Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059472

ARTICLE

Federated Learning and Optimization for Few-Shot Image Classification

Yi Zuo, Zhenping Chen*, Jing Feng and Yunhao Fan

School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, 215009, China

* Corresponding Author: Zhenping Chen. Email: zhpchen10@163.com

Received: 09 October 2024; Accepted: 13 December 2024; Published: 06 March 2025

ABSTRACT: Image classification is crucial for various applications, including digital construction, smart manufacturing, and medical imaging. Focusing on the inadequate model generalization and data privacy concerns in few-shot image classification, in this paper, we propose a federated learning approach that incorporates privacy-preserving techniques. First, we utilize contrastive learning to train on local few-shot image data and apply various data augmentation methods to expand the sample size, thereby enhancing the model's generalization capabilities in few-shot contexts. Second, we introduce local differential privacy techniques and weight pruning methods to safeguard model parameters, perturbing the transmitted parameters to ensure user data privacy. Finally, numerical simulations are conducted to demonstrate the effectiveness of our proposed method. The results indicate that our approach significantly enhances model generalization and test accuracy compared to several popular federated learning algorithms while maintaining data privacy, highlighting its effectiveness and practicality in addressing the challenges of model generalization and data privacy in few-shot image scenarios.

KEYWORDS: Federated learning; contrastive learning; few-shot; differential privacy; data augmentation

1 Introduction

With the increasing of the number of categories covered by image classification systems, the cost of annotating large datasets rises sharply, especially for rare categories [1]. To train high-performance classification models, a significant quantity of labeled data is needed [2]. However, in many practical applications, obtaining such data is not only difficult but also expensive [3]. This is particularly true in scenarios involving sensitive data, such as medical images [4], financial data, or personal health records, where traditional centralized learning methods cannot meet privacy protection requirements. Because of this, few-shot learning is particularly important for image classification problems, where it is difficult to train a model with high generalization skills using a limited number of labeled samples [5]. The few-shot image classification problem has substantial practical significance in the context of federated learning [6]. To solve this problem, federated learning (FL) is essential [7]. Federated learning efficiently preserves data privacy and removes the need to upload sensitive data to a central server by allowing local model training on individual clients and aggregating model parameters on a central server [8]. However, the application of federated learning in few-shot image classification presents a prominent challenge: as each client has only a limited number of samples, the resulting model is prone to overfitting [9], leading to inadequate generalization capabilities. To address this issue, data augmentation becomes an important technique. By expanding the limited training data, data augmentation improves the model's adaptability to unknown data [10], thereby mitigating overfitting. However, within the federated learning framework, despite ensuring data privacy, the

model training process may still involve the transmission of gradients or model parameters, which introduces potential privacy leakage risks [11]. Even if the raw data is not directly shared, gradients or model parameters may still contain sufficient sensitive information, leading to the leakage of individual data points. To further enhance privacy protection, differential privacy techniques are introduced. By adding noise to the weights during model training, differential privacy ensures that model updates do not reveal any detailed information about individual data points, thereby maximizing data privacy protection.

In this paper, we address the requirement on both model generalization and data privacy and propose one differential federated few-shot contrastive learning (DFFCL) method. The main contributions include the following three aspects:

(1) To address the limited sample size and weak feature learning capabilities of models in few-shot datasets, we incorporate the few-shot contrastive learning into the federated learning framework. The resilience and generalization of the global model during aggregation are enhanced by using local devices to carry out the few-shot contrastive learning, which allows each device to efficiently update its model depending on its unique data properties.

(2) We employ the PatchMix augmentation techniques for few-shot image datasets to enhance the training set and boost model generalization. Moreover, methods like rotation, flipping, Gaussian noise, and color transformation, including PatchMix data augmentation, are utilized to improve the model capacity for extensibility.

(3) We combine the local differential privacy and model pruning techniques in the federated learning framework; in this way, we resolve the issue of traditional differential privacy techniques requiring a trusted central party to manage data and enforce privacy protections. By pruning local model weights and adding Laplacian noise, the model's adaptability to the data is limited, ensuring privacy protection while maintaining predictive capabilities, thus making federated learning more secure and reliable in few-shot scenarios for protecting client privacy.

2 Related Works

2.1 Federated Few-Shot Contrastive Learning

Federated learning has recently gained popularity as a decentralized method for improving privacy protection [12]. It protects privacy by allowing the aggregation of locally updated models from client devices to produce a shared model while guaranteeing that only local clients can access the original data [13]. Researchers have been concentrating on employing Few-Shot Learning (FSL) in federated learning contexts to address the problem of inadequate labeled training data on participating clients in federated learning. In [14], Li et al. proposed a differential private technique to safeguard parameter transmission between devices or learning stages, which is one study that links few-shot learning with federated learning. However, federated learning and few-shot learning are treated as two independent applications, and the goal of [14] is to protect data privacy during model sharing rather than to execute few-shot learning on federated devices. Fan et al. in [15] were the first to introduce Federated Few-Shot Learning (FedFSL), where they proposed adversarial learning techniques and refined client models to create a more distinct feature space, better suited for representing unseen data samples. Building on this, Wang et al. in [16] introduced a federated few-shot learning architecture, employing two independently trained models and a tailored training strategy to address the challenges of global data variability and local data insufficiency. Yao et al. in [17] addressed data heterogeneity with graph-aided federated learning and a few-shot node inhibition mechanism, optimizing client relationships. In contrast, combining contrastive learning and data augmentation enhances feature learning and model generalization in few-shot tasks. Hoang et al. [18] explored federated few-shot learning

for cough classification on edge devices, focusing on effective classification with limited samples while protecting data privacy and reducing computational load. The personalized federated few-shot learning method proposed by Zhao et al. [19] can effectively improve model performance in decentralized data environments, but it has limitations in enhancing model generalization.

Compared to traditional federated few-shot learning methods, contrastive learning-based approaches can better leverage the limited training data, achieving better performance in more complex visual tasks. It is becoming more and more clear that the contrastive learning based federated few-shot learning approach is preferable in terms of innovation. The model can better capture generalizable feature representations through contrastive learning by leveraging similarities and differences between different clients.

2.2 Data Augmentation

Since well-designed methods can greatly enhance the performance of models, data augmentation is essential to deep learning [20]. Common data augmentation methods include Cutout [21], CutMix [22], AugMix [20], and so on. Cutout, introduced in [21], involves removing portions of the input image during training, while CutMix, presented in [22], enhances model performance by blending patches of varying sizes and applying similar labels. AugMix, described in [20], combines multiple augmented input images with randomly selected weights. Additionally, MaskMix employs a multi-path weighted fusion strategy to further improve the quality of the transformed image pairs [23].

It is important to mention that PatchMix, as proposed in [24], tailors image frames for enhancement, effectively reducing inductive bias and enhancing federated few-shot learning. Therefore, this paper incorporates PatchMix, applying a randomly selected scale for each patch generation, which significantly increases data diversity and better facilitates the model ability to learn comprehensive, holistic features by exposing it to variations in patch size.

2.3 Differential Privacy

Differential privacy (DP) has emerged as the benchmark for publishing and analyzing sensitive data, attracting widespread attention [25]. But most differential privacy studies focus on centralized settings [26]. The application of local differential privacy (LDP) [27] has become increasingly widespread to eliminate the dependency on trusted central parties in differential privacy techniques. In LDP, each participant locally protects data privacy, thus safeguarding user privacy. Pittaluga et al. in [28] proposed a new image feature privatization method, which relies on local differential privacy with strong privacy guarantees. However, in federated learning, data still needs to be shared among different participants, which may lead to some privacy leakage. Ni et al. in [29] adopted a dual randomization data perturbation technique, allowing users to perturb their private values locally. However, this approach requires additional communication overhead to ensure data privacy. Sun proposed an adaptive weight clipping method in [30], applying varying degrees of noise perturbation to different layers' weights rather than the weights themselves. However, it should be noted that the method in [30] relies on specific network structures or tasks.

In this paper, we apply weight pruning on local clients and introduce local differential privacy noise prior to transmitting the pruned weights to the server for aggregation. This approach mitigates the risk of overfitting and enhances the model's generalization performance.

3 Theoretical Foundation

3.1 Contrastive Learning

Contrastive learning is an approach that directly learns the similarity and dissimilarity between different samples and applies this to downstream tasks. In contrast to supervised learning, where samples are directly labeled, contrastive learning is commonly used in scenarios where labeled data is scarce [31]. The principle of contrastive learning is to emphasize the relative distance between samples in the feature space, which enables better discrimination between different samples. By using a contrastive loss, the model is optimized to make the learned feature space reflect the inherent relationships between samples. Contrastive learning excels in limited-sample training because it learns richer feature representations by comparing pairs of samples. By maximizing similarity between similar samples and minimizing distance between dissimilar ones, this approach helps the model deeply understand the data. With limited data, contrastive learning effectively captures intrinsic data structure, reduces overfitting, and improves generalization to new samples, boosting classification accuracy [10].

In a few-shot learning setup, a limited set of labeled samples, called support samples, is used to guide model training, whereas the remaining samples serve as query samples. This setup divides the few-shot problem into two types: "shot," which refers to the number of samples per class, and "way," which refers to the number of classes. The main aim of contrastive learning is to make the distance between support samples and query samples in the feature space as optimized as possible. The typical contrastive loss can be expressed as:

$$L_{ij} = -\log \frac{\exp(sim(x_i, x_j)/\tau)}{\sum_{k=1}^{2N} \exp(sim(x_i, x_k)/\tau)}, \tag{1}$$

where the query sample, the positive support sample, and the negative support sample are denoted by x_i, x_j, and x_k, respectively; τ is a temperature parameter that regulates the distribution's sharpness.

Remark 1. *The model draws samples from the same class closer in the feature space while simultaneously separating samples from other classes due to the contrastive loss in (1). This aspect of contrastive loss allows the model to capture feature representations that generalize well across different classes, even with very few labeled samples [4,32].*

3.2 Data Augmentation Method: PatchMix

The main idea of PatchMix proposed in [24] is to incorporate patches from different parts of an image into the training process, enhancing the data variety and reducing the model reliance on simple correlations in the image. The model is used to learn the true semantic characteristics of the image. Fig. 1 illustrates the PatchMix augmentation process, where the key idea is to extract patches from different parts of the original image and mix them into the query image to generate augmented samples.

More specifically, PatchMix operates on each query sample x_i with label y_i, as well as another sample x_k with label y_k, where the patch p from x_k is swapped into the query sample x_i. For example, as shown in Fig. 1, a random region $[w_1 : h_1, w_2 : h_2]$ from x_k is extracted, where $[w_1 : h_1]$ represents the top-left corner and $[w_2 : h_2]$ represents the bottom-right corner. This patch p from x_k is then mixed into the corresponding region of the original image x_i. Each mixed patch p retains its original label, ensuring that the region remains associated with its original class. The specific operation is as follows:

$$x_i[w_1 : h_1, w_2 : h_2] = x_k[w_1 : h_1, w_2 : h_2]. \tag{2}$$

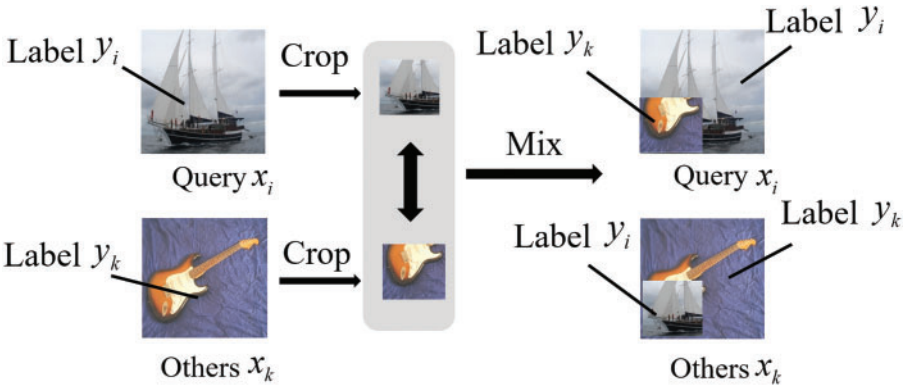

Figure 1: PatchMix data augmentation process

Remark 2. *In standard training, models often pick up biases, focusing on background features over key objects, which leads to overfitting and poor generalization. PatchMix tackles this by augmenting data with mixed patches that retain class labels, helping models to distinguish task-relevant (causal) features from incidental (non-causal) ones. This reduces dependency on superficial correlations, improving the model adaptability across diverse environments essential for real-world applications where datasets often lack full coverage due to cost, privacy, or physical limits. To mitigate mislabeling and irrelevant background interference, we could select patches from images with similar labels.*

3.3 Differential Privacy Technique

Dwork first proposed differential privacy in [25], providing a strong privacy protection paradigm independent of past information. The Laplace mechanism is a main tools used to achieve differential privacy [33]. The Laplace mechanism is typically used for numerical data. By preventing user data from being disclosed during the model aggregation process, the use of differential privacy in federated learning further improves data privacy. The ability to generalize is further improved since, even after adding noise, the data from a single user won't have a major impact on the global model.

Given a dataset D and a query function or algorithm K, definitions on ε-differential privacy, sensitivity, and Laplace mechanism are defined as follows.

Definition 1. A mechanism K is regarded as providing ε-differential privacy under the condition that, given any two neighboring datasets D and D' which vary in only one element, and for any subset of outputs S, the following inequality holds:

$$\Pr[K(D) \in S] \leq \exp(\varepsilon) \cdot \Pr[K(D') \in S]. \tag{3}$$

where ε is a non-negative privacy parameter that controls the privacy guarantee. A small ε provides stronger privacy guarantees.

Definition 2. For a query function K that maps the dataset D to $\mathbb{R}^d$, the sensitivity Δf of K, which represents the greatest alteration in the output resulting from any single modification in the input, is defined in the following way:

$$M(D) = K(D) + \mathrm{Lap}(\Delta f/\varepsilon). \tag{4}$$

Definition 3. The Laplace mechanism incorporates random noise that is drawn from the Laplace distribution. The scale parameter for this noise is set as $\Delta f/\varepsilon$. Here, Δf stands for the sensitivity of the query

function, while ε plays a role in governing the magnitude of the noise. The noise has a standard deviation of $\Delta f/\varepsilon$.

Remark 3. *The Laplace mechanism guarantees that the output of the query function won't change substantially, irrespective of whether an individual's data is contained within the dataset. In this way, it offers protection against potential privacy attacks that aim to deduce information regarding individual data entries.*

4 Main Method

In this section, we will illustrate the main results of this paper, including the overall system design, federated few-shot learning, PatchMix augmentation, and local differential privacy-based model protection.

4.1 Overall System Design

To carry out the task of few-shot image classification, we aim to address these two important problems: model training and data privacy protection. In this paper, we enhance the generalization ability of the model by employing data augmentation methods such as PatchMix for few-shot data and leveraging differential privacy techniques to protect sensitive data. Based on the federated learning framework, a novel few-shot image classification method that ensures privacy protection is proposed. The research framework of DFFCL is shown in Fig. 2.

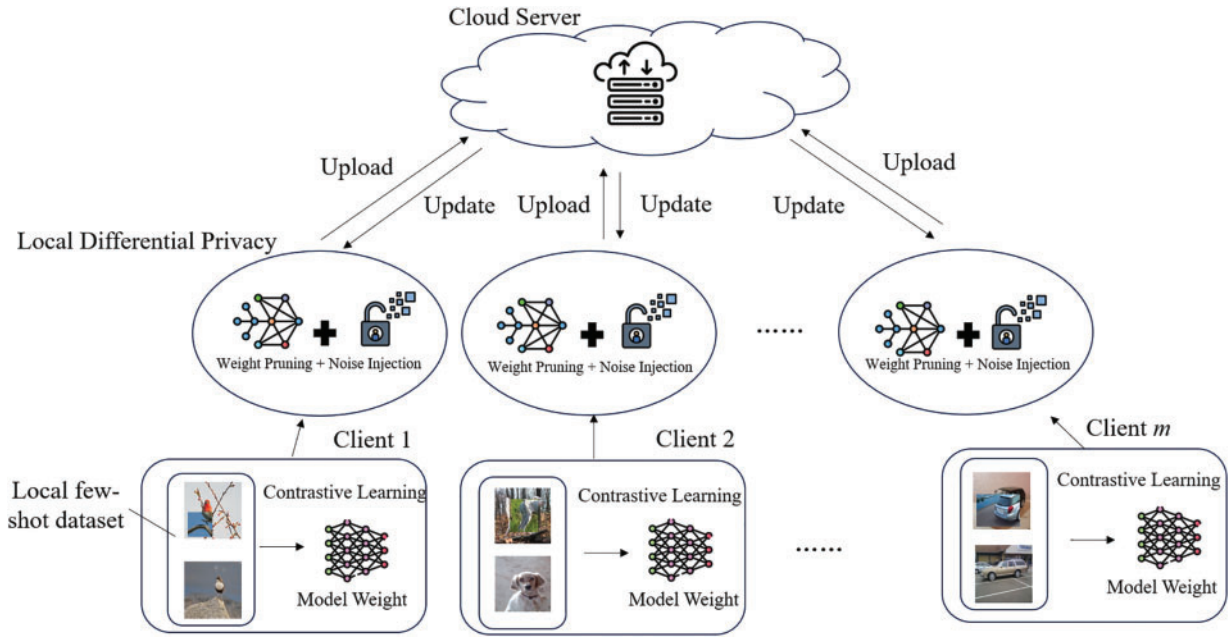

Figure 2: DFFCL framework diagram

In Fig. 2, the DFFCL framework illustrates a decentralized learning scenario involving m clients. Each client i $(i = 1, \ldots, m)$ maintains its local model parameters, ω_i, and optimizes its own local loss function $L(\omega_i)$. The clients utilize local few-shot datasets and apply techniques such as contrastive learning and data augmentation to enhance image classification capabilities. To ensure data privacy and security, local differential privacy mechanisms, including weight pruning and noise injection, are implemented before model updates are uploaded to the cloud server. In addition to enhancing the model performance in

scenarios with limited data samples, our method protects user privacy when training collaboratively. For every client i, the local objective is specified as follows:

$$\min L(\omega_i) = \frac{1}{Q_i} \sum_{i=1}^{Q_i} f(x_i, y_i; \omega_i), \tag{5}$$

where Q_i is the number of data owned by client i; x_i and y_i represent the data samples and their labels, respectively; $f(\cdot)$ is the local evaluation function used to assess the model's prediction loss on data samples. It is assumed that all clients have a similar loss function. The global objective is the weighted average of the local objectives, which is defined as:

$$\min L(\omega) = \sum_{i=1}^{m} \frac{Q_i}{Q} L(\omega_i), \tag{6}$$

where $L(\omega)$ is the global loss function, and ω is the global model parameters, and Q is the total number of data samples of all the m clients.

4.2 Federated Few-Shot Learning

4.2.1 Client-Side Few-Shot Contrastive Learning

We will describe in this section how to incorporate few-shot contrastive learning into the federated learning architecture.

Let's start by thinking about how to use the federated learning strategy to carry out the local training on each client i. The federated few-shot learning used in this paper is N-way k-shot. Also, we assume that: 1) each client's local dataset contains support images from N classes. 2) For each class, it has k support images and n_q query images. In FSL, for each training session or query sample x_i^q, there is a corresponding label y_i^q. Therefore, for each query sample, support samples sharing the same label are considered positive examples, while those assigned different labels are treated as negative examples. Positive and negative examples in the support set are distinguished by x_j^s and x_k^s, respectively. And the corresponding labels are presented by y_j^s and y_k^s, respectively. Let Φ represent the embedded network, and the deep learning-based embedded network is ResNet12 proposed in [34]. The client-side few-shot local contrastive learning procedure is illustrated in Fig. 3.

The N-way k-shot setting is used for both training and testing. z represents the output features of the embedding network Φ, whereas z_j^s and z_k^s stand for the positive and negative output features, respectively. Motivated by [35], in this paper, we assume that both z_j^s and z_k^s have been normalized, so no further normalization of the output features is required.

For learning from a partial view of the image, each query sample x_i^q is divided into $w \times h$ patches. For each small patch of query sample x_i^q, it is denoted as x_{iwh}^q, labeled as y_{iwh}^q, and output feature denoted as z_{iwh}^q. Similarly, the support image also undergoes random masking, and part of the support image is selected and input into the embedding network Φ. Random masks can be used to help models better learn invariant features. By using random masks, the model can be trained in different contexts, thereby improving its adaptability to different backgrounds and targets. Random masks and PatchMix will not conflict. In fact, they can be combined to improve the model. Certain parts of the input image can be changed during each training session, providing the model with more diversity and broader feature learning, which can help reduce the impact of label errors.

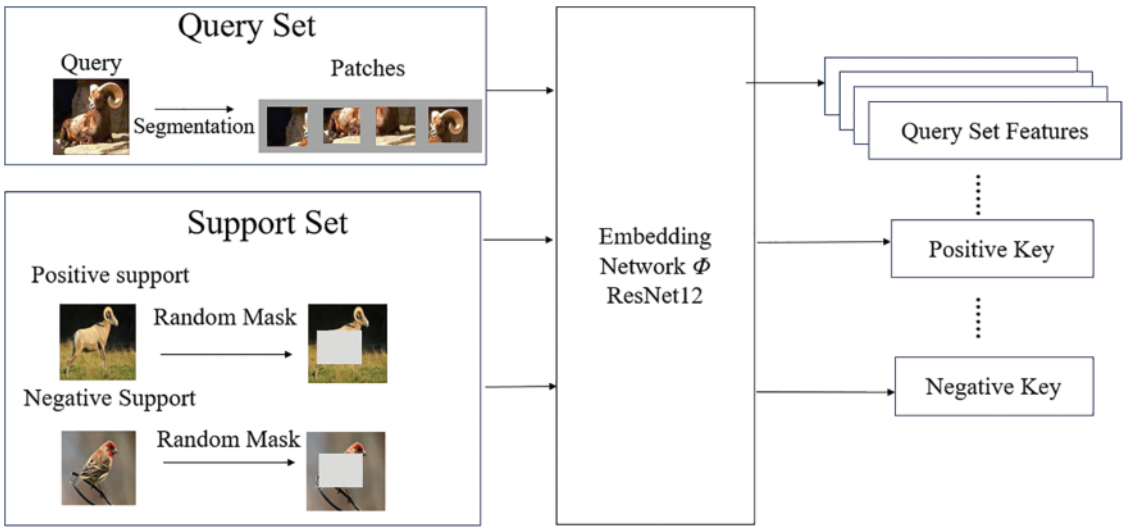

Figure 3: Client-side few-shot local contrastive learning

For both query and support samples, identical embedding networks Φ are applied. Motivated by [36], the loss function for a small query sample x_{iwh}^{q} is expressed:

$$L = -\log\left(\frac{\sum_{y_j^s=y_i^q} e^{z_{iwh}^{q}\cdot z_j^s}}{\sum_{y_j^s=y_i^q} e^{z_{iwh}^{q}\cdot z_j^s} + \sum_{y_k^s\neq y_i^q} e^{z_{iwh}^{q}\cdot z_k^s}}\right), \tag{7}$$

where $z_{iwh}^{q}\cdot z_j^s$ represents the inner product between the features of the query z_{iwh}^{q} and positive support image z_j^s. Similar for the inner product of $z_{iwh}^{q}\cdot z_k^s$. Keep in mind that the goal of (7) is to minimize the inner product for the negative support images and maximize it for the query and positive support images.

For a given query image, the dot product of the features between all support samples is calculated, while the network Φ is frozen during the test phase. For each query sample x_i^q, its feature z_i^q is first obtained. Then, the support sample z_j^s with the largest inner product with respect to z_i^q is identified as:

$$j^* = \arg\max_j z_i^q z_j^s. \tag{8}$$

Finally, the prediction result $y_i^q = y_j^s$ is obtained, yielding the classification outcome for the query image.

4.2.2 Server-Side Model Aggregation

In the federated learning context, the training data is spread across different devices. Also, each device is responsible for data with several specific types. Specifically, we consider the local training on each device i. The device i has local dataset D_i, updates its model parameters by reducing its local loss function. The objective of each device is to reduce the local loss. All the m devices communicate with the server through periodic aggregation iterations.

During every training iteration t, the local model parameter of each device ω_i^t is updated as follows:

$$\omega_i^{t+1} = \omega_i^t - \alpha\nabla L(\omega_i^t), \tag{9}$$

where ω_i^{t+1} denotes the newly refined local model weights after the t-th training iteration, α is the learning rate, and $\nabla L(\omega_i^t)$ is the gradient of the local loss function L concerning the model parameters ω_i. Once local training on all the m devices is completed, each device i sends its updated model parameter ω_i^{t+1} to the server for model aggregation. The server then combines the weights from all the m devices to obtain the updated global model.

To implement weighted aggregation based on local validation loss, a weighted average formula is introduced. The loss of each client i on its local validation set is denoted as L_i, and the weighting factor is represented as $\beta_i = \frac{1}{L_i}$, where a lower loss corresponds to a higher weight. The aggregated global model parameter ω^{t+1} can be expressed as:

$$\omega^{t+1} = \frac{\sum_{i=1}^{m} \beta_i \omega_i^{t+1}}{\sum_{i=1}^{m} \beta_i}. \tag{10}$$

Here, m represents the total number of devices that are taking part, and ω^{t+1} refers to the aggregated global model parameters. At the conclusion of each aggregation round, the server transmits the revised global model parameters ω^{t+1} to all m devices. Subsequently, these devices utilize these parameters during the following training cycle.

4.3 PatchMix Augmentation for Few-Shot Image Data

In this section, we explain how model generalization is achieved through the implementation of the PatchMix augmentation method.

It should be noted that the classification model may fare well in generalizing training data for few-shot image classification. We use the PatchMix image augmentation method to address the challenge of learning from small amounts of training data in new scenarios and improve the model's generalization across target classes. We do this by randomly blending query images with patches from other images to improve training and reduce the risk of overfitting.

For each query image $\{x_i^q,\ y_i^q\}$ with width W and height H, another query image is randomly selected as gallery image $\{x_k^s,\ y_k^s\}$, and information is collected from this gallery image for replacement. Next, a frame is randomly selected with width $\hat{w}$ and height $\hat{h}$. A random variable λ that follows the uniform distribution is chosen between $[0, 1]$, and the value of width $\hat{w}$ of the frame is obtained by multiplying the image width W by $\sqrt{1-\lambda}$, while the value of height $\hat{h}$ of the frame is obtained by multiplying the image height H by $\sqrt{1-\lambda}$. The aforementioned procedure is specified as:

$$\begin{aligned} \lambda &= \text{Unif}[0,\ 1], \\ \hat{w} &= W\sqrt{1-\lambda}, \\ \hat{h} &= H\sqrt{1-\lambda}, \end{aligned} \tag{11}$$

where $\text{Unif}[0, 1]$ indicates the uniformed distribution between 0 and 1. Moreover, the center $(c_w,\ c_h)$ of the patch p is random chosen as:

$$\begin{aligned} c_w &= \text{Unif}[w/2,\ W - w/2], \\ c_h &= \text{Unif}[h/2,\ H - h/2]. \end{aligned} \tag{12}$$

Let w_1, w_2, h_1, h_2 be the left, right, top, and bottom boundaries of the frame with center coordinates (c_w, c_h), width $\hat{w}$, and height $\hat{h}$. Then, a mask M and the mixed image $\hat{x}_i^q$ can be generated as:

$$M = \begin{cases} 1, & \text{if } w_1 \le i \le w_2,\ h_1 \le j \le h_2, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

and

$$\hat{x}_i^q = M \odot x_k^s + (1 - M) \odot x_i^q. \tag{14}$$

Unlike CutMix, which uses image-level labels, PatchMix retains pixel-level labels for supervision. The selected patch is remapped to the original query image x_i^q, with its placement in x_i^q determined based on specific design criteria as:

$$\begin{aligned} w_1' &= \frac{w_1}{W},\ w_2' = \frac{w_2}{W}, \\ h_1' &= \frac{h_1}{H},\ h_2' = \frac{h_2}{H}. \end{aligned} \tag{15}$$

(15) maps the position back to the original query image x_i^q. Moreover, the new label $Y_{i,j}$ is applied to the blended image as:

$$Y_{i,j} = \begin{cases} y_k^s, & \text{if } w_1' \le i \le w_2',\ h_1' \le j \le h_2', \\ y_i^q, & \text{o.w..} \end{cases} \tag{16}$$

Eventually, the blended image x_i^q is fed into the embedding network Φ and integrated with the original few-shot image data. In this way, the augmented data with label $Y_{i,j}$ is generated and to be used in the model training procedure.

4.4 Model Protection Based on Local Differential Privacy

By including privacy-preserving noise in the local model update procedure, differential privacy is accomplished. Each client updates its model in this work using local data of its own. Without interacting with the server or other clients, this operation is carried out locally. To safeguard privacy, noise is injected to the model weights after the model updates. In this context, local differential privacy is used to safeguard the privacy of model updates, with a primary focus on noise addition, sensitivity calculation, and weight clipping.

In order to prevent any one weight element from making an excessive contribution to the model update, weight clipping is first used. This can increase the model's robustnes by reducing the impact of individual data points. The following is the definition of the clipping process:

$$\omega_i^{\text{clipped}} = \omega_i \cdot \min\left(1, \frac{C}{\|\omega_i\|_2}\right), \tag{17}$$

where ω_i represents the model weights, C is a hyperparameter denoting the clipping threshold, and $\|\omega_i\|_2$ indicates the 2−norm of ω_i. The value of C is adjusted according to the specific requirements of the model and the privacy budget.

Next, the sensitivity Δf of the Laplace noise is determined, which is defined as the maximum difference in model weights computed from any two neighboring datasets. For the updated model weights, the change

in each element is limited by the sensitivity Δf. Therefore, weight clipping helps bind the sensitivity of the model updates.

Finally, to ensure privacy protection for each weight update, the Laplace noise is added to the clipped model weights as:

$$\omega_i^{\text{private}} = \omega_i^{\text{clipped}} + \text{Lap}(C/\varepsilon), \tag{18}$$

where $\text{Lap}(C/\varepsilon)$ represents Laplace noise with scale parameter ε is the privacy budget, and $\omega_i^{\text{private}}$ indicates the noised-added model weights. While local differential privacy and weight clipping may affect few-shot learning performance, optimizing noise and clipping can reduce this impact. Contrastive learning and data augmentation further balance data privacy and classification accuracy.

Remark 4. *In this paper, we choose the values of ε and C carefully because they directly impact the balance between privacy protection and accuracy. Smaller ε results in stronger privacy protection but will add more noise to the model, which may affect the classification accuracy. Conversely, a larger ε introduces less noise, providing weaker privacy protection but maintaining higher classification accuracy.*

To further clarify the implementation steps of the above methods, Algorithm 1 summarizes the entire process of few-shot contrastive learning in federated learning, including PatchMix data augmentation, local model training on the client side, and the differential privacy-based model protection strategy.

Algorithm 1: Differentially private federated few-shot contrastive learning (DFFCL) with weighted aggregation and PatchMix augmentation

Input: Local datasets $\{D_i\}_{i=1}^m$ for each client i, privacy parameters ε, C, learning rate α
Output: Global model parameters $\omega^{(T)}$
for each round $t = 1, 2, \ldots, T$ **do**
  for each client $i = 1, \ldots, m$ **in parallel do**
    Local Few-Shot Dataset Preparation:
    Perform PatchMix data augmentation on local dataset D_i
    Client-side Few-Shot Contrastive Learning:
    for each query sample x_j^q in D_i **do**
      Extract feature $z_j^q = \Phi(x_j^q)$
      for each support sample x_j^s in the support set of x_j^q **do**
        Extract feature $z_j^s = \Phi(x_j^s)$
        Compute similarity $s_j = z_j^q \cdot z_j^s$
      end for
      Identify $j^* = \arg\max_j s_j$ for highest similarity
      Predict label $\hat{y}^q = y_{j^*}^s$
    end for
    Model Training with Local Objective:
    for each local iteration **do**
      Update local model parameters $\omega_i^{(t+1)} = \omega_i^{(t)} - \alpha \nabla L(\omega_i^{(t)})$
      Clip weights: $\omega_i^{\text{clipped}} = \omega_i \cdot \min\left(1, \frac{C}{\|\omega_i\|_2}\right)$
      Add noise for differential privacy: $\omega_i^{\text{private}} = \omega_i^{\text{clipped}} + \text{Lap}(C/\varepsilon)$
    end for

(Continued)

Algorithm 1 (continued)

Compute validation loss L_i on local validation set
Compute weight $\beta_i = \frac{1}{L_i}$
Send $\omega_i^{\text{private}}$ and β_i to the server
end for
Server: Aggregate model parameters using weighted average:
$\omega^{(t+1)} = \frac{\sum_{i=1}^{m} \beta_i \cdot \omega_i^{\text{private}}}{\sum_{i=1}^{m} \beta_i}$
end for

5 Experiments Results and Analysis

The settings of our few-shot image categorization approach are demonstrated in this section.

5.1 Experimental Environment

A system running Ubuntu 20.04.6 with deep learning frameworks based on Pytorch 2.0.0 was used for the studies. Python 3.9.17 was the programming environment. The GPU was an NVIDIA GeForce RTX 4090, and the CPU was an Intel Xeon Platinum 8336C. 64 GB RAM was also included in the device.

5.2 Datasets

In the experiments, we utilized the MiniImageNet [37] and TieredImageNe [38] datasets. The MiniImageNet dataset comprises 100 classes, each containing 600 images. The TieredImageNet dataset follows a hierarchical structure, consisting of 608 classes organized into 34 super categories. With 351 classes set aside for training, 97 for validation, and 160 for testing, it has a total of 779,165 images. Both datasets use images with a resolution of 84×84 pixels. During training, data augmentation was performed by using random cropping and horizontal flipping.

5.3 Baseline Methods

We contrasted the suggested DFFCL approach with the following five baseline approaches in order to assess its performance. We reimplemented and improved all baseline procedures in our experiments to provide a fair comparison. This adjustment guarantees that each method operates under optimized settings, thereby enhancing the validity and fairness of the experimental results.

- **Local:** This baseline trains a model independently on each client without any communication between clients.
- **FL-MAML:** This approach combines MAML with federated learning [39], enabling local updates on each client and sending the new model back to the server.
- **FedFSL:** This approach [15] builds a consistent feature space by combining adversarial learning strategies [40] and MAML, aggregating based on FedAvg.
- **F2L:** This method [16] combines meta-learning to construct a consistent feature space, aggregating based on the FedAvg algorithm.
- **DFFCL:** The proposed method utilizes contrastive learning with few-shot learning to enhance performance. PatchMix augmentation is applied for more efficient training, and feature aggregation is performed using FedAvg strategies.

We created tests with the settings 5-way 1-shot and 5-way 5-shot to assess the model's performance on both tasks, since the 1-shot task is a more severe situation and the 5-shot task offers a more difficult scenario. The model architecture selected was ResNet12, and the TADAM optimization approach was used for optimization. The optimizer we selected was Stochastic Gradient Descent (SGD) [41]. In [42], a gradual decrease of the learning rate during training was introduced, at an initial learning rate of 0.1. For MiniImageNet, we trained the model for 120,000 steps with validation every 16,000 steps. For TieredImageNet, training was conducted for 240,000 steps with validation every 20,000 steps. Regular validation helps assess the model convergence and prevents overfitting and underfitting issues [39]. For the experiment, we used a test set of 2000 episodes, evaluating four episodes in each testing session, which contributes to stable test results [43].

5.4 Experimental Results and Analysis

5.4.1 Comparison of Classification Accuracy on the MiniImageNet Dataset

Using the MiniImageNet dataset, we first compare the classification accuracy of FL-MAML, FedFSL, F2L, and DFFCL. We perform trials under both IID and non-IID data partition conditions to demonstrate the stability of our system across different data distributions. Samples from every class are distributed equally across all clients in the IID scenario. We use a common strategy uesd in [44,45] for the non-IID situation, allocating samples to clients using a Dirichlet distribution with a concentration parameter of 1.0. The average classification accuracy over ten trials is used to evaluate performance.

Table 1 displays the outcomes of the experiment.

Table 1: Comparison on classification accuracy

Distribution	**IID**		**Non-IID**	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
FL-MAML	52.47%	65.59%	47.67%	63.34%
FedFSL	54.36%	70.29%	53.82%	69.23%
F2L	56.31%	**74.23%**	56.01%	**73.24%**
DFFCL	**58.14%**	73.41%	**57.15%**	72.16%

From Table 1, it is evident that DFFCL, proposed in this study, demonstrates superior performance in the 1-shot learning task, achieving an accuracy of 58.14% on IID data, outperforming all other methods. This result indicates DFFCL's capability in extracting robust feature representations under extreme few-shot conditions by effectively utilizing the limited information available. For the 5-shot learning scenario, DFFCL scores 73.41%, which, while slightly lower than F2L at 74.23%, remains competitive. This balanced performance across both 1-shot and 5-shot tasks highlights DFFCL's adaptability and generalization strength, making it a valuable approach in scenarios with constrained data and high variability.

5.4.2 Comparison of Data Augmentation Techniques

In this section, we first conducted comparative experiments focused on data augmentation techniques using the MiniImageNet and TieredImageNet datasets, specifically within the federated learning framework for 1-shot and 5-shot tasks. Initially, we compared the local model and the model without PatchMix augmentation, denoted as DFFCL-P, to perform an ablation study, aiming to examine the effectiveness of

PatchMix augmentation in federated few-shot image classification. Outcomes of experiment are presented in Table 2.

Table 2: Results of ablation experiments

Dataset	MiniImageNet		TieredImageNet	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Local	55.71%	68.97%	57.59%	77.10%
DFFCL-P	56.62%	71.36%	58.48%	77.76%
DFFCL	**58.14%**	**73.41%**	**60.67%**	**78.77%**

As shown in Table 2, the DFFCL method delivers notable improvements on both the MiniImageNet and TieredImageNet datasets. In particular, for the MiniImageNet 1-shot and 5-shot tasks, DFFCL outperforms other methods, demonstrating an approximately 2% higher accuracy. This performance can be credited to the implementation of PatchMix, which introduces randomness into the data augmentation process, increasing data diversity and helping the model capture more robust feature representations. These experiment results validate the effectiveness of the PatchMix-based data augmentation method in federated learning scenarios. Similarly, on the TieredImageNet dataset, DFFCL shows a similar trend, with significant accuracy gains over the baseline and the DFFCL-P models. This confirms the general applicability of the proposed DFFCL method, not only on MiniImageNet but also on TieredImageNet, where it shows improved performance across all tasks.

We next contrasted the PatchMix method with the Mixup and AugMix data augmentation techniques in the 1-shot and 5-shot settings. Table 3 displays the experimental outcomes. PatchMix outperforms Mixup and AugMix, increasing classification accuracy by roughly 2% for both tasks, as indicated in Table 3. These findings also show how PatchMix may successfully increase data diversity, which helps the model better adapt to unseen samples and strengthen its generalization ability on overfitted data.

Table 3: Comparison of classification accuracy among several data augmentation methods

Method	1-shot	5-shot
Mixup	57.18%	71.75%
AugMix	57.74%	72.55%
PatchMix	**58.14%**	**73.41%**

5.4.3 Effect of Client Number on Classification Accuracy

We will continue to examine how the quantity of clients affects classification accuracy in this section. As a case study, we employ the MiniImageNet dataset, with 2, 4, 8, 16, and 32 clients participating, respectively. With varying client numbers, we assessed accuracy for the 5-way 1-shot and 5-way 5-shot tasks. Fig. 4 provides an illustration of the results.

From Fig. 4, we can observe the following. 1) DFFCL shows a clear trend of accuracy improvement across all client number settings. With the number of clients increasing, DFFCL improves significantly, indicating that it adapts well to distributed learning environments and is able to maintain robust feature representation even when more clients participate. 2) In the 5-way 1-shot scenario, DFFCL maintains stable

performance even when the number of clients reaches 32, outperforming other methods. In the 5-way, 5-shot scenario, DFFCL consistently maintains stable accuracy, showing a slight performance decrease as the number of clients increases but still retaining a clear advantage over the other methods. 3) In contrast, FedFSIL, F2L, and FL-MAML experience noticeable drops in classification accuracy as the quantity of clients expands, particularly when the number of clients reaches 32, where the gap between DFFCL and the other methods becomes more apparent.

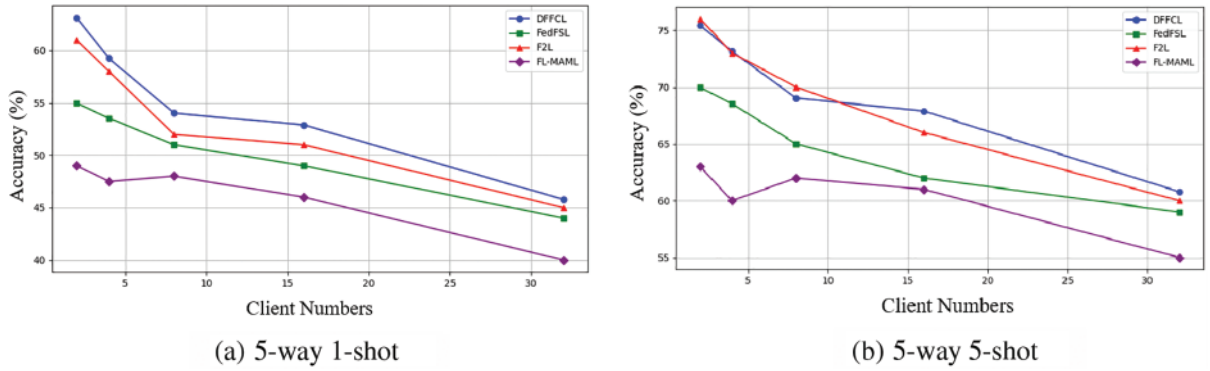

Figure 4: Effect of client number on classification accuracy

These experimental results demonstrate the potential of DFFCL in maintaining high accuracy and stability, even in challenging federated few-shot learning scenarios with increasing numbers of clients. Whether in situations with small amounts of data (1-shot) or relatively larger datasets (5-shot), DFFCL exhibits strong robustness, making it particularly suitable for applications requiring distributed learning in environments with many clients.

5.4.4 Impact of Differential Privacy Noise on Classification Accuracy

The effect of noise on model performance under various parameter settings is further investigated in this section. Finding the ideal balance between model accuracy and privacy protection is crucial in differential privacy. The amount of noise supplied grows as the privacy budget ε drops, which can both boost privacy protection and decrease model accuracy. Table 4 displays the model's performance under various ε [46] and C [47] conditions for the 5-way 1-shot scenario with four clients.

Table 4: Experimental results under different ε and C

Setting	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 1$
$C = 0.01$	50.36%	52.89%	54.16%
$C = 0.05$	52.36%	53.34%	54.54%
$C = 0.1$	53.78%	54.21%	55.54%

From Table 4, one can have that, when $\varepsilon = 0.05$, the classification accuracy is between 50.36% and 54.34%, depending on the value of C. As ε increases (up to 1), accuracy improves progressively. For example, with $\varepsilon = 0.1$, the accuracy ranges from 52.89% to 55.54%.

The accuracy improves as the privacy budget rises, according to the results. The model's performance is obviously impacted by the additional noise at smaller ε values (stronger privacy protection). This illustrates how differential privacy noise permits the model to retain a respectable accuracy range despite adding randomness. A key consideration in federated learning applications with varying privacy needs is the balance between the ε and C parameters, which can offer both robust model performance and efficient privacy protection.

To evaluate the performance of the DFFCL in balancing privacy and utility, we compared it with two privacy-preserving methods, LDP-FedSGD [48] and FedIOD [49], both of which also incorporate privacy techniques, and the comparison result is shown in Fig. 5. The accuracy of each method increases as the privacy budget ε increases. Overall, our DFFCL outperforms the two comparison methods across privacy levels, with its advantage becoming more pronounced as ε grows. This demonstrates DFFCL's effectiveness in achieving a superior balance between privacy and model accuracy, especially under large privacy budgets.

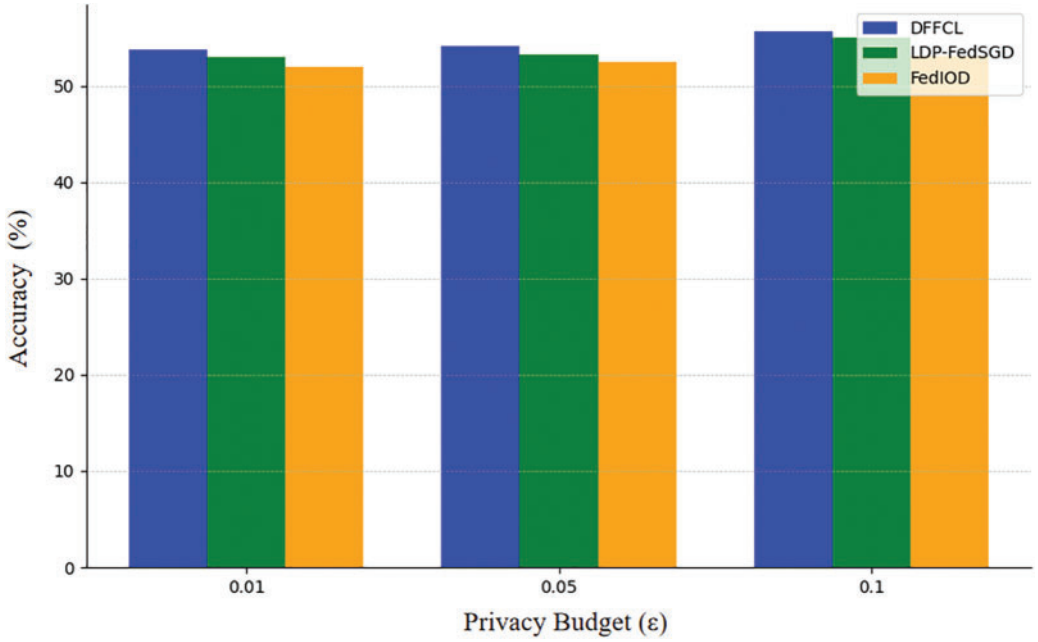

Figure 5: Comparison of DFFCL and others

6 Conclusion

In this paper, a federated few-shot learning framework, DFFCL, that incorporates differential privacy protection to address the unique characteristics and privacy requirements of few-shot image classification has been proposed. First, contrastive learning has been employed for local few-shot image training, and the PatchMix has been used for data augmentation to enhance the diversity of the training data. Then, local differential privacy techniques and weight clipping have been adopted to protect the privacy of client data during model updates. Also, the effectiveness of DFFCL was verified through experimental results. Experiment results demonstrated that the DFFCL has a strong capability to effectively handle limited data and improve generalization performance, especially when enhanced with PatchMix-based data augmentation, which significantly boosts the model's ability to generalize.

Future research will explore optimization strategies for federated few-shot learning, including reducing the communication costs, improving the computational efficiency of few-shot learning in distributed data

environments. Additionally, there will be further investigation into the specific impact of differential privacy techniques on model performance to achieve more refined privacy control mechanisms.

Acknowledgement: We would like to extend our sincere appreciation to the editor and reviewers for their valuable feedback and constructive comments, which significantly improved the quality of this paper. Their insightful suggestions have greatly contributed to the refinement of our work.

Funding Statement: This work was supported by Suzhou Science and Technology Plan (Basic Research) Project under Grant SJC2023002; Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX23_3322.

Author Contributions: Yi Zuo: Writing original draft, Software, Methodology, Formal analysis. Zhenping Chen: Writing review & editing, Supervision, Funding acquisition. Jing Feng: Supervision, Resources, Writing review. Yunhao Fan: Investigation, Writing review & editing, Data curation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in https://github.com/renmengye/few-shot-ssl-public (accessed on 07 October 2024) and https://www.kaggle.com/datasets/arjun2000ashok/tieredimagenet (accessed on 07 October 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Vairetti C, Assadi JL, Maldonado S. Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification. Expert Syst Appl. 2024;246:123149. doi:10.1016/j.eswa.2024.123149.
2. Xiao T, Xia T, Yang Y, Huang C, Wang X. Learning from massive noisy labeled data for image classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 2691–9. doi:10.1109/CVPR.2015.7298885.
3. Dhillon GS, Chaudhari P, Ravichandran A, Soatto S. A baseline for few-shot image classification. 2019. doi:10.48550/arXiv.1909.02729.
4. Zhang Y, Jiang H, Miura Y, Manning CD, Langlotz CP. Contrastive learning of medical visual representations from paired images and text. In: Proceedings of the 7th Machine Learning for Healthcare Conference; 2022. p. 2–25. doi:10.48550/arXiv.2010.00747.
5. Lin X, Li Z, Zhang P, Liu L, Zhou C, Wang B, et al. Structure-aware prototypical neural process for few-shot graph classification. IEEE Trans Neural Netw Learn Syst. 2022;35(4):4607–21. doi:10.1109/TNNLS.2022.3173318.
6. Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P. Rethinking few-shot image classification: a good embedding is all you need?. In: ECCV 2020: Computer Vision-ECCV 2020; 2020. p. 266–82. doi:10.1007/978-3-030-58568-6_16.
7. Li L, Fan Y, Tse M, Lin K. A review of applications in federated learning. Comput Ind Eng. 2020;149:106854. doi:10.1016/j.cie.2020.106854.
8. McMahan B, Moore E, Ramage D, Hampson S, Arcas B. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017; 2017. p. 1273–82. doi:10.48550/arXiv.1602.05629.
9. Song Y, Wang T, Cai P, Mondal S, Sahoo J. A comprehensive survey of few-shot learning: evolution, applications, challenges, and opportunities. ACM Comput Surv. 2023;55(13s):1–40. doi:10.1145/3582688.
10. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning; 2020. p. 1597–607. doi:10.48550/arXiv.2002.05709.

11. Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning. In: 2019 IEEE Symposium on Security and Privacy (SP); 2018. p. 1–15. doi:10.1109/SP.2019.00065.
12. Sun H, Chen X, Yuan KG. FL-EASGD: federated learning privacy security method based on homomorphic encryption. Comput Mater Contin. 2024;79(2):2361–73. doi:10.32604/cmc.2024.049159.
13. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. Proc Mach Learn Syst. 2020;2:429–50. doi:10.48550/arXiv.1812.06127.
14. Li J, Khodak M, Caldas S, Talwalkar A. Differentially private meta-learning. In: 8th International Conference on Learning Representations; 2020. p. 3678–6. doi:10.48550/arXiv.1909.05830.
15. Fan C, Huang J. Federated few-shot learning with adversarial learning. In: 2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt); 2021. p. 1–8. doi:10.23919/WiOpt52861.2021.9589192.
16. Wang S, Fu X, Ding K, Chen C, Chen H, Li J. Federated few-shot learning. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2023. p. 2374–85. doi:10.1145/3580305.3599347.
17. Yao Z, Song P, Zhao CH. Finding trustworthy neighbors: graph aided federated learning for few- shot industrial fault diagnosis with data heterogeneity. J Process Control. 2023;129:103038. doi:10.1016/j.jprocont.2023.103038.
18. Hoang ND, Tran-Anh D, Luong M, Tran C, Pham C. Federated few-shot learning for cough classification with edge devices. Appl Intell. 2023;53(23):28241–53. doi:10.1007/s10489-023-05006-4.
19. Zhao Y, Yu G, Wang J, Domeniconi C, Guo M, Zhang X, et al. Personalized federated few-shot learning. IEEE Trans Neural Netw Learn Syst. 2022;35(2):2534–44. doi:10.1109/TNNLS.2022.3190359.
20. Hendrycks D, Mu N, Cubuk ED, Zoph B, Gilmer J, Lakshminarayanan B. AugMix: a simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations; 2019. p. 5213–22. doi:10.48550/arXiv.1912.02781.
21. DeVries T, Taylor GW. Improved regularization of convolutional neural networks with cutout. 2017. doi:10.48550/arXiv.1708.04552.
22. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y. CutMix: regularization strategy to train strong classifiers with localizable features. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019. p. 6023–32. doi:10.48550/arXiv.1905.04899.
23. Xing Y, Wei J, Wang R, Huang R. MaskMix: mask mixing augmentation method for change detection. Appl Res Comput. 2023;40(12):3834–3840 3847. doi:10.19734/j.issn.1001-3695.2023.06.0228.
24. Xu C, Liu C, Sun X, Yang S, Wang Y, Wang C, et al. PatchMix augmentation to identify causal features in few-shot learning. IEEE Trans Pattern Anal Mach Intell. 2022;45(6):5523–32. doi:10.1109/TPAMI.2022.3223784.
25. Dwork C. Differential privacy. In: Proceedings of the International Colloquium on Automata, Languages, and Programming; 2006. p. 1–12.
26. Hassan MU, Rehmani MH, Chen J. Differential privacy techniques for cyber physical systems: a survey. IEEE Commun Surv Tutor. 2019;22(1):746–89. doi:10.1109/COMST.2019.2944748.
27. Wang N, Xiao X, Yang Y, Zhao J, Hui SC, Shin H, et al. Collecting and analyzing multidimensional data with local differential privacy. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE); 2019. p. 638–49. doi:10.1109/ICDE.2019.00063.
28. Pittaluga F, Zhuang B. LDP-Feat: image features with local differential privacy. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023. p. 17580–90. doi:10.48550/arXiv.2308.11223.
29. Ni Y, Li J, Chang W, Xiao J. A LDP-based privacy-preserving longitudinal and multidimensional range query scheme in IOT. IEEE Internet Things J. 2024;11(3):5210–21. doi:10.1109/JIOT.2023.3306003.
30. Sun L, Qian J, Chen X. LDP-FL: practical private aggregation in federated learning with local differential privacy. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence; 2021. p. 527–39. doi:10.48550/arXiv.2007.15789.
31. Tian Y, Sun C, Poole B, Krishnan D, Schmid C, Isola P. What makes for good views for contrastive learning?. Adv Neural Inf Process Syst. 2020;33:6827–39. doi:10.48550/arXiv.2005.10243.

32. Park S, Lee J, Lee P, Hwang S, Kim D, Byun H. Fair contrastive learning for facial attribute classification. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 10389–98. doi:10.48550/arXiv.2203.16209.
33. McSherry F, Talwar K. Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07); 2007. p. 94–103. doi:10.1109/FOCS.2007.66.
34. Oreshkin B, Rodriguez P, Lacoste A. TADAM: task dependent adaptive metric for improved few- shot learning. Adv Neural Inf Process Syst. 2018;31:251–62. doi:10.48550/arXiv.1805.10123.
35. Chen X, Fan H, Girshick R, He K. Improved baselines with momentum contrastive learning. 2020. doi:10.48550/arXiv.2003.04297.
36. Liu C, Fu Y, Xu C, Yang S, Li J, Wang C, et al. Learning a few-shot embedding model with contrastive learning. Proc AAAI Conf Artif Intell. 2021;35(10):8635–43. doi:10.1609/aaai.v35i10.17047.
37. Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D. Matching networks for one shot learning. Adv Neural Inf Process Syst. 2016;29:7527–39. doi:10.48550/arXiv.1606.04080.
38. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015;115(3):211–52. doi:10.1007/s11263-015-0816-y.
39. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning; 2017. p. 1126–35. doi:10.48550/arXiv.1703.03400.
40. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2020;63(11):139–44. doi:10.1145/3422622.
41. Bottou L. Large-scale machine learning with stochastic gradient descent. In: Proceeding of COMPSTAT; 2010. p. 177–86. doi:10.1007/978-3-7908-2604-3_16.
42. Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, et al. Large minibatch SGD: training imageNet in 1 hour. 2010. doi:10.48550/arXiv.1706.02677.
43. Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. Adv Neural Inf Process Syst. 2017;30:3474–85. doi:10.48550/arXiv.1703.05175.
44. Hsu TH, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification. 2019. doi:10.48550/arXiv.1909.06335.
45. Yu T, Bagdasaryan E, Shmatikov V. Salvaging federated learning by local adaptation. 2020. doi:10.48550/arXiv.2002.04758.
46. Abadi M, Chu A, Goodfellow I, McMaha HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016; 318, p. 308–18. doi:10.1145/2976749.
47. McMahan HB, Andrew G, Erlingsson U, Chien S, Mironov I, Papernot N, et al. A general approach to adding differential privacy to iterative training procedures. 2018. doi:10.48550/arXiv.1812.06210.
48. Zhao Y, Zhao J, Yang M, Wang T, Wang N, Lyu L, et al. Local differential privacy-based federated learning for internet of things. IEEE Internet Things J. 2020;8(11):8836–53. doi:10.1109/JIOT.2020.3037194.
49. Zhang X, Zhang X, Sun X, Zhang F, Zhang B, Li C, et al. DP-FedIOD: a differential privacy and federated learning based framework for aerial insulators orientation detection. IEEE Access. 2024;12(9):44826–40. doi:10.1109/ACCESS.2024.3380195.