electronics

MDPI

Article

Bidirectional Corrective Model-Contrastive Federated Adversarial Training

Yuyue Zhang, Yicong Shi and Xiaoli Zhao *

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; m320121365@sues.edu.cn (Y.Z.); m320121104@sues.edu.cn (Y.S.)
* Correspondence: zhaoxiaoli@sues.edu.cn; Tel.: +86-1356-403-4656

Abstract: When dealing with non-IID data, federated learning confronts issues such as client drift and sluggish convergence. Therefore, we propose a Bidirectional Corrective Model-Contrastive Federated Adversarial Training (BCMCFAT) framework. On the client side, we design a category information correction module to correct biases caused by imbalanced local data by incorporating the local client's data distribution information. Through local adversarial training, more robust local models are obtained. Secondly, we propose a model-based adaptive correction algorithm in the server that leverages a self-attention mechanism to handle each client's data distribution information and introduces learnable aggregation tokens. Through the self-attention mechanism, model contrast learning is conducted on each client to obtain aggregation weights of corrected client models, thus addressing the issues of accuracy degradation and slow convergence caused by client drift. Our algorithm achieves the best natural accuracy on the CIFAR-10, CIFAR-100, and SVHN datasets and demonstrates excellent adversarial defense performance against FGSM, BIM, and PGD attacks.

Keywords: federated learning; adversarial attacks; defense models; contrastive learning; robustness

check for updates

Citation: Zhang, Y.; Shi, Y.; Zhao, X. Bidirectional Corrective Model-Contrastive Federated Adversarial Training. *Electronics* **2024**, *13*, 3745. https://doi.org/10.3390/electronics13183745

Academic Editor: Subir Halder

Received: 30 July 2024
Revised: 2 September 2024
Accepted: 11 September 2024
Published: 20 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The development of artificial intelligence technology mainly relies on data-driven techniques; without a huge number of data, it is hard to train models that match the requisite requirements. Data, being some of the most essential resources in today's society, face the huge dilemma of how to harness their value while respecting data privacy. Federated learning allows several users (clients) to collectively train a neural network model while retaining the privacy and security of their data [1–3]. However, training on local clients also increases the susceptibility of the global model in federated learning to adversarial attacks [4–6]. For instance, attackers can successfully trick the global model with a high success rate by adding minor adversarial perturbations to evaluate samples during the model inference stage. This has generated worries regarding the security and dependability of federated learning in real-world applications [7].

Non-independent and identically distributed (Non-IID) training data across multiple clients is a consequence of data scarcity, privacy protection restrictions, and the heterogeneity of data sources in federated learning [8–11]. As a result, local client training experiences optimization bias, which hinders the model's ability to converge to a stable optimum. Client drift is the term used to describe the phenomena wherein, as a result of differing local data distributions, the local models of several clients gradually diverge from one another. Differences in the local models might cause instability and inconsistencies when they are aggregated, which can make the aggregated global model function poorly. In federated learning systems, client drift is a significant issue as it gets worse when non-IID data are provided.

In classical machine learning, adversarial training (AT) is one of the best methods for making neural networks more resistant [12,13]. Differences in the local models might cause

instability and inconsistencies when they are aggregated, which can make the aggregated global model function poorly. In federated learning systems, client drift is a significant issue as it gets worse when non-IID data are provided. Adversarial training was added by Zizzo et al. to the federated learning client-side training procedure, enhancing the global model's defense against adversarial attacks [5,6]. Federated adversarial training (FAT) is a technique that needs more complicated models and a higher quantity of training data than normal training [14–17]. Federated adversarial training (FAT) shows poorer natural accuracy and slower convergence than regular federated learning, similar to classic adversarial training [5]. Furthermore, boosting adversarial robustness frequently results in a drop in natural accuracy [18], meaning that naturally trained models have far lower natural accuracy than conventional training [19]. By giving minority class data more weight during local adversarial training, CalFAT [20] overcomes the instability of adversarial training; nevertheless, it only addresses the problem of data imbalance inside clients, ignoring the model drift brought on by data heterogeneity among clients. FedDisco [21] modifies aggregation weights according to the size of each client's dataset and the variations between the local and global class distributions. However, it does not account for more complex federated learning scenarios, such as adversarial attacks occurring during the model inference phase.

This article proposes a Bidirectional Corrective Model-Contrastive Federated Adversarial Training (BCMCFAT) framework to address the low accuracy and instability of adversarial training caused by non-IID data in federated adversarial training. On the server side, class distributions from each client are leveraged, and a self-attention technique is applied to record distributional variances. Subsequently, local model-contrastive learning is used to identify the aggregation weights. On the client side, a bias correction module is developed to adapt model outputs depending on local data distributions.

Three points best describe our contributions:

- We present a server-side adaptive correction algorithm based on model contrast, which uses a self-attention mechanism to learn the differences between the distributions of each client and performs local model-contrastive learning to determine the corrected aggregation weights.
- We introduce a client-side local adaptive correction model that uses local data distributions to correct model outputs through a bias correction module.
- To evaluate the resilience and efficacy of our model, we check our technique on numerous datasets.

2. Related Work

2.1. Federated Learning

Federated learning is a decentralized machine learning architecture where numerous clients work together to build a common global model while keeping their data private [22]. The usual training procedure contains three major steps: (1) Server Initialization: The central server sets up the initial local models for the clients. (2) Local Training: Each client trains its model with local data and broadcasts the changed model back to the server. (3) Server Aggregation: The server aggregates the local models to build an updated global model, which is subsequently broadcast to the clients. (4) Client Model Update: Clients utilize the revised global model to revise their local models, setting the scene for the upcoming training session.

Currently, many studies are exploring ways to improve federated learning's handling of non-IID (non-independent and identically distributed) data [23–25]. These studies can be categorized into two main approaches: improving local training on clients and enhancing server aggregation.

In terms of improving local training on clients, FedProx [8] introduces a proximal term based on the distance between the global model and the local models. This term controls the training of local models during the federated learning process. On the other hand, SCAFFOLD [26] corrects local model updates by introducing control variates during

local training and synchronizing these variates across all clients. The gradient in the local training process is controlled by the difference between local and global control variates. However, SCAFFOLD has only been tested on a single dataset, lacking generalization tests on more diverse datasets.

For improving server aggregation, FedMA [27] employs a Bayesian non-parametric method for layer-wise matching and weighted averaging. FedAvgM [28] uses momentum to update the global model on the server. Most of these methods primarily allocate weights based on the size of the datasets. In contrast, FedDisco [21] identifies the difference between local and global class distributions as a more effective criterion for determining aggregation weights.

Most of the existing methods focus on classical federated learning scenarios. However, the complexity of non-IID data significantly increases the difficulty of adversarial defense. The BCMCFAT approach proposed in this paper addresses the robustness loss in federated learning caused by non-IID data. By accounting for both intra-client class distribution differences and inter-client data distribution disparities, BCMCFAT effectively mitigates challenges such as client drift and slow convergence in federated adversarial training. However, similar to traditional adversarial training methods, our approach relies on generating adversarial samples to enhance network robustness, which inevitably increases computational complexity and the resource consumption of clients in the federated learning process.

2.2. Self-Supervised Learning

Self-supervised learning has gained significant interest due to its capability to learn useful representations without relying on costly labeled data [29–31]. Specifically, contrastive learning has become a well-liked method in the visual domain. The goal of this approach is to maximize the distance between representations of distinct pictures (negative pairs) and minimize the similarity between representations of enhanced perspectives of the same image (positive pairs) within the feature space.

The most classic contrastive learning framework is SimCLR [32]. In SimCLR, each image from the original dataset generates two augmented samples, x_i and x_j. Each augmented sample is first transformed into a feature representation through a feature extraction network $f(\cdot)$ and then projected into a feature space by a projection head $g(\cdot)$. The NT-Xent contrastive loss is applied to the projection vectors $g(f(\cdot))$. Specifically, given $2N$ augmented samples, the contrastive loss can be defined as follows:

$$l_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{1}$$

where $\text{sim}(\cdot)$ denotes the cosine similarity function, z_i represents the feature of the augmented image, τ is the temperature parameter, and $1_{[k \neq i]}$ is an indicator function that equals 1 when $k \neq i$.

To solve the difficulty of effectively storing and exploiting a high number of negative samples in large-scale datasets, MoCo [33] proposes momentum-updated encoders and a dynamic negative sample queue. This approach allows for contrastive learning in a larger context, enabling the model to learn richer representations. BYOL [34] further resolves the dependency on negative samples in contrastive learning by using an exponential moving average technique for updating the target net based on the online network, allowing the self-supervised learning process to track useful signals.

Recent research has explored combining contrastive learning with federated learning. van Berlo et al. [35] first introduced the concept of federated unsupervised representation learning based on autoencoders, but their study did not consider the non-IID nature of federated data. Zhang et al. [36] advocated employing a common dictionary module to overcome the non-IID problems in federated learning. Zhuang et al. [37] combined the classic self-supervised framework BYOL with federated learning and introduced adaptive

updates for local predictors to further improve performance. Zhuang et al. [38] further studied the essential components of federated self-supervised learning frameworks and solved the non-IID issue using the Exponential Moving Average for adaptive updates to local models. Li et al. [39] aggregated online networks from client nodes using FedAVG and then improved accuracy by applying an incremental moving average update approach to the target network. This study covers supervised learning settings and presents a model comparison to learn distinct data distributions among clients with the goal of achieving optimal aggregation weights, while previous research has concentrated on unsupervised learning environments.

2.3. Adversarial Training

Adversarial examples are samples constructed by introducing tiny changes to original samples to deceive machine learning models [40]. Malicious users, hackers, or other nefarious actors may create adversarial examples to mislead models and achieve illicit objectives, which significantly affects the practical applications of deep learning networks.

To defend against the potential threats posed by adversarial examples, adversarial defense has increasingly attracted the attention of researchers. One of the best strategies for adversarial defense is adversarial training, which works on the basis of introducing adversarial cases into the model to increase its robustness. Common methods for generating adversarial examples include FGSM [41], BIM [42], and PGD [43].

For instance, FGSM generates adversarial examples as follows:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)) \tag{2}$$

where ϵ is the magnitude of the perturbation, sign is the sign function that returns the sign of the gradient, L is the loss function, θ denotes the model parameters, x is the clean input image, and y is the input label. By applying a single-step perturbation proportionate to the gradient of the loss function with respect to the input image, FGSM creates adversarial instances. This method is computationally efficient but may not always produce the most effective adversarial examples. BIM extends FGSM by applying iterative updates to generate adversarial examples:

$$x^{(t+1)} = \text{clip}_{x,\delta}\left(x^t + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x^t, y))\right) \tag{3}$$

where δ is the perturbation constraint, ϵ is the magnitude of the pixel update per iteration, sign is the sign function that returns the sign of the gradient, L is the loss function, θ denotes the model parameters, x^t is the image at iteration t, and y is the input label. BIM applies FGSM's perturbation multiple times, with each step clipping the result to ensure that the adversarial example remains within the allowed perturbation range. This iterative approach often produces more robust adversarial examples compared to a single-step FGSM attack.

PGD further generalizes the iterative approach:

$$x^{(t+1)} = \prod_{x+\delta}\left(x^t + \alpha \cdot \text{sign}(\nabla_x L(\theta, x^t, y))\right) \tag{4}$$

where δ represents the perturbation constraint, α is the step size for each iteration, sign is the sign function that returns the sign of the gradient, L is the loss function, θ denotes the model parameters, x^t is the image at iteration t, and y is the input label. PGD applies a fixed step size perturbation iteratively and projects the result to ensure that it remains within the allowed perturbation range. This approach is notable for its ability in producing strong adversarial cases and is often used for testing the resilience of models.

The model learns to recognize and defend against tiny adjustments that aim to mislead its assessments by adding these adversarial events to the usual training dataset and training the model on the augmented dataset. The main training objective of adversarial training can be summarized by the following formula:

$$\min_{w} \mathbb{E}_{(x,y)\sim D}[\max_{\|\delta\|\leq\varepsilon} L(x+\delta, y, w)] \tag{5}$$

where w represents the model parameters, δ denotes the perturbation, ϵ is the perturbation bound, x is the clean input image, and y is the input label.

Research on enhancing the adversarial robustness of federated learning is currently quite limited. Directly incorporating adversarial training into federated learning is a straightforward solution. Zizzo et al. [5] were the first to apply adversarial training methods in federated learning, developing MixFAT to improve the robustness of the global model. MixFAT applies adversarial training to only a portion of local data to enhance the stability and convergence of federated learning. Chen et al. [20] discovered that there is a significant accuracy gap between MixFAT and centralized adversarial training, especially when data heterogeneity is strong. They proposed CalFAT, which addresses data heterogeneity by increasing the weight of minority classes in client data. Hong et al. [6] introduced an effective robustness propagation method using batch normalization to address the imbalance in computational resources among participants. Our BCMCFAT is designed for scenarios where participants have average computational resources but face extreme imbalances in data distribution between clients. Compared to existing federated adversarial training methods, BCMCFAT directly focuses on the distribution of non-IID data that causes issues such as client drift, by calibrating the biases introduced by non-IID data at both the client and server levels.

3. Method

3.1. Problem Statement

Robust federated learning intends to cooperatively train a universal robust model θ with N participants (clients). Each client i holds local data D^i, which is not accessible to the server or anyone else to ensure confidentiality of the information. The data across clients are typically non-IID, showing significant distributional differences. As a result, no single client can independently train a well-performing model. The purpose of this work is to leverage image data from multiple parties to learn a universal model while ensuring privacy. The global objective function for robust federated learning is defined as follows:

$$\underset{\theta}{\operatorname{argmin}}\, L(\theta) = \sum_{i=1}^{N} L_i(\theta) \tag{6}$$

where $L_i(\theta) = \mathbb{E}_{(x,y)\sim D_i}\left[\max_{\|\delta\|\leq\epsilon} L(x+\delta, y, \theta)\right]$ denotes the adversarial loss for client i.

From a statistical standpoint, in traditional federated adversarial training, the training objective for each client i is to compute the local class probability $p_i(y \mid x)$. The local class probability can be parameterized using θ_i^*:

$$p_i(y \mid x) = \hat{p}(y \mid x; \theta_i^*) = \sigma^y(f_{\theta_i^*}(x)) \tag{7}$$

Chen et al. [20] found that if the intra-class distributions among clients are imbalanced, let θ_i be the maximum likelihood estimate of θ_i^* for client i given the local data in Equation (5), and s^2 be the sample variance of the local model parameters. Then, s^2 almost surely converges to a non-zero constant:

$$s^2 \xrightarrow{\text{a.s.}} (s^*)^2 \neq 0 \tag{8}$$

where $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence. $(s^*)^2$ quantifies the heterogeneity of the true parameters $\{\theta_i^* \mid i \in [m]\}$, reflecting the differences in class probabilities among clients. Equation (8) suggests that when the label distributions across clients are skewed, the local models in previous FAT methods exhibit significant heterogeneity. During aggregation, this heterogeneity may hinder the global model's convergence and perhaps cause divergence.

This research offers a Bidirectional Corrective Model-Contrastive Federated Adversarial Training framework to overcome the instability and low accuracy caused by heterogeneous data in federated adversarial training (FAT). The heterogeneity of the data impacts federated model training in two ways. First, the heterogeneity of clients' data leads to poor training performance of local models. Second, the heterogeneous local models trained on heterogeneous data affect the performance of the global model during aggregation. Therefore, this paper introduces an adaptive correction algorithm based on model contrast for aggregating models from various clients on the server. Additionally, a correction module incorporating data class distribution is proposed to assist in training client models.

3.2. Model-Contrast-Based Adaptive Correction Algorithm

In federated learning, client drift can lead to model aggregation that deviates from the global optimum when conducted only on the basis of dataset quantity. This work suggests a model-contrast-based adaptive correction technique on the server side to overcome this difficulty. The server employs a self-attention mechanism to learn and adjust for data distribution differences across clients, thereby mitigating client drift. The structure of the proposed model-contrast-based adaptive correction algorithm is illustrated in Figure 1.

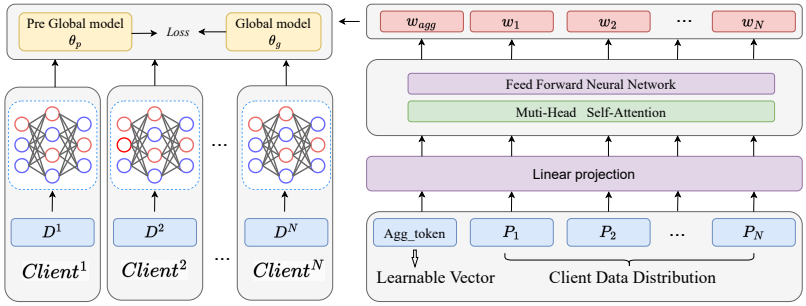


Figure 1. Overall structure of the Bidirectional Corrective Model-Contrastive Federated Adversarial Training.

Client i uses its local data D_i to train a local model in each round of federated learning, after which it uploads the model to the server. The server does not have access to the clients' data in order to protect data privacy. This study uses model contrast for global model aggregation on the server, inspired by contrastive learning. The following is the definition of the model contrast loss on the server:

$$l_{\text{con}} = -\log \frac{\exp(\text{sim}(\theta_g, \theta_p)/\tau)}{\sum_{i=1}^{N} 1_{[k \neq i]} \exp(\text{sim}(\theta_i, \theta_k)/\tau)} \tag{9}$$

where τ is the temperature constant, θ_g represents the global model aggregated in the current round using the learnable aggregation vector $\text{Agg}_{\text{token}}$ with output weights w_{agg}, θ_p is the global model aggregated in the previous round, and θ_i denotes the global model aggregated with the output weights w_i from the data distribution information of client i. Specifically, the process for generating the aggregation weight matrix $W = \{w_{\text{agg}}, w_1, w_2, \ldots, w_N\}$ can be expressed as follows:

$$W = f_{\theta_\alpha}(\text{Agg_token}, P_1, P_2, \ldots, P_N) \tag{10}$$

where θ_α is the self-attention network shown in Figure 1, $\text{Agg}_{\text{token}}$ is the learnable aggregation vector, and P_i represents the data distribution of client i.

To ensure fairness in federated learning, this work suggests utilizing the output of the learnable aggregation vector (Agg_token), denoted by w_{agg}, as the aggregation weights for global model aggregation. Specifically, the aggregation vector is linearly mapped with each client's data distribution information P_i. Adaptive corrected aggregation weights are

obtained by learning the differences between client data distributions through a multi-head self-attention technique. The global model θ_g is then aggregated using the output w_{agg} of the aggregation vector.

The specific algorithmic process for the model-contrast-based adaptive correction algorithm is shown in Algorithm 1.

Algorithm 1 Model-Contrast-Based Adaptive Correction Algorithm

Require: Federated training rounds R; number of clients N; server training iterations E; client data distribution information $\{P_i\}_{i=1}^{N}$; server aggregation weight calculation model θ_a
Ensure: Global model θ_g

Initialize θ_g^0; $\theta_p \leftarrow$ None
for round $r = 0, 1, \ldots, R-1$ **do**
  for client $i = 0, 1, \ldots, N-1$ **do**
    Update local model $\theta_i^r \leftarrow \theta_g^r$
    Train local model θ_i^r
    Upload local model θ_i^r to server
  end for
end for
if $\theta_p =$ None **then**
  $\theta_g^{(r+1)} \leftarrow \frac{1}{N}\sum_{i=1}^{N}\theta_i^r$
  Continue
end if
for iteration $e = 0, 1, \ldots, E-1$ **do**
  Compute aggregation weight matrix $W = \{w_{\text{agg}}, w_1, w_2, \ldots, w_N\}$ by Equation (10)
  Compute global model $\{\theta_g, \theta_1, \ldots, \theta_N\} \cdot W$
  Compute global model θ_g using output weight of aggregation vector
  Calculate model contrast loss l_{con} using Equation (9)
  Backpropagate and update aggregation model θ_a
end for
Return Global model θ_g

3.3. Local Adaptive Correction Model

The data distribution of local clients is a fundamental cause of local training drift. Inspired by this observation, this paper designs a local adaptive correction model that corrects the network output through an inter-class difference-aware module, as shown in Figure 2.

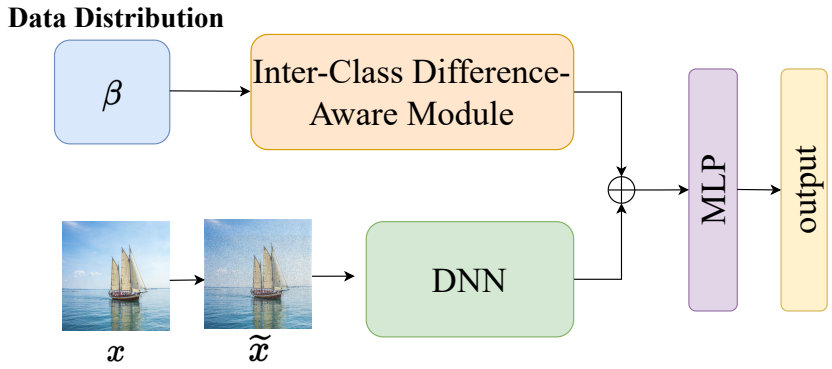


Figure 2. Overall structure of the local adaptive correction model.

For each client i, we can obtain $\hat{q}_i(y \mid x; \theta_i)$ by optimizing the following objective:

$$\min_{\theta_i} \frac{1}{n_i} \sum_{j=1}^{n_i} l_{\text{ce}}(f_{\theta_i}(\tilde{x}_{ij}); y_{ij}) \tag{11}$$

where θ_i represents the model parameters for client i, n_i is the number of samples in client i's dataset, $l_{\text{ce}}(\cdot;\cdot)$ is the loss function (cross-entropy), and $\tilde{x}_{ij}$ is the adversarial example of x_{ij}. The standard cross-entropy loss is defined as follows:

$$l_{\text{ce}}(f_{\theta_i}(\tilde{x}_{ij})) = -\log \sigma^{y_{ij}}(f_{\theta_i}(\tilde{x}_{ij})) \tag{12}$$

To leverage inter-class difference information for correcting model drift, this paper designs an inter-class difference-aware module to adjust the model outputs. The correction is represented by the following formula:

$$f_{\theta_i}(\tilde{x}_{ij}; \beta_i) = f_{d_i}(\tilde{x}_{ij}) + f_{c_i}(\beta_i) \tag{13}$$

where f_{d_i} represents the deep neural network of client i, and f_{c_i} denotes the inter-class difference-aware module of client i. The function f_{c_i} is defined as follows:

$$f_{c_i}(\beta_i) = f_{l_i}(\log(\beta_i + \mu)) \tag{14}$$

where f_{l_i} represents the linear network of client i, and $\mu > 0$ is a small constant added for the stability of the logarithm function, set to 1×10^{-7}.

Algorithm 2 describes the training process of the local adaptive correction model proposed in this section.

Algorithm 2 Client-Based Model Contrast Adaptive Correction Algorithm

Require: Global model at round r, θ_g^r; Number of clients N; Number of client training epochs E; Client data distribution information β_i;
Ensure: Client local model θ_i^{E-1};
1: Download local model $\theta_i^r \leftarrow \theta_g^r$
2: **for** $e = 0, 1, \ldots, E-1$ **do**
3: Generate adversarial samples $\tilde{x}$
4: Forward propagate adversarial sample $\tilde{x}$ to obtain pre-correction output $\tilde{z}$
5: Obtain corrected output z by Equation (14)
6: Compute gradient and update local model $\theta_i^{e+1} \leftarrow \theta_i^e + \eta \nabla l_{\text{ce}}(z; y)$
7: **end for**
8: Return Local model θ_i^{E-1}

4. Results

4.1. Datasets and Models

In this experiment, a CNN with two convolutional layers and one fully connected layer is utilized as the classification model. The experiments utilize the following datasets:

- CIFAR-10 [44] With 60,000 32 × 32 color pictures divided into ten categories, CIFAR-10 is a popular image classification dataset. Ten thousand test photos and fifty thousand training images make up the dataset.
- CIFAR-100 [44] With 60,000 32 × 32 color pictures divided into 100 categories, CIFAR-100 expands on CIFAR-10. Ten thousand test photographs and fifty thousand training images are also included.
- SVHN [45] Digits taken from Google Street View pictures make up the Street View House Numbers (SVHNs) image categorization collection. There are 26,032 photos in the test set and 73,257 images in the training set out of the total of 32 × 32 color images.

The hyperparameters are shown in Table 1. To guarantee quick convergence, the learning rate is first set to 0.1 and is progressively lowered during training to optimize the model's parameters. The batch size is set to 128 to balance computational efficiency and the amount of information from model updates. The local training epochs are set to 5 to prevent excessive local models' deviation from the global model while reducing the communication frequency. The number of global communication rounds is set to 100, with the final number determined based on performance on the validation set to ensure convergence. The cross-entropy loss function is utilized for classification tasks, while the SGD optimizer is used to minimize computing cost. The weight decay is set to 10^{-4} to prevent overfitting without overly restricting model complexity. The ReLU activation function is chosen to avoid the vanishing gradient problem. Communication is performed after every round of training to ensure timely and consistent model parameter updates.We have implemented BCMCFAT using the PyTorch deep learning framework in Python. To simulate federated learning, we have trained each client on an NVIDIA RTX 3090 GPU.

Table 1. Neural network hyperparameters for federated learning.

Hyperparameter	Description	Value
Learning Rate	Step size for local optimization	0.1
Batch Size	Number of samples per client batch	128
Local Epochs	Number of local training epochs per round	5
Global Rounds	Number of communication rounds	100
Optimizer	Optimization algorithm for local updates	SGD
Loss Function	Function to compute loss	Cross-Entropy
Weight Decay	Regularization term for local models	10^{-4}
Activation Function	Non-linear function used in the network	ReLU
Communication Frequency	Frequency of model updates aggregation	Every 1 round

4.2. Experimental Setup

We compare the proposed Bidirectional Corrective Model-Contrastive Federated Adversarial Training framework with a number of cutting-edge federated adversarial training algorithms, such as FedPGD [43], FedTRADES [12], FedMART [13], FedGAIRAT [17], FedRBN [6], and CalFAT [20], in order to assess its performance under data heterogeneity. The results are discussed in Section 4.3.

We simulate non-IID data distributions with label skew across distinct clients using a Dirichlet distribution to emulate real-world federated learning settings. The Dirichlet [10] distribution's probability density function is provided by

$$f(x;\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{15}$$

where $x = \{x_1, x_2, \ldots, x_k\}$ is a probability vector such that $0 \leq x_i \leq 1$ and $\sum_{i=1}^{K} x_i = 1$. The Γ function, also known as the Gamma function, is expressed as follows:

$$\Gamma(\alpha_i) = \int_0^{\infty} t^{\alpha_i - 1} e^{-t}\, dt \tag{16}$$

In a federated learning environment, sampling from a Dirichlet distribution on IID data can generate non-IID data. Given an original IID dataset with N classes, the heterogeneous local training datasets for each client can be obtained by sampling the original dataset with a probability parameter vector $q = (q_i \geq 0, i \in [1, N])$ to create non-IID datasets. The non-IID datasets can be generated for each local client using the formula:

$$q \sim \mathrm{Dir}(\alpha p) \tag{17}$$

where α is a hyperparameter that regulates the level of data heterogeneity for each client, and p is a probability vector that represents the prior probability distribution of N classes.

By adjusting the hyperparameter α, datasets with varying degrees of heterogeneity can be generated. As $\alpha \to 0$, the data distribution becomes more skewed, with samples drawn predominantly from a single class. Conversely, the data distribution becomes more uniform as $\alpha \to \infty$, with samples distributed more uniformly throughout all classes. For the CIFAR-10 and CIFAR-100 datasets, non-IID data are created using $\alpha = 0.1$ and given at random to each client in this study. For example, the data distributions for each client under various α settings in the CIFAR-10 dataset are shown in Figure 3.

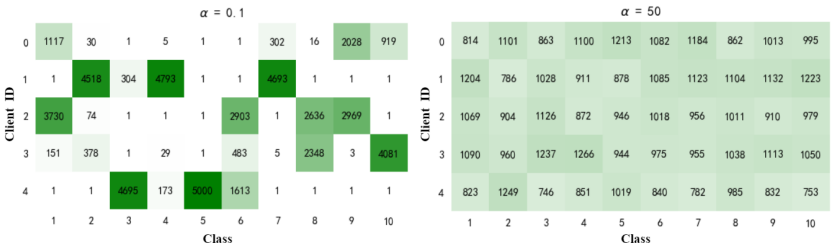


Figure 3. Diagram illustrating data distributions under different α parameters for the CIFAR-10 dataset.

4.3. Comparative Experiments

In this part, we test the model's performance under natural settings (natural accuracy) and its resilience under various adversarial attacks (robustness accuracy). The comparative experiments use datasets sampled from the Dirichlet distribution to simulate federated learning with non-IID data, distributed across 5 clients, with 100 communication rounds. Two convolutional layers and one fully linked layer make up the models. The choice of FGSM, BIM, and PGD for evaluating adversarial defense algorithms is primarily motivated by the need to cover a range of attack complexities and strengths. FGSM is used for baseline testing and initial evaluation, assessing the defense algorithm's resistance to simple adversarial attacks due to its simplicity and efficiency. BIM is employed to test the defense algorithm's robustness against iterative attacks, examining its performance under continuous optimization. Its iterative nature and increased attack strength make it valuable for evaluating defenses against more sophisticated adversarial samples. PGD is selected to evaluate the defense algorithm's effectiveness against the strongest adversarial attacks, as it represents one of the most challenging and advanced adversarial attack methods. By using these three attack models, a comprehensive evaluation of the defense algorithm's performance is achieved, ensuring its robustness across different types of adversarial attacks. The adversarial attacks are configured as follows: FGSM (Fast Gradient Sign Method) with a perturbation limit of $\epsilon = 8/255$; BIM (Basic Iterative Method) with a perturbation limit of $\epsilon = 8/255$, attack step size of $8/255$, and attack steps of 20; PGD20 (Projected Gradient Descent) with a perturbation limit of $\epsilon = 8/255$, attack step size of $2/255$, and attack steps of 20.

Tables 2–4 show a comparison of the natural and robust accuracy of our proposed technique with previous federated learning adversarial defensive systems on the non-IID CIFAR-10, CIFAR-100, and SVHN datasets.

The approach we propose beats most current algorithms on clean data. On the CIFAR-10 and CIFAR-100 datasets, our method achieved accuracies of 64.75% and 45.12%, respectively, which are comparable to or slightly better than the best-performing methods. On the SVHN dataset, our method achieved an accuracy of 84.27%, surpassing all other methods. These results indicate that local adaptive correction model effectively mitigates the bias caused by data imbalance in federated learning environments, enabling the model to better exploit clean data for learning.

Under numerous adversarial attacks (such as PGD20, FGSM, and BIM), our method remains competitive across different datasets. For instance, our method achieved 31.52% accuracy under the PGD20 attack and 40.61% accuracy under the FGSM attack on the

CIFAR-10 dataset, which are comparable to or slightly better than other methods. On the CIFAR-100 dataset, our method achieved 15.46% accuracy under PGD20 and 21.02% accuracy under FGSM, also demonstrating relatively good performance. On the SVHN dataset, our method achieved 44.02% accuracy under PGD20 and 57.62% accuracy under FGSM, showing stronger robustness. This indicates that the server-side calibration strategy plays a crucial role in mitigating the negative impact of adversarial samples on model performance, effectively enhancing the defense capabilities of the global model.

In a few cases, the robust accuracy of our method shows minor differences compared to other methods and even performs slightly worse on some metrics. The observed results may be related to several factors. Firstly, our proposed bidirectional correction mechanism is specifically built to address the non-IID characteristics of federated data, which inherently increases complexity. While this approach is effective in certain scenarios, its reliance on local client correction strategies may not fully address various adversarial threats, especially when facing complex attacks like PGD20 or BIM. Additionally, the server-side adaptive calibration algorithm based on self-attention mechanisms might introduce a more subtle trade-off between robustness and generalization, leading to varying performance under different types of attacks.

Table 2. Accuracy of different defense methods on the CIFAR-10 dataset.

Defense Method	Clean Accuracy	PGD20	FGSM	BIM
MixFAT	53.35	26.27	29.14	26.31
FedPGD	46.96	26.74	29.14	26.59
FedTRADES	46.06	26.31	27.75	26.32
FedMART	25.67	18.10	18.50	18.21
FedGAIRAT	48.42	27.20	29.30	26.55
FedRBN	48.42	26.30	26.87	26.25
CalFAT	64.69	31.12	35.03	31.50
Ours (CNN)	64.75	31.52	40.61	31.52

Table 3. Accuracy of different defense methods on the CIFAR-100 dataset.

Defense Method	Clean Accuracy	PGD20	FGSM	BIM
MixFAT	34.43	14.36	15.69	14.60
FedPGD	33.96	14.67	16.07	14.68
FedTRADES	29.55	14.30	15.01	14.11
FedMART	19.96	12.83	13.00	12.91
FedGAIRAT	34.92	14.90	16.18	15.37
FedRBN	28.55	14.15	14.69	13.41
CalFAT	44.57	15.21	17.63	15.60
Ours	45.12	15.46	21.02	15.32

Table 4. Accuracy of different defense methods on the SVHN dataset.

Defense Method	Clean Accuracy	PGD20	FGSM	BIM
MixFAT	19.57	19.75	19.61	19.66
FedPGD	19.55	19.52	19.33	19.37
FedTRADES	56.96	34.90	36.92	35.15
FedMART	19.85	19.79	19.94	19.71
FedGAIRAT	58.41	36.69	38.30	36.52
FedRBN	53.88	32.32	34.48	32.52
CalFAT	84.15	41.68	48.38	42.40
Ours	84.27	44.02	57.62	44.03

4.4. Ablation Study

4.4.1. Impact of Non-IID Data on Adversarial Training

In this section, the non-IID CIFAR-10 data distribution in federated learning is replicated using the Dirichlet function with two possibilities of α ($\alpha = 0.1$ and $\alpha = 50$), as illustrated in Figure 3. Figure 4 illustrates the effect of non-IID data on federated adversarial training with different α settings on the CIFAR-10 dataset.

According to the experimental results, the data distribution approaches independent and identical distribution (IID) at $\alpha = 50$. This leads to more stable federated adversarial training and improved accuracy of the model. Conversely, when $\alpha = 0.1$, the data distribution becomes more non-IID. In this case, the severe heterogeneity of the data leads to lower final accuracy and instability during federated adversarial training. Therefore, it may be concluded that the level of heterogeneity in non-IID data in federated learning has a direct impact on model training stability and final accuracy. The stronger the heterogeneity (i.e., the lower the α), the more pronounced the impact. The proposed Bidirectional Corrective Model-Contrastive Federated Adversarial Training framework intends to reduce the negative effects of non-IID data on federated adversarial training.

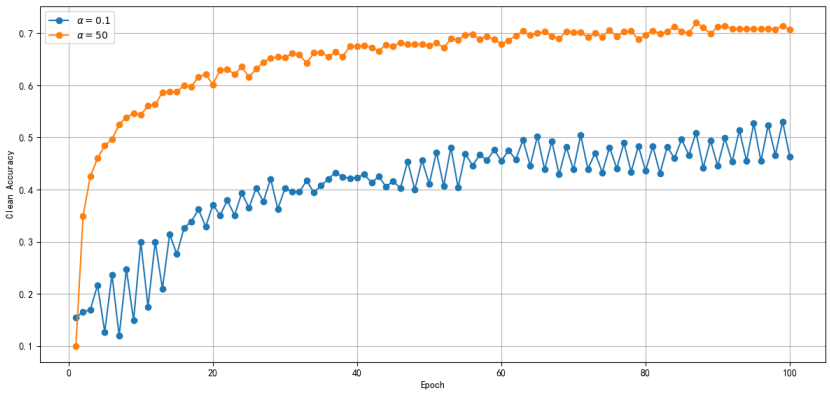


Figure 4. Federated adversarial training under different α parameters on the CIFAR-10 dataset.

4.4.2. Ablation Study on Client- and Server-Side Correction Models

The Bidirectional Corrective Model-Contrastive Federated Adversarial Training framework consists of two main components: (1) the Client-side Adaptive Correction Module (Local Correction, LC); and (2) the Server-side Model-Contrastive Adaptive Correction Algorithm (MC). Utilizing the CIFAR-10 dataset, an ablation research was carried out to confirm the efficacy of these two components. The ablation study used Dirichlet function sampled datasets to simulate non-IID data in federated learning, distributed to 5 clients, with 100 communication rounds. The models used consisted of two simple convolutional layers and one linear layer. Considering that PGD is a strong iterative attack, the results from ablation tests under PGD are representative. Therefore, we used PGD attack results as the key indicator of robustness.

Table 5 displays the ablation study's findings. "FedPGD" denotes the standard federated PGD adversarial training, which does not include the server-side Model-Contrastive Adaptive Correction (MC) module or the client-side adaptive correction (LC) module. "FedPGD+MC" indicates the standard federated PGD adversarial training with the addition of the server-side Model-Contrastive Adaptive Correction (MC) module. "FedPGD+LC" represents the standard federated PGD adversarial training with the inclusion of the client-side adaptive correction (LC) module. "BCMCFAT" refers to the proposed Bidirectional Corrective Model-Contrastive Federated Adversarial Training framework.

Table 5 shows that comparing "FedPGD+MC" with "FedPGD" reveals that using the model-contrastive adaptive correction module (MC) on the server improves the aggregation of heterogeneous robustness models, thus enhancing performance compared to standard FedPGD. Comparing "FedPGD+LC" to "FedPGD", it is clear that the client-side adap-

tive correction module (LC) enhances local model training. The textcolorred"BCMCFAT" method outperforms both "FedPGD+MC" and "FedPGD+LC", demonstrating that the combination of both modules is effective in federated adversarial training.

Figure 5 demonstrates the training stages of the studies stated previously. It can be observed that both "FedPGD+MC" and "FedPGD+LC" show better stability compared to "FedPGD." Among these, the curve for "BCMCFAT" is the best, demonstrating both superior stability and the highest final accuracy. Therefore, the inclusion of both MC and LC modules contributes to the stability of federated adversarial training and achieves better performance.

Table 5. Module ablation experiments on the CIFAR-10 dataset.

Defense Method	Clean Accuracy	PGD20
FedPGD	47.26	22.31
FedPGD+LC	48.46	23.90
FedPGD+MC	60.61	29.87
Ours	62.28	31.23

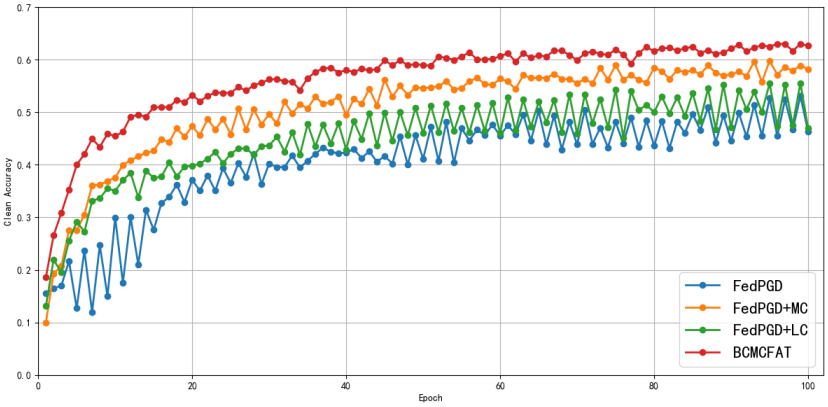


Figure 5. Training process for ablation experiments on the CIFAR-10 dataset.

5. Conclusions

We provide a unique strategy for adversarial defense in federated learning under non-IID scenarios, based on a server-side model-contrastive correction framework and a client-side adaptive correction module. On the server side, model contrast is performed by leveraging differences in data distributions among clients to correct aggregation weights and prevent model drift. Simultaneously, on the client side, model outputs are corrected by utilizing inter-class differences in client data distributions, addressing instability issues in adversarial training caused by non-IID data. Comparative experiments across various datasets and baseline methods demonstrate that this approach significantly enhances adversarial training performance in non-IID settings. This method effectively addresses adversarial security challenges posed by heterogeneous client data in real-world scenarios, making it broadly applicable to multi-client collaborative adversarial training environments. Specifically, our method is applicable to collaborative scenarios in intelligent driving systems. In such cases, federated adversarial training can prevent malicious attackers from deliberately designing image perturbations to interfere with vehicle perception systems, which could otherwise misinterpret dangerous situations (such as pedestrians crossing the road) as safe, leading to potential safety risks. Although our method performs exceptionally well on image data, the techniques for generating adversarial examples (such as PGD, FGSM, and BIM) are specifically designed for image data, limiting the current application of our approach to this domain. Further research is needed to adapt these techniques for other types of data, such as tabular data. Additionally, addressing the challenge of providing robust and stable algorithms in environments with a large number of clients and heterogeneous computational capabilities is an important direction for future work. We also plan to

further refine the federated adversarial training framework in future research, including integrating privacy protection technologies and improving communication efficiency to tackle additional real-world challenges.

Author Contributions: Conceptualization, Y.Z.; Methodology, Y.Z., Y.S. and X.Z.; Resources, X.Z.; Software, Y.Z. and X.Z.; Validation, Y.Z.; Writing—original draft, Y.Z. and Y.S.; Writing—review and editing, Y.Z., Y.S. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China (62376151), Shanghai Science and Technology Commission (22DZ2205600).

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.Y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
2. Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 8432–8440.
3. Zhang, J.; Li, Z.; Li, B.; Xu, J.; Wu, S.; Ding, S.; Wu, C. Federated learning with label distribution skew via logits calibration. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 26311–26329.
4. Lyu, L.; Yu, H.; Ma, X.; Chen, C.; Sun, L.; Zhao, J.; Yang, Q.; Philip, S.Y. Privacy and robustness in federated learning: Attacks and defenses. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 8726–8746 [CrossRef] [PubMed]
5. Zizzo, G.; Rawat, A.; Sinn, M.; Buesser, B. FAT: Federated Adversarial Training. In Proceedings of the Annual Conference on Neural Information Processing Systems, Online, 6–12 December 2020.
6. Hong, J.; Wang, H.; Wang, Z.; Zhou, J. Federated robustness propagation: Sharing adversarial robustness in federated learning. *arXiv* **2021**, arXiv:2106.10196.
7. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
8. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
9. Panchal, K.; Choudhary, S.; Mitra, S.; Mukherjee, K.; Sarkhel, S.; Mitra, S.; Guan, H. Flash: Concept drift adaptation in federated learning. In Proceedings of the International Conference on Machine Learning. PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 26931–26962.
10. Guo, Y.; Tang, X.; Lin, T. Fedbr: Improving federated learning on heterogeneous data via local learning bias reduction. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 12034–12054.
11. Chen, H.; Hao, M.; Li, H.; Chen, K.; Xu, G.; Zhang, T.; Zhang, X. GuardHFL: Privacy guardian for heterogeneous federated learning. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 4566–4584.
12. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 5 May 2017; pp. 39–57.
13. Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
14. Chen, C.; Zhang, J.; Xu, X.; Lyu, L.; Chen, C.; Hu, T.; Chen, G. Decision boundary-aware data augmentation for adversarial training. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1882–1894. [CrossRef]
15. Wang, D.; Jin, W.; Wu, Y.; Khan, A. Atgan: Adversarial training-based gan for improving adversarial robustness generalization on image classification. *Appl. Intell.* **2023**, *53*, 24492–24508. [CrossRef]
16. Carmon, Y.; Raghunathan, A.; Schmidt, L.; Duchi, J.C.; Liang, P.S. Unlabeled data improves adversarial robustness. In Proceedings of the NIPS'19: 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
17. Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; Kankanhalli, M. Geometry-aware Instance-reweighted Adversarial Training. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
18. Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; Madry, A. Robustness May Be at Odds with Accuracy. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

19. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 2206–2216.
20. Chen, C.; Liu, Y.; Ma, X.; Lyu, L. Calfat: Calibrated federated adversarial training with label skewness. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 3569–3581.
21. Ye, R.; Xu, M.; Wang, J.; Xu, C.; Chen, S.; Wang, Y. Feddisco: Federated learning with discrepancy-aware collaboration. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 39879–39902.
22. Arazzi, M.; Conti, M.; Nocera, A.; Picek, S. Turning privacy-preserving mechanisms against federated learning. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, Copenhagen, Denmark, 26–30 November 2023; pp. 1482–1495.
23. Zhu, Z.; Hong, J.; Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 12878–12889.
24. Collins, L.; Hassani, H.; Mokhtari, A.; Shakkottai, S. Exploiting shared representations for personalized federated learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 2089–2099.
25. Jiang, M.; Wang, Z.; Dou, Q. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 22 February–1 March 2022; Volume 36, pp. 1087–1095.
26. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. Scaffold: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 5132–5143.
27. Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; Khazaeni, Y. Federated learning with matched averaging. *arXiv* **2020**, arXiv:2002.06440.
28. Hsu, T.M.H.; Qi, H.; Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv* **2019**, arXiv:1909.06335.
29. Wen, X.; Zhao, B.; Zheng, A.; Zhang, X.; Qi, X. Self-supervised visual representation learning with semantic grouping. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16423–16438.
30. Feng, C.; Patras, I. Adaptive soft contrastive learning. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 2721–2727.
31. Poklukar, P.; Polianskii, V.; Varava, A.; Pokorny, F.T.; Kragic, D. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In Proceedings of the ICLR 2022-International Conference on Learning Representations, Virtual Event, 25–29 April 2022.
32. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 1597–1607.
33. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
34. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21271–21284.
35. van Berlo, B.; Saeed, A.; Ozcelebi, T. Towards federated unsupervised representation learning. In Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking, New York, NY, USA, 27 April 2020; pp. 31–36.
36. Zhang, F.; Kuang, K.; Chen, L.; You, Z.; Shen, T.; Xiao, J.; Zhang, Y.; Wu, C.; Wu, F.; Zhuang, Y.; et al. Federated unsupervised representation learning. *Front. Inf. Technol. Electron. Eng.* **2023**, *24*, 1181–1193. [CrossRef]
37. Zhuang, W.; Gan, X.; Wen, Y.; Zhang, S.; Yi, S. Collaborative unsupervised visual representation learning from decentralized data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4912–4921.
38. Zhuang, W.; Wen, Y.; Zhang, S. Divergence-aware Federated Self-Supervised Learning. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
39. Li, S.; Mao, Y.; Li, J.; Xu, Y.; Li, J.; Chen, X.; Liu, S.; Zhao, X. FedUTN: Federated self-supervised learning with updating target network. *Appl. Intell.* **2023**, *53*, 10879–10892. [CrossRef]
40. Song, Z.; Zhang, Z.; Zhang, K.; Luo, W.; Fan, Z.; Ren, W.; Lu, J. Robust single image reflection removal against adversarial attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 24688–24698.
41. Wong, E.; Rice, L.; Kolter, J.Z. Fast is better than free: Revisiting adversarial training. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
42. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial Examples in the Physical World. In *Artificial Intelligence Safety and Security*; CRC: Boca Raton, FL, USA, 2018; p. 99.
43. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

44. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, University of Toronto, Toronto, ON, Canada, 2009.
45. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Reading Digits in Natural Images with Unsupervised Feature Learning. *Nips Workshop Deep. Learn. Unsupervised Feature Learn*. **2011**, *2011*, 4.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.