training) to train a local model on its own data and periodically shares its model parameter with a central server for aggregation. Over the past few years, FL has gained significant attention in a wide range of applications such as next word prediction (McMahan et al. 2017), visual object detection for safety (Liu et al. 2020), recommendation (Wu et al. 2021, 2020; Cui et al. 2021), etc.

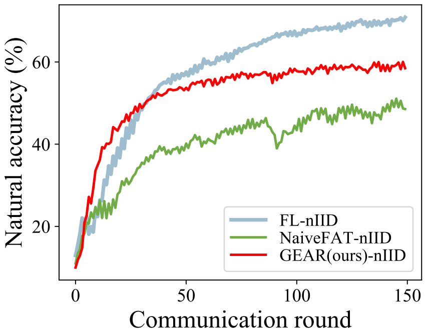


Figure 1: Natural accuracy of standard FL with non-IID data (FL-nIID), NaiveFAT with non-IID data (NaiveFAT-nIID), and our proposed GEAR with non-IID data (GEAR-nIID) on CIFAR10 dataset.

However, similar to the centralized learning, recent studies (Zizzo et al. 2020; Hong et al. 2021) have shown that FL is also vulnerable to well-crafted adversarial examples. During inference time, the attackers can add a very small perturbation to the test data, making the test data almost indistinguishable from natural data and yet classified incorrectly by the global model. In cross-silo FL, such vulnerability especially matters and may cause heavy losses. For example, in a medical-based cross-silo FL, a vulnerable global model may lead to incorrect diagnoses, even leading to the loss of human lives. Similarly, in a financial-based cross-silo FL, lacking of adversarial robustness may lead to a huge money loss and cause financial chaos. Therefore, it is urgent to train a global model that is robust against adversarial attacks.

Adversarial training (AT) is one of the most effective strategies for enhancing the model robustness against adversarial attacks (Madry et al. 2017; Dong et al. 2020, 2021). Since FL clients also desire model robustness against malicious attackers at inference time, recent studies (Zizzo et al. 2020; Hong et al. 2021) proposed to apply AT to FL, i.e, federated adversarial training (FAT), in order to achieve adversarial robustness in FL. (Zizzo et al. 2020) noticed that conducting AT on all clients leads to divergence of the model. To solve this problem, they conducted AT on only a proportion of clients for better convergence, and we term their method as NaiveFAT. Another recent work (Hong et al. 2021) considered hardware heterogeneity in FL, i.e., only limited users can afford AT. Hence, they conduct adversarial training (AT)

[*] Chen Chen and Jie Zhang contributed equally. This work was done at Sony AI.

[†] Corresponding author.

This work was accepted at AAAI 2022 workshop on Trustable, Verifiable and Auditable Federated Learning (FL-AAAI-22).

on only a proportion of clients that have powerful computation resources while conducting standard training on the rest of the clients. We remark that this setting is different from our considered cross-silo FL setting (Yang et al. 2019), in which robustness matters a lot. Different from cross-device FL, in cross-silo FL, a handful of clients have relatively high participation levels and technical capabilities, i.e., they possess significant computational power and sophisticated technical capabilities (Lyu et al. 2022). Therefore, clients in cross-silo FL are capable of conducting AT in order to collaboratively achieve adversarial robustness in FL. However, most of the existing FAT works (Zizzo et al. 2020) suffer from either low natural accuracy or low robust accuracy. Moreover, none of the previous works provide a more in-depth understanding of the challenges behind adversarial robustness in FL, especially in not independent and identically distributed (non-IID) settings. This motivates us to explore a more effective approach to maintain both natural accuracy and robust accuracy in FL. Meanwhile, we aim to provide a more in-depth understanding of adversarial robustness in FL from the perspective of decision boundary.

In terms of data settings in FL, there may exist label skewness in each client, i.e., their data are non-IID. Adapting adversarial training to FL becomes more challenging in the non-IID settings. Hence, in the rest of the paper, we mainly focus on FAT under the more challenging non-IID data setting. In Figure 1, we report the natural accuracy (accuracy on natural data) of standard FL with non-IID data (FL-nIID) (McMahan et al. 2017), NaiveFAT with non-IID data (NaiveFAT-nIID) (Zizzo et al. 2020), and our proposed GEAR with non-IID data (GEAR-nIID) on CIFAR10 dataset (Krizhevsky, Hinton et al. 2009), respectively. We use the default settings (shown in Section 4.1) to train these models. From Figure 1, we observe that: (1) The natural accuracy of NaiveFAT-nIID (green line) is much lower than FL-nIID (blue line), which implies that directly adapting adversarial training to FL is not suitable and can harm the performance of the global model; (2) our proposed GEAR with non-IID data (GEAR-nIID) (red line) significantly outperforms NaiveFAT-nIID (green line).

We argue that the degradation of the performance in NaiveFAT (with non-IID data) is mainly caused by the bad decision boundary. To this end, we propose a novel FAT method called marGin-based fEderated Adversarial tRaining (GEAR), which aims to enhance FAT by learning a better decision boundary. In particular, we design a new loss function for GEAR which encourages the minority classes to have larger margins and regularizes the decision boundary to be smooth. In summary, our main contributions include:

- We present a novel FAT method named marGin-based fEderated Adversarial tRaining (GEAR), which can encourage a larger margin between the training data of minority classes and the decision boundary by introducing a margin-based cross-entropy loss, and regularizes the decision boundary to be smooth by introducing a regularization loss, thus delivering a better decision boundary for the global model.
- To the best of our knowledge, this work is the first to investigate the impact of decision boundary on FAT and delivers the best natural accuracy and robust accuracy. We show that the performance degradation of NaiveFAT is mainly caused by the bad decision boundary, thus providing a more in-depth understanding of the challenges behind adversarial robustness in FL.
- Extensive experiments validate the effectiveness of our proposed method. For example, our GEAR can improve the natural accuracy and robust accuracy (against FGSM attack) of the best baseline method (FedTRADES) on SVHN dataset by 20.17% and 10.73%, respectively.

2 Notation and preliminaries

2.1 Notation

Let $\mathcal{S}$ be the set of clients and $m = |\mathcal{S}|$ be the number of clients. Subscript i denotes the elements of client i, e.g., $\mathcal{D}_i$ denotes the local training dataset of client i, and θ_i denotes the local model parameter of client i. We use $\hat{\theta}$ to denote the server's global model parameter. Suppose there are C classes for each input. Subscript j denotes the j-th record, e.g., (x_j, y_j) denotes the j-th record and the corresponding label (class) with $y_j \in \{1, \cdots, C\}$. $z_j = [z_j^1, \cdots, z_j^C]$ denotes the output of the model for the j-th training record and z_j^l be the output for the l-th class.

2.2 Centralized adversarial training

Let $\mathcal{D} = \{x_j, y_j\}_{j=1}^n$ be the training dataset, n be the size of data. The objective function of centralized adversarial training (AT) (Madry et al. 2017) can be expressed as:

$$\min_\theta \frac{1}{n} \sum_{j=1}^{n} \ell_{ce}(z_j, y_j), \tag{1}$$

where

$$z_j = f_\theta(x_{adv,j}), \tag{2}$$

and

$$x_{adv,j} = \underset{x' \in \mathcal{B}_\epsilon(x_j)}{\operatorname{argmax}}\ \ell_{ce}(f_\theta(x'), y_j). \tag{3}$$

$\mathcal{B}_\epsilon(x_j) = \{x' \mid \|x' - x_j\|_\infty < \epsilon\}$ is the closed ball of radius $\epsilon > 0$ centered at x_j, $\|\cdot\|_\infty$ is the L_∞ norm, x_{adv} is the most adversarial data within the ϵ-ball centered at x, $f_\theta(\cdot)$ is the model with parameter θ. Centralized AT utilizes the cross-entropy loss function to optimize the parameter:

$$\ell_{ce}(z_j, y_j) = -\log \frac{\exp(z_j^{y_j})}{\sum_{l=1}^{C} \exp(z_j^l)}, \tag{4}$$

where $\exp(a) = e^a$ is the exponential function, z_j^l is the output of the j-th training record for the l-th class, and $z_j^{y_j}$ is the output of the j-th training record for the ground truth class.

A standard centralized AT uses projected gradient decent (PGD) to approximate Eq. (3). In particular, PGD iteratively generates adversarial data as follows:

$$x_j^{(k+1)} = \Pi_{\mathcal{B}_\epsilon(x_j^{(0)})} \left(x_j^{(k)} + \alpha \operatorname{sign}(\nabla_{x_j^{(k)}} \ell(f_\theta(x_j^{(k)}), y_j)) \right), \tag{5}$$

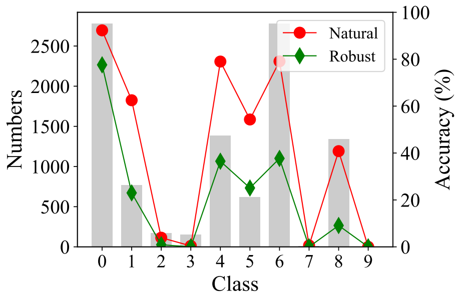


Figure 2: CIFAR10 class-wise natural accuracy (red line) and robust accuracy (against PGD-20 attack) (green line) of the local model of a randomly selected client in NaiveFAT. The grey histogram demonstrates the training data size of each class. The client receives the global model (which has already converged) from the server, and trains on its local data for one epoch. Then, we evaluate the model with the (balanced) test data.

where $k = 0, \cdots, K-1$ is the step number, K is the maximum number of steps, $\alpha > 0$ is the step size, $x_j^{(0)}$ is the natural data, $x_j^{(k)}$ is the adversarial data 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 $\operatorname{sign}(a)$ is the sign function that returns 1 if $a > 0$, otherwise returns 0.

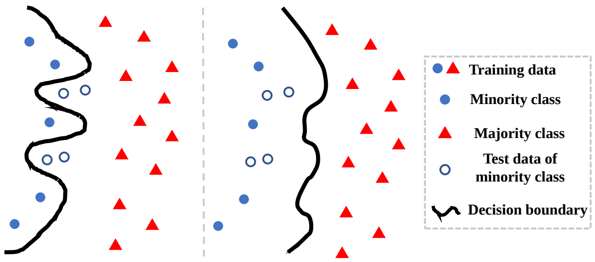
2.3 Federated adversarial training

Federated adversarial training (FAT) is first introduced by (Zizzo et al. 2020), which aims to deal with adversarial examples in FL under the statistical heterogeneity setting. We call their proposed method as NaiveFAT. In particular, suppose there are m clients in a FAT system, and each client i has its local dataset $\mathcal{D}_i = \{x_j, y_j\}_{j=1}^{n_i}$ with $n_i = |\mathcal{D}_i|$ being the size of local data. Each client i optimizes its local model by minimizing the following objective function:

$$\min_{\theta_i} \frac{1}{n_i}\left(\sum_{j=1}^{n_i'} \ell_{ce}(f_{\theta_i}(x_{adv,j}), y_j) + \sum_{j=n_i'}^{n_i} \ell_{ce}(f_{\theta_i}(x_j), y_j)\right), \tag{6}$$

where $x_{adv,j}$ is the most adversarial data within the ϵ-ball centered at x_j (defined in Eq. (3)), n_i' is a hyperparameter that controls the ratio of clients which conduct AT, and θ_i is the local model parameter. Similar to centralized AT, the most adversarial data $x_{adv,j}$ is generated by PGD (as shown in Eq. (5)).

After training the local model, client i uploads its local model parameter θ_i to the central server for aggregation. After receiving the local model parameters from the clients, the server derives the global model parameter according to the aggregation framework such as FedAvg (McMahan et al. 2017). Afterwards, the server distributes the global model parameter to all clients for training in the next round. Compared with standard FL, a proportion of clients in NaiveFAT conducts AT instead of standard training.



(a) NaiveFAT (b) GEAR (ours)

Figure 3: The illustration of decision boundaries of NaiveFAT and our GEAR. Blue solid circles are training data of the minority class while red solid triangles are training data of the majority class. Blue hollow circles are test data of the minority class. The decision boundary of NaiveFAT is distorted and very close to the minority class, and such a bad decision boundary cannot well-separate the test data of minority class. The decision boundary of GEAR is smoother (compared to NaiveFAT) and can better separate the test data of the minority class.

Although NaiveFAT can gain robustness in defending against adversarial examples, the performance of NaiveFAT is relatively low. As shown in Figure 1, NaiveFAT with non-IID data (NaiveFAT-nIID) is significantly lower than the standard FL with non-IID data (FL-nIID).

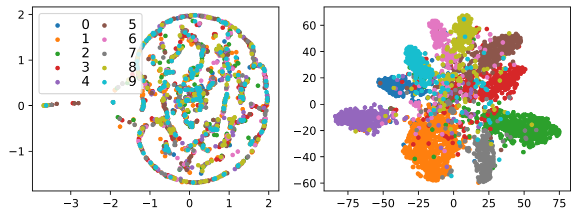
3 marGin-based fEderated Adversarial tRaining (GEAR)

3.1 NaiveFAT fails to learn a good decision boundary

As discussed in the previous section, NaiveFAT (Zizzo et al. 2020) cannot achieve high performance. We argue this is due to the bad decision boundary. We summarize the challenges in learning the decision boundary in FAT as follows.

The first challenge is the statistical heterogeneity, where label skewness prevalently exists in FL. The decision boundary learned on the skewed data is close to the data with minority classes and thus has a much lower performance on minority classes (Van Horn and Perona 2017; Buda, Maki, and Mazurowski 2018). In Figure 2, we show the performance of the learned local model of a randomly selected client. We use the default settings (shown in Section 4.1) to train the model. There is a label skewness on this client and the grey histogram demonstrates the training data size of each class. The performance of majority classes obviously outperform the performance of minority classes. For example, Class 0 which belongs to the majority classes has the highest natural and robust accuracy. On the other hand, Class 7 from the minority classes achieves a much lower performance.

The second challenge is due to that each client utilizes AT for local updating, i.e., it uses the most adversarial data to update the local model, which leads to a distorted decision



(a) NaiveFAT (b) GEAR

Figure 4: T-SNE feature visualization of NaiveFAT and GEAR on SVHN dataset. Different colored dots denote test data from different classes. The data from different classes are mixed together and unable to be well-separated in NaiveFAT while GEAR can learn more discriminative features.

boundary. The most adversarial data usually significantly cross over the decision boundary and are located at the area of other classes (Zhang et al. 2020; Ding et al. 2020). When the data on each client are non-IID, the minority classes will definitely be overwhelmed by the majority classes, resulting in a distorted decision boundary that is very close to the minority classes. We illustrate this phenomenon in Figure 3(a). Since red triangles are data of the majority class, the most adversarial data of red triangles will cross over the decision boundary and locate at the area of blue circles. These adversarial data will push the decision boundary away from red triangles and locate at the area of blue circles, resulting in a bad decision boundary and a low performance on minority classes (blue circles).

Due to the above two challenges, the global model is hard to provide satisfactory performance. As discussed above, each client may upload a distorted model to the server for aggregation. Such distorted local models will result in a slow and unstable convergence, i.e., the global model cannot effectively learn from the distorted local models. To illustrate this fact, we show the learned features extracted from the second last layer of the NaiveFAT model trained on SVHN dataset in Figure 4(a). We use the default settings (shown in Section 4.1) to train NaiveFAT. The extracted features are visualized in 2-dimensional space by t-SNE (Van der Maaten and Hinton 2008). As shown in this figure, data from different classes are mixed together and hard to be separated. This illustration further verifies that the server cannot learn a good global model from the distorted local models. By contrast, our proposed GEAR (shown in Figure 3(b)) can well separate the classes.

3.2 Detail of GEAR

To address the above challenges, we propose a novel FAT method called marGin-based fEderated Adversarial tRaining (GEAR), which encourages the minority classes to have larger margins and regularizes the decision boundary to be smooth. The overall training process of GEAR is shown in Algorithm 1.

Algorithm 1: Training process of GEAR

```
Input: Number of client $m$, clients' local dataset $\{\mathcal{D}_1, \cdots, \mathcal{D}_m\}$, data size of each client $\{n_1, \cdots, n_m\}$, total size of data $n$, learning rate $\eta$, local epochs $E$, communication rounds $T$, and loss function $\ell_{GEAR}(\cdot, \cdot, \cdot)$
Output: Global model parameter $\hat{\theta}$
procedure ServerAggregation
    for communication round $t$ = 1, 2, ..., $T$ do
        for each client $i = 1, \ldots, m$ do in parallel
            $\theta_i \leftarrow$ClientUpdate$(i, \hat{\theta})$
        end for
        $\hat{\theta} \leftarrow \sum_{i=1}^{m} \frac{n_i}{n} \theta_i$
    end for
    return $\hat{\theta}$
end procedure
procedure ClientUpdate$(i, \hat{\theta})$
    $\theta_i = \hat{\theta}$
    for local epoch=1, $\cdots$, $E$ do
        for $j = 1, \cdots, n_i$ do
            Sample $(x_j, y_j)$ from $\mathcal{D}_i$
            Generate adversarial data $x_{adv,j}$
            $z_j \leftarrow f_{\theta_i}(x_{adv,j})$
        end for
        $\theta_i \leftarrow \theta_i - \eta \frac{1}{n_i} \sum_{j=1}^{n_i} \nabla_{\theta_i} \ell_{GEAR}(z_j, y_j, \theta_i)$
    end for
    return $\theta_i$
end procedure
```

In particular, for each client i, we propose to minimize the following objective function:

$$\min_{\theta_i} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell_{GEAR}(z_j, y_j, \theta_i), \tag{7}$$

where $z_j = f_{\theta_i}(x_{adv,j})$. We follow the setting of (Zhang et al. 2019) and generate the most adversarial data $x_{adv,j}$ by maximizing the following loss function:

$$x_{adv,j} = \underset{x' \in \mathcal{B}_\epsilon(x_j)}{\operatorname{argmax}} \, \ell_{ce}(f_{\theta_i}(x'), f_{\theta_i}(x_j)), \tag{8}$$

where $\ell_{ce}(\cdot, \cdot)$ is the cross-entropy loss.

The loss function of GEAR can be formulated as follows:

$$\ell_{GEAR}(z_j, y_j, \theta_i) = \ell_{mar}(z_j, y_j) + \lambda \ell_{reg}(\theta_i), \tag{9}$$

where $\ell_{mar}(\cdot, \cdot)$ is the margin-based cross-entropy loss, $\ell_{reg}(\cdot)$ is the regularization loss, and λ is the scaling factors. Below, we introduce the margin-based cross-entropy loss and regularization loss respectively, in order to remedy the bad decision boundary learned by NaiveFAT, as shown in Figure 3(a).

Margin-based cross-entropy loss Since the learned decision boundary of NaiveFAT is too close to the training data in minority classes (refer to Figure 3(a)), we argue to encourage a larger margin between the minority classes and the decision boundary in order to improve performance. Inspired by (Cao et al. 2019), we add a margin to the output of the ground truth class as follows:

$$\ell_{mar}(z_j, y_j) = -\log \frac{\exp(z_j^{y_j} - m_i^{y_j})}{\exp(z_j^{y_j} - m_i^{y_j}) + \sum_{l \neq y_j} \exp(z_j^l)}, \tag{10}$$

where

$$m_i^l = \frac{\delta \min\{\sqrt[4]{n_i^1}, \cdots, \sqrt[4]{n_i^C}\}}{\sqrt[4]{n_i^l}} \quad \text{for } l = 1, \cdots, C, \tag{11}$$

m_i^l is the margin for l-th class on client i, n_i^l is the size of data with l-th class on client i, $y_j \in \{1, \cdots, C\}$ is the ground truth class, and δ is a hyperparameter that controls the maximum margin. If class l is the minority class on client i, i.e., n_i^l is small, then its margin m_i^l will be high.

Regularization loss We further propose to regularize the decision boundary by adding a L_2 norm:

$$\ell_{reg}(\theta_i) = \|\theta_i\|_2 . \tag{12}$$

Combining all the above two loss functions (Refer to Eq. 9), each client optimizes its local model parameters as follows:

$$\theta_i = \theta_i - \eta \frac{1}{n_i} \sum_{j=1}^{n_i} \nabla_{\theta_i} \ell_{GEAR}(z_j, y_j, \theta_i). \tag{13}$$

After optimizing the local model parameter, each client uploads its local model parameter to the server. The server computes the global model parameter as follows:

$$\hat{\theta} = \sum_{i=1}^{m} \frac{n_i}{n} \theta_i, \tag{14}$$

where $n = \sum_{i=1}^{m} n_i$ is the total size of training data in all clients, and $\hat{\theta}$ is the global model parameter.

We remark that by utilizing the above loss functions, we can (1) increase the margin between the decision boundary and the minority classes; and (2) smooth the decision boundary. We illustrate the decision boundary learned by GEAR in Figure 3(b). The decision boundary is much smoother (compared to NaiveFAT in Figure 3(a)) and can better separate the test data of the minority class.

To further show the efficacy of our GEAR, we illustrate the learned features extracted from the second last layer of our GEAR model on SVHN dataset by using t-SNE in Figure 4(b). GEAR can learn more discriminative features (compared to NaiveFAT in Figure 4(a)), thus it can better separate the data from different classes.

4 Experiments

4.1 Experimental setup

Datasets Our experiments are conducted on 3 real-world datasets: CIFAR10 (Krizhevsky, Hinton et al. 2009), CIFAR100 (Krizhevsky, Hinton et al. 2009), and SVHN (Netzer et al. 2011). CIFAR10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images in CIFAR10 dataset. CIFAR100 dataset is similar to CIFAR10 dataset, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. SVHN is a real-world image dataset consisting of 73,257 training data and 26,032 testing data in 10 classes. SVHN dataset is obtained from house numbers in Google Street View images.

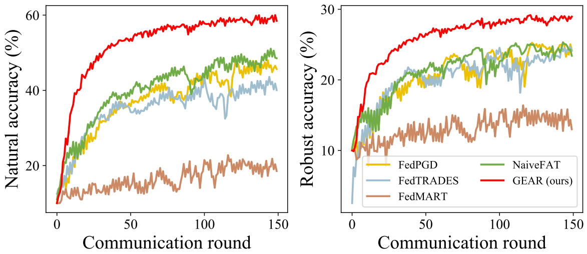


Figure 5: Natural accuracy and robust accuracy (against PGD-20 attack) of different methods. Our GEAR (red line) achieves the best natural accuracy and robust accuracy.

Data partition To simulate real-world statistical heterogeneity, we use Dirichlet distribution to generate non-IID data partition among clients (Yurochkin et al. 2019; Li et al. 2021). In particular, we sample $p_i^l \sim Dir(\beta)$ and allocate a p_i^l proportion of the data of class l to client i, where $Dir(\beta)$ is the Dirichlet distribution with a concentration parameter β. By sampling from the Dirichlet distribution, each client can have relatively few data samples in some classes. To simulate a highly skewed data partition, we set $\beta = 0.1$ as default.

Baselines We compare our proposed GEAR with the first FAT method (NaiveFAT) (Zizzo et al. 2020). We also investigate the combination of the state-of-the-art centralized AT methods with FL, i.e., we extend standard PGD (Madry et al. 2017), TRADES (Zhang et al. 2019), and MART (Wang et al. 2020)) to FL, and name them as FedPGD, FedTRADES, and FedMART.

Metric For evaluations, we report natural test accuracy (Natural) for natural test data and robust test accuracy for adversarial test data. The adversarial test data are generated by FGSM (fast gradient sign method) (Wong, Rice, and Kolter 2020), BIM (basic iterative method with 20 steps) (Kurakin, Goodfellow, and Bengio 2016), PGD-20 (projected gradient descent with 20 steps) (Madry et al. 2017), CW (CW with 20 steps) (Carlini and Wagner 2017), and AA (auto attack) (Croce and Hein 2020) with the same perturbation bound $\epsilon = 8/255$. The step size α for BIM, PGD-20 attack, and CW attack is $2/255$.

Setting In our experiments, we consider $||\tilde{x} - x||_\infty < \epsilon$ with the same ϵ in both training and evaluations. To generate the most adversarial data to update the model, we follow the same setting as (Rice, Wong, and Kolter 2020), i.e., we set the perturbation bound $\epsilon = 8/255$; PGD step number $K = 10$; and PGD step size $\alpha = 2/255$. We train the model by using SGD with momentum$=$ 0.9 and learning

Table 1: Natural and robust accuracies across different datasets. Best results are in bold.

Dataset	SVHN						CIFAR10						CIFAR100					
Metric	Natural	FGSM	BIM	CW	PGD-20	AA	Natural	FGSM	BIM	CW	PGD-20	AA	Natural	FGSM	BIM	CW	PGD-20	AA
NaiveFAT	19.64	19.63	19.64	19.63	19.65	14.71	53.23	29.02	26.37	22.79	26.22	21.80	34.39	15.79	14.63	11.34	14.50	9.31
FedPGD	19.47	19.46	19.45	19.46	19.47	13.67	47.21	28.80	26.68	24.53	26.50	**22.80**	34.06	16.10	14.76	11.53	14.70	10.80
FedTRADES	56.84	36.96	35.11	31.16	34.97	30.56	46.14	27.68	26.36	22.81	26.29	21.60	29.35	14.99	14.20	10.52	14.23	9.56
FedMART	19.81	19.80	19.79	19.80	19.81	14.56	25.68	18.49	18.15	15.39	18.15	14.30	19.85	13.01	12.77	9.99	12.79	8.67
GEAR (ours)	**77.01**	**47.69**	**42.05**	**31.72**	**41.74**	**35.69**	**59.91**	**33.74**	**29.90**	**24.68**	**29.63**	22.68	**41.09**	**17.77**	**15.62**	**12.16**	**15.36**	**11.39**

Table 2: Natural and robust accuracies across different β on CIFAR10 dataset. Best results are in bold.

non-IID	β=0.05						β=0.2						β=0.3					
Metric	Natural	FGSM	BIM	CW	PGD-20	AA	Natural	FGSM	BIM	CW	PGD-20	AA	Natural	FGSM	BIM	CW	PGD-20	AA
NaiveFAT	49.10	27.49	25.32	22.17	25.24	22.51	54.85	31.27	28.77	26.08	28.46	25.21	58.93	31.68	28.17	24.96	28.00	24.34
FedPGD	47.13	26.63	24.96	20.75	25.03	21.28	52.22	30.31	28.64	25.49	28.59	24.92	56.12	30.86	28.46	25.07	28.29	23.64
FedTRADES	40.24	26.02	25.06	22.48	24.99	20.16	48.52	29.94	28.73	25.57	28.65	24.15	54.26	30.83	29.39	24.74	29.26	23.87
FedMART	29.84	21.90	21.39	18.31	21.41	17.89	38.38	27.59	27.05	23.31	26.99	21.89	40.96	28.32	27.88	23.12	27.80	22.16
GEAR (ours)	**61.00**	**32.40**	**29.75**	**23.55**	**29.50**	**25.66**	**71.55**	**33.80**	**30.70**	**26.25**	**29.35**	**26.32**	**69.95**	**34.25**	**30.80**	**25.76**	**30.96**	**26.84**

rate $\eta = 0.01$. The maximum margin $\delta = 0.5$, the number of communication rounds $T = 150$, and the number of local epochs $E = 1$. We set scaling factor $\lambda = 2e - 3$ for CIFAR10 and CIFAR100 datasets, and $\lambda = 2e - 4$ for SVHN dataset. All methods use FedAvg for aggregation and are trained on the same CNN model, which keeps the same as (McMahan et al. 2017). Recall that compared with cross-device setting, FAT matters more in the cross-silo setting, in which the number of clients is relatively small, and each client has powerful computation resources to handle the large computation cost of AT (Lyu et al. 2020). Thus, we set the number of clients $m = 5$ by default, and in each epoch, all clients are involved in the training.

4.2 Experimental results

Evaluation on different datasets Table 1 shows the results of different methods on different datasets. From the table, we can observe that:

(1) Our GEAR significantly outperforms all baselines on all datasets. For example, GEAR outperforms the best baseline method (FedTRADES) by 10.73% on SVHN dataset under FGSM attack. These results validate the efficacy of our GEAR.

(2) Our GEAR shows a significant improvement in natural accuracy compared to other baselines. For example, GEAR can improve the natural accuracy by 20.17% of the best baseline method (FedTRADES) on SVHN dataset. We hypothesise that the reason lies in the increased margin of minority classes, which can significantly improve the natural accuracy on minority classes of each client.

(3) All methods demonstrate the worst performance on CIFAR100 dataset. We conjecture that this is due to the more classes in CIFAR100, which makes the training hard. Nevertheless, our GEAR still achieves the best performance on CIFAR100 dataset.

(4) NaiveFAT delivers a similar performance compared to FedPGD. This implies that naively conducting AT on a proportion of clients cannot really improve the natural accuracy nor improve the robustness of the model.

(5) FedMART has almost the worst performance across all methods, which indicates that MART is not suitable to be directly applied to cross-silo FL.

Learning curves of different methods To better compare our GEAR with the baseline methods, we plot the learning curves (i.e, performance across different communication rounds) of all methods in Figure 5. As shown in the figure, GEAR achieves the best natural accuracy and robust accuracy from the beginning to the end of the training, which indicates that the design of our GEAR is profitable across the whole training process.

Moreover, our GEAR is stable during the whole training process while the accuracy curves of other baselines oscillate strongly. Such oscillations make the training hard and lead to bad performance. We hypothesise that the smooth decision boundary of our GEAR makes the training stable.

Impact of label skewness We find that the performance of these FAT methods are closely related to the level of label skewness. We investigate the impact of label skewness by varying the Dirichlet parameter $\beta = \{0.05, 0.2, 0.3\}$ and report the results on CIFAR10 dataset in Table 2. Not surprisingly, our GEAR outperforms all baselines on all β. This further verifies the consistent effectiveness of GEAR on non-IID data.

Note that as β decreases (i.e., the labels of the data on each client are more imbalanced), the performances 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 on extreme label skewness scenarios. However, our GEAR still can achieve 54.03% natural accuracy and 31.99% robust accuracy (against FGSM attack), which are higher than all the baselines.

Impact of regularization weight λ We further conduct experiments across different λ to show the effectiveness of our proposed regularization loss. Table 3 demonstrates the results of GEAR on CIFAR10 dataset across $\lambda = \{2e-2, 2e-3, 2e-4\}$ and $\beta = \{0.05, 0.1, 0.2, 0.3\}$.

The results show that when $\lambda = 2e - 3$, GEAR achieves the best performance across all β. When λ is too large (e.g., $\lambda = 2e - 2$), the model is hard to learn, which results in a bad performance. When λ is too small (e.g., $\lambda = 2e - 4$), the model learns a bad decision boundary, which also results in a bad performance. These results also imply that our reg-

Table 3: Natural and robust accuracies of GEAR across different λ and β on CIFAR10 dataset. Best results are in bold.

Dirichlet parameter	β=0.05		β=0.1		β=0.2		β=0.3	
Metric	Natural	PGD-20	Natural	PGD-20	Natural	PGD-20	Natural	PGD-20
$\lambda = 2e-2$	35.35	18.75	41.71	24.37	44.24	24.18	49.72	28.37
$\lambda = 2e-3$	**54.09**	**29.05**	**59.91**	**29.63**	**61.19**	**29.79**	**63.27**	**30.85**
$\lambda = 2e-4$	52.13	23.76	58.48	28.91	61.03	27.29	63.07	28.81

Table 4: Natural and robust accuracies across different number of clients $m = \{20, 50, 100\}$ on CIFAR10 dataset. Best results are in bold.

m	20		50		100	
Method	Natural	PGD-20	Natural	PGD-20	Natural	PGD-20
Fed_PGD	29.48	18.51	29.73	17.84	27.92	15.25
Fed_TRADES	30.23	18.12	21.53	14.63	24.15	14.24
Fed_MART	22.68	17.81	18.25	14.44	22.29	14.89
FAT	25.34	18.33	22.78	14.89	21.79	15.12
Ours	**44.89**	**21.84**	**40.91**	**18.57**	**40.26**	**17.92**

ularization loss can make the decision boundary smoother, and further benefits the model.

Moreover, we notice λ is more sensitive when β is very small (i.e., the labels of data on each client are more imbalanced). For example, when $\beta = 0.05$, the robust accuracy (against PGD-20 attack) drops from 29.05% to 23.76% as λ decreases from $2e-3$ to $2e-4$. We hypothesise that the reason is when the labels are more imbalanced, the model is harder to train and the regularization loss can better enhance the training. This implies that the proposed regularization loss plays an important role in enhancing FAT especially when the labels of data on each client are more imbalanced.

Evaluation on different number of clients To show the capability of our GEAR, we train our GEAR with different number of clients m. Table 4 reports the results across $m = \{20, 50, 100\}$ clients. GEAR achieves the best performance across all m, which further validate that GEAR can be applied in most practical cross-silo settings.

As m increases, the performance of all baselines decrease rapidly. We conjecture that the reason is that too many clients in FAT make the model harder to converge. However, the performance of our GEAR is relatively stable and only decreases slightly as m increases. This demonstrates that GEAR has a stable training process.

Evaluation on different FL framework We conduct experiments on different FL frameworks, i.e., FedAvg (McMahan et al. 2017), FedProx (Li et al. 2018), and Scaffold (Karimireddy et al. 2020). The results for different FAT methods on different FL frameworks are shown in Table 5. It can be observed that our GEAR has better natural accuracy and robust accuracy (against PGD-20 attack) than all baselines, which indicate our method can be well adapted to most FL frameworks.

Moreover, the performance of all methods on FedAvg is lower than the performance on FedProx and Scaffold. We conjecture this is because FedProx and Scaffold can alleviate the label skewness issue.

Table 5: Natural and robust accuracies across different FL framework on CIFAR10 dataset. Best results are in bold.

FL framework	FedAvg		FedProx		Scaffold	
Metric	Natural	PGD-20	Natural	PGD-20	Natural	PGD-20
NaiveFAT	53.23	26.22	54.96	28.15	54.99	27.91
FedPGD	47.22	26.50	48.45	28.39	48.70	28.46
FedTRADES	46.15	26.29	47.99	27.64	47.41	27.40
FedMART	25.68	18.16	27.32	19.58	27.04	19.52
GEAR (ours)	**59.91**	**29.63**	**61.08**	**31.22**	**61.07**	**30.79**

5 Conclusion

In this paper, we focus on the problem of adversarial robustness in FL, and propose a novel marGin-based fEderated Adversarial tRaining Approach called GEAR in order to maintain both natural accuracy and robust accuracy in FL. Our proposed GEAR can increase the margin between the training data of minority classes and the decision boundary by introducing a margin-based cross-entropy loss, and regularizes the decision boundary to be smooth by introducing a regularization loss, thus delivering a better decision boundary for the global model. To the best of our knowledge, this work is the first to investigate the impact of decision boundary in federated adversarial training (FAT) and delivers the best natural accuracy and robust accuracy by far. Extensive experiments across various datasets under different methods, different label skewness, different number of clients, and different FL protocols all validate the effectiveness of our proposed method. For example, on SVHN dataset, GEAR can improve the natural accuracy and robust accuracy (against FGSM attack) of the best baseline method (FedTRADES) by 20.17% and 10.73%, respectively.

References

Buda, M.; Maki, A.; and Mazurowski, M. A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106: 249–259. 3

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 1567–1578. 4

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. IEEE. 5

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2206–2216. 5

Cui, J.; Chen, C.; Lyu, L.; Yang, C.; and Li, W. 2021. Exploiting Data Sparsity in Secure Cross-Platform Social Recommendation. *Advances in Neural Information Processing Systems*, 34. 1

Ding, G. W.; Sharma, Y.; Lui, K. Y. C.; and Huang, R. 2020. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*. 4

Dong, J.; Cong, Y.; Sun, G.; Fang, Z.; and Ding, Z. 2021. Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1

Dong, J.; Cong, Y.; Sun, G.; Zhong, B.; and Xu, X. 2020. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4023–4032. 1

Hong, J.; Wang, H.; Wang, Z.; and Zhou, J. 2021. Federated Robustness Propagation: Sharing Adversarial Robustness in Federated Learning. *arXiv preprint arXiv:2106.10196*. 1

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. 7

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. In *Technical report*. 2, 5

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*. 5

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021. Federated Learning on Non-IID Data Silos: An Experimental Study. *arXiv preprint arXiv:2102.02079*. 5

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*. 7

Liu, Y.; Huang, A.; Luo, Y.; Huang, H.; Liu, Y.; Chen, Y.; Feng, L.; Chen, T.; Yu, H.; and Yang, Q. 2020. FedVision: An Online Visual Object Detection Platform Powered by Federated Learning. In *IAAI*. 1

Lyu, L.; and Chen, C. 2021. A Novel Attribute Reconstruction Attack in Federated Learning. *arXiv preprint arXiv:2108.06910*. 1

Lyu, L.; Yu, H.; Ma, X.; Chen, C.; Sun, L.; Zhao, J.; Yang, Q.; and Yu, P. S. 2022. Privacy and Robustness in Federated Learning: Attacks and Defenses. *arXiv preprint arXiv:2012.06337*. 2

Lyu, L.; Yu, H.; Zhao, J.; and Yang, Q. 2020. Threats to Federated Learning. In *Federated Learning*, 3–16. Springer. 6

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*. 1, 2, 5

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR. 1, 2, 3, 6, 7

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 5

Rice, L.; Wong, E.; and Kolter, J. Z. 2020. Overfitting in adversarially robust deep learning. In *ICML*. 5

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *AAAI Conference on Artificial Intelligence*. 1

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11). 4

Van Horn, G.; and Perona, P. 2017. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*. 3

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *ICLR*. 5

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. In *ICLR*. 5

Wu, C.; Wu, F.; Liu, R.; Lyu, L.; Huang, Y.; and Xie, X. 2021. FedKD: Communication Efficient Federated Learning via Knowledge Distillation. *arXiv preprint arXiv:2108.13323*. 1

Wu, C.; Wu, F.; Lyu, L.; Di, T.; Huang, Y.; and Xie, X. 2020. FedCTR: Federated Native Ad CTR Prediction with Multi-Platform User Behavior Data. *arXiv preprint arXiv:2007.12135*. 1

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2): 1–19. 2

Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; Hoang, N.; and Khazaeni, Y. 2019. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, 7252–7261. 5

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 7472–7482. PMLR. 4, 5

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. 2020. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *ICML*. 4

Zizzo, G.; Rawat, A.; Sinn, M.; and Buesser, B. 2020. Fat: Federated adversarial training. *arXiv preprint arXiv:2012.01791*. 1, 2, 3, 5