IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 July 2024, accepted 10 August 2024, date of publication 16 August 2024, date of current version 16 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444483

RESEARCH ARTICLE

From Sample Poverty to Rich Feature Learning: A New Metric Learning Method for Few-Shot Classification

LEI ZHANG[1,2], YITING LIN[1,2], (Member, IEEE), XINYU YANG[1,2], TINGTING CHEN[1], XIYUAN CHENG[1,3], AND WENBIN CHENG[1,2]

[1]School of Computing, University of Electronic Science and Technology of China Zhongshan Institute, Zhongshan 528402, China
[2]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chendu 611731, China
[3]School of Automation, Guangdong University of Technology, Guangzhou 510006, China

Corresponding author: Wenbin Cheng (chengwenbin@zsc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62271130, in part by the Science and Technology Foundation of Guangdong Province under Grant 2023A1515010066, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010317.

ABSTRACT With the rapid development of deep learning in the field of computer vision, few-shot learning has emerged as an effective approach to tackle the challenge of data scarcity, garnering widespread attention from researchers. Despite significant progress in few-shot learning, current few-shot image classification methods have not fully exploited the feature extraction capabilities of the backbone network and the existing labeled data. To address this issue, we introduces a few-shot image classification method based on metric learning, which aims to more fully explore and utilize these resources. During training, we employed the PatchUp technique to perform block-level operations in the hidden layer feature space, obtaining more diverse feature representations, thereby expanding the range of data representation and aiding in the formation of smoother classification decision boundaries. Additionally, the introduction of a self-supervised auxiliary loss helps the network to learn deep semantic information stably, thereby enhancing the classification performance for new categories. Furthermore, the centralization and normalization of features extracted by the backbone network significantly enhance the model's performance in few-shot image classification tasks. This paper conducted extensive experiments on multiple public datasets (including miniImageNet, tieredImageNet, FC-100, and CUB-200-2011), demonstrating the effectiveness and superior performance of the proposed method. The experimental results show that the method significantly improves the model's classification accuracy and generalization ability in the 1-shot learning scenario, providing strong support for further research and application in the field of few-shot learning.

INDEX TERMS Few-Shot Learning, Image Classification, Metric Learning.

I. INTRODUCTION

In recent years, with the widespread application of large-scale datasets and powerful computing resources, deep learning has made significant advances in various fields [1], [2], [3]. In particular, image classification technology has developed rapidly in the field of computer vision, leading to the emergence of many mature models for visual task classification [4], [5], [6]. However, these models typically require a large number of labeled samples for training. In practical applications, obtaining a large-scale, high-quality labeled dataset often faces numerous constraints. For example, within the medical sector, there exists a dearth of data pertaining to infrequent occurrences, coupled with the high costs associated with data labeling. Similarly, in the realm of metal damage detection, the acquisition of a substantial

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

dataset of images depicting metal surface defects presents a considerable challenge. Furthermore, in the domain of semiconductor chip defect detection, the diversity of chip models and the paucity of defect-related data significantly impede the process of image collection. Therefore, effectively learning and completing specific tasks under conditions of limited sample availability has become an urgent problem to address.

When data samples are scarce, models struggle to train sufficiently and are prone to overfitting. Furthermore, traditional deep learning models typically exhibit good recognition capabilities only for categories present in the training dataset and find it difficult to generalize to unseen categories. In contrast, humans can learn quickly from a small amount of data. For example, a child who has never seen a whale in real life can immediately recognize it as a ''whale'' when they see one at an ocean park, having seen whales on television or in images. This rapid learning and generalization ability significantly surpasses that of machine learning. To effectively address the overfitting problem caused by data scarcity and enable models to learn quickly, researchers aim to mimic human learning methods. Inspired by humans' ability to rapidly acquire information about object categories and perform classification and recognition with minimal labeled data, the concept of few-shot learning has emerged [7], [8], [9].

The concept of few-shot learning was first proposed by Fei-Fei and others in 2003 [9]. Few-shot learning for image classification can effectively utilize a small number of labeled samples to learn a new category, thereby accomplishing image classification tasks for that category. In few-shot classification tasks, it is necessary to learn and understand new categories from very few instances, but with only a limited number of annotated samples per category. Traditional machine learning models struggle to succeed in such scenarios, making it essential to leverage prior knowledge acquired from previous learning to assist the model in its learning process.

In the field of few-shot image classification, research methods primarily fall into three categories: data augmentation [10], [11], [12], meta-learning [13], [14], [15], and metric learning [16], [17], [18]. Data augmentation generates new samples through networks, which can effectively address few-shot tasks when there is a sufficient amount of data [19], [20], [21]. Meta-learning integrates the concept of meta-learning into few-shot classification tasks, enabling rapid learning across different tasks through a cross-task approach, effectively incorporating prior knowledge into various tasks [22], [23], [24]. Metric learning involves iteratively training models to learn a robust feature embedding network, utilizing this network to understand sample features and completing few-shot classification tasks by measuring similarities between samples [25].

While these methods have shown promising results, they also introduce increased model complexity and reliance on more training data [26], [27]. Although directly introducing large volumes of new training data to acquire more knowledge is an appealing solution, it contradicts the aim of reducing dependency on large datasets in few-shot learning. Another viable approach is to enhance the efficiency of models in accumulating knowledge from existing data. Research indicates that current models may not fully leverage the feature extraction capabilities of backbone networks and existing training data. Currently, some scholars have proposed using regularization techniques to enhance model generalization [28]. For instance, [29] proposed a method combining Manifold Mixup [30] with self-supervised auxiliary loss to achieve a backbone network with strong generalization capabilities. However, due to the linear interpolation strategy of Manifold Mixup, it may not generate samples across a wider range in feature space, potentially leading to a lack of diversity in learned feature representations.

Addressing these issues in the context of few-shot image classification tasks, this paper proposes a metric learning-based approach. During training, we strengthen the feature extraction capabilities of the backbone network using PatchUp [31] and self-supervised auxiliary loss. Unlike Manifold Mixup's [30] linear interpolation in feature space, PatchUp operates at the block level in the hidden layer's feature space (mixing or swapping), yielding diversified feature representations. Expanding the representation range of data aids in forming smoother decision boundaries for classification, enhancing the model's adaptability to data variations. Furthermore, to tackle the challenge of generalizing to new categories, particularly in potential data distribution shift scenarios, we introduce a self-supervised learning auxiliary loss. Integrating this auxiliary loss helps the network stabilize in learning feature representations rich in deep semantic information. Finally, during the classification task, we perform simple feature distribution optimization [32] on the features extracted by the feature extraction network. Remarkably, our method achieves significant advances in inductive few-shot image classification.

The main contributions of this paper are as follows:

1. We have discovered that PatchUp [31], a feature space block-level regularization technique, significantly enhances the generalization capability of models compared to other regularization methods. Additionally, we have analyzed and validated the effectiveness of various data augmentation techniques, such as CutMix [33], MixUp [34], PatchUp, and Manifold Mixup [30], in the context of few-shot image classification.

2. Introducing self-supervised loss during training has proven effective in achieving robust semantic feature learning, leading to a notable improvement in the performance of few-shot classification tasks. Additionally, we propose a strategy that combines data augmentation with self-supervised auxiliary loss to further enhance the model's ability to generalize to new categories.

3. By centralizing and normalizing the features extracted from the backbone network, we have significantly improved the model's performance in few-shot image classification tasks, especially in the highly challenging 1-shot learning scenarios.

4. We propose a comprehensive method that integrates PatchUp, self-supervised auxiliary loss, and feature preprocessing for in-depth research on few-shot image classification tasks. Extensive experiments across multiple datasets have validated the effectiveness of our proposed method and demonstrated its superior performance in few-shot classification tasks.

The structure of the subsequent sections in this paper is organized as follows: Section II provides a brief overview of common data augmentation techniques and self-supervised learning approaches. Section III details the methodology proposed in this paper. Section IV presents the classification results and further substantiates the effectiveness of each component through ablation studies. The paper concludes with a summary in the final section.

II. RELATED THEORY

A. DATA AUGMENTATION

In the field of deep learning, data augmentation is a key technology that simulates various possible testing scenarios by diversifying training data, thereby improving the model's generalization ability and reducing overfitting, especially in situations with limited data volume. Common data augmentation methods include Dropout [35], DropBlock [36], Mixup [34], Manifold Mixup [30], Cutout [37], CutMix [33], Puzzle Mix [38], and PatchUp [31].

Dropout technology enhances the robustness of the model and prevents overfitting by randomly discarding neurons in the network during training, introducing noise. SpatialDropout and DropBlock are extensions of Dropout, which further regularize and reduce dependencies between features by randomly discarding the entire feature map or continuous blocks within the feature map. Mixup technology linearly interpolates samples in the input space to generate new training data, while Manifold Mixup extends this concept to the feature space and interpolates on the hidden layer representation to promote smooth behavior between training samples and enhance generalization ability. The Cutout method forces the model to learn different local features by randomly masking regions in the input image, thereby improving the generalization ability of local features. CutMix further combines the Cutout idea by cutting and exchanging image blocks to generate new samples, enhancing the model's ability to recognize local and global structures. Puzzle Mix utilizes significance analysis to optimize the sample mixing strategy, selectively interpolating and mixing in key areas of the image, avoiding inappropriate mixing in key recognition areas, thereby improving the quality of mixed samples. The PatchUp method operates on continuous blocks in the feature space, enhancing the diversity and robustness of feature representations through operations such as swapping or interpolation. By applying regularization on feature maps in hidden layers, the model's generalization ability for different feature dimensions is further improved.

B. SELF-SUPERVISED LEARNING

Self-Supervised Learning (SSL) is a form of unsupervised learning where models are trained to predict certain attributes or features of the data they generate themselves, rather than relying on explicit labels [39]. This approach leverages the intrinsic structural properties of the data as supervisory signals, enabling the model to learn useful representations without the need for external annotations. The core idea of SSL is to use the inherent structure and relationships within the data as supervisory information, guiding the model to learn effective representations of the data through the design of prediction tasks or constructive tasks.Self-Supervised Learning encompasses a variety of tasks, including but not limited to image rotation prediction [40], image colorization [41], and time series forecasting [42]. By utilizing unlabeled data to learn robust feature representations, SSL enhances the generalization capabilities of models while reducing reliance on costly annotated data. Consequently, it improves the performance of machine learning models at a lower cost.

III. METHODOLOGY

A. PROBLEM DEFINITION

Within the framework of few-shot learning, we distinguish between two key datasets: the base class dataset and the novel class dataset. The base class dataset serves as auxiliary data, primarily used for training, to enhance the model's ability to recognize novel classes through knowledge transfer. The novel class dataset is the target dataset for classification tasks, containing classes that the model needs to recognize but has not seen during training. These two datasets are mutually exclusive in terms of class labels, representing entirely different entity sets. Here, the base dataset $\mathbb{D}_{\text{base}}$ is represented as:

$$\mathbb{D}_{\text{base}} = \{(X_i, Y_i) \mid X_i \in \mathcal{X}_{\text{base}}, Y_i \in \mathcal{Y}_{\text{base}}\}_{i=1}^{N_{\text{base}}} \quad (1)$$

where each instance X_i is labeled with Y_i. Similarly, the novel class dataset $\mathbb{D}_{\text{novel}}$ is represented as:

$$\mathbb{D}_{\text{novel}} = \left\{(\widetilde{X}_j, \widetilde{Y}_j) \mid \widetilde{X}_j \in \mathcal{X}_{\text{novel}}, \widetilde{Y}_j \in \mathcal{Y}_{\text{novel}}\right\}_{j=1}^{N_{\text{novel}}} \quad (2)$$

and there is no overlap in the label space, i.e., $\mathcal{Y}_{\text{base}} \cap \mathcal{Y}_{\text{novel}} = \varnothing$.

In the context of few-shot image classification tasks, we typically divide the novel class dataset into two parts: the support set and the query set. The support set consists of a small number of labeled samples provided during the testing phase, used for rapid adjustment and adaptation of the model. The query set comprises unlabeled samples during the testing phase, requiring the model to classify based on the learning outcomes from the support set. Few-shot classification tasks

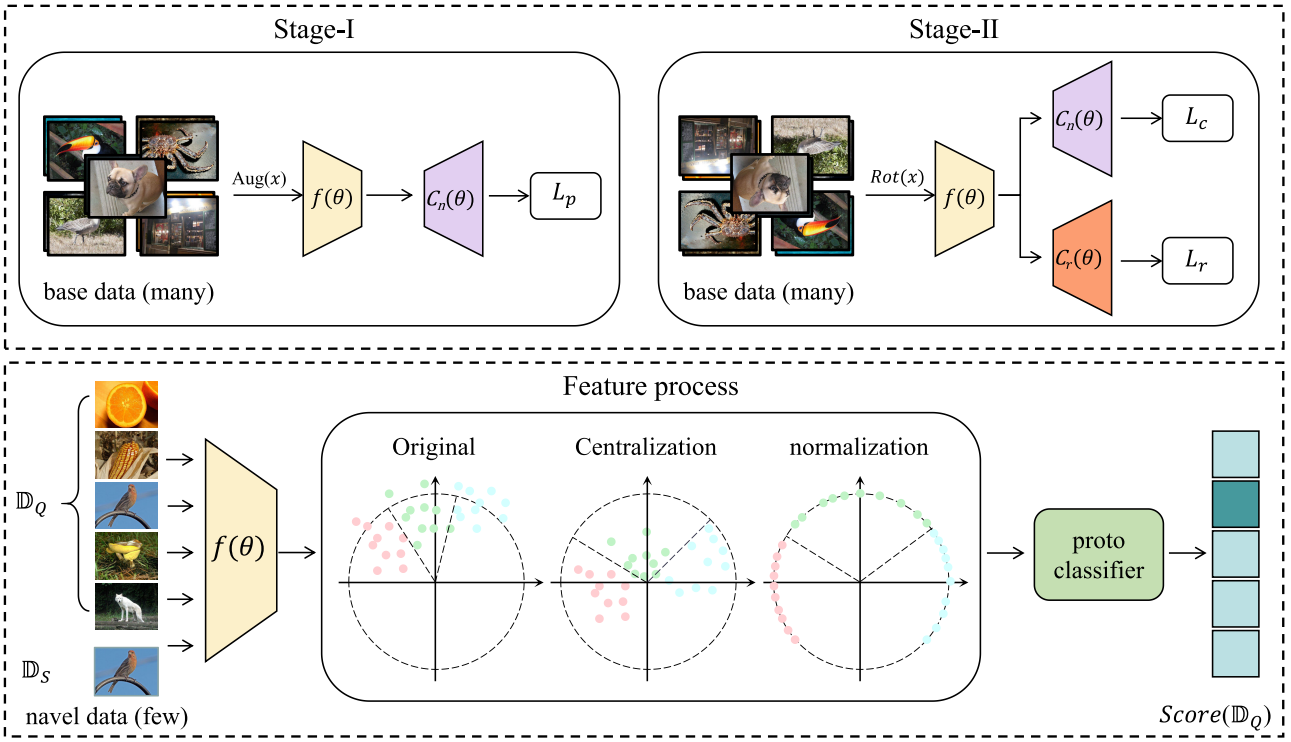


FIGURE 1. Overall framework of the model.

can be further divided based on the number of samples in the support set. When the support set contains N classes, each with K labeled samples, the task is referred to as an N-way K-shot classification. If each class in the support set has only one labeled sample, i.e., $K = 1$, this special case is referred to as one-shot classification. In practical testing, N-way K-shot classification tasks are performed by randomly selecting K samples from N classes in the support set. The performance of the classifier is evaluated based on its prediction accuracy on the query set, i.e., the proportion of correctly predicted classes compared to the actual classes. This accuracy is a key metric for assessing the performance of few-shot classifiers, reflecting the model's ability to recognize and generalize to new classes.

B. OVERALL FRAMEWORK

The paper introduces a novel approach to few-shot image classification that is predicated on a metric learning framework and is structured into two principal components: network training and few-shot image classification, as illustrated in Figure 1. In the network training phase, the PatchUp technique is employed, a block-level regularization method in the feature space that effectively enhances the model's generalization capabilities. Additionally, a self-supervised auxiliary loss is incorporated to encourage the model to learn robust semantic features. During the few-shot image classification phase, the network, once trained, is utilized for feature extraction. The extracted features are then centralized and normalized to improve their discriminative power. Ultimately, a prototype network based on metric learning is employed as the fundamental few-shot learning (FSL) classifier to fulfill the task of image classification.

C. BACKBONE TRAINING

Inspired by S2M2 [29], during the Backbone Training phase, in addition to the main classifier $C_n(\cdot)$ (where n denotes the total number of base categories within the dataset $\mathbb{D}_{\text{base}}$; for instance, if using the 64 base categories of the miniImageNet dataset $\mathbb{D}_{\text{base}}$, then $n = 64$), we introduce an additional four-way classifier $C_r(\cdot)$. The goal of this four-way classifier is to recognize which of the four possible rotations the input sample has undergone: $\{0°, 90°, 180°, 270°\}$. In this self-supervised task, the input images are rotated by different angles, and the auxiliary objective of the model is to predict the rotation angle of these images. During the model training process, each epoch consists of two stages of forward-backward propagation.

In the **Stage-I**, the input data is only used for the main classifier, and the PatchUp algorithm is applied for data augmentation. PatchUp includes both soft and hard modes. In PatchUp-hard, the hard mode directly replaces a selected region of one image with the corresponding region of another image in the hidden representations, resulting in augmented samples with clear boundaries. The soft mode, on the other hand, blends the selected regions in the hidden representations using linear interpolation, thereby enhancing

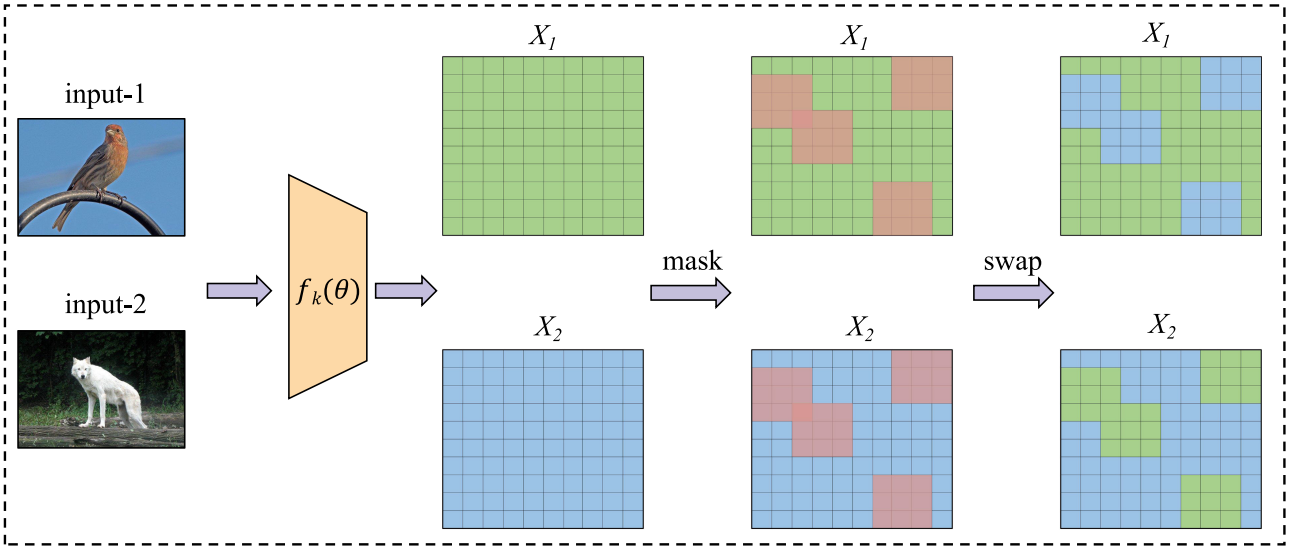


FIGURE 2. PatchUp-hard process for two hidden representations associated with two samples randomly selected in the mini-batch (a, b), $X_1 = g_k(a)$, $X_2 = g_k(b)$.

data diversity and model generalization. In this paper, we use the PatchUp-Hard mode. The specific processing flow of PatchUp-hard is shown in Figure 2.

When training a neural network $f(x) = f_k(g_k(x))$, where g_k represents the part of the network that maps the input data to the hidden representation of the k th layer, and f_k represents the part that maps the hidden representation to the output $f(x)$. The process involves the following steps:

1. Randomly select the k -th layer from the set of eligible layers S, which may include the input layer $g_0(x)$.

2. Process two random mini-batches of data (x_i, y_i) and (x_j, y_j) until the k -th layer, obtaining the representations $(g_k(x_i), y_i)$ and $(g_k(x_j), y_j)$ at the k -th layer of the network.

3. Create a binary mask M on the feature maps of the k -th layer, where 1 indicates unchanged features and 0 indicates features to be swapped or mixed.

4. For PatchUp-Hard, swap the corresponding feature blocks in $g_k(x_i)$ and $g_k(x_j)$ under the effect of mask M, generating a new hidden representation $\phi_{\text{hard}}(g_k(x_i), g_k(x_j))$.

5. Starting from layer k, continue the forward propagation of the network using the new hidden representation until the output $\tilde{z}$.

6. Calculate the loss L_{patchup} using the output and update all the parameters of the neural network. For Hard PatchUp, the loss function can be represented as:

$$L_{patchup} = p_u \ell(C_n(\tilde{z}), y_i) + (1 - p_u)\ell(C_n(\tilde{z}), y_j) \quad (3)$$

where p_u is the proportion of unchanged feature map parts, and ℓ is the cross entropy loss function

In the **Stage-II**, the original input is applied with angle rotation $A_r = \{0°, 90°, 180°, 270°\}$ and provided to both the main classifier and the auxiliary classifier to obtain loss $L_{total} = L_{rot} + L_{class}$. Then, update all parameters of the neural network again based on this loss.

$$L_{rot} = \sum_{x \in \mathbb{D}_{\text{base}}} \sum_{r \in A_r} \ell(C_r(f_\theta(x^r), r)) \quad (4)$$

$$L_{class} = \sum_{x \in \mathbb{D}_{\text{base}}} \sum_{n \in class(\mathbb{D}_{\text{base}})} \ell(C_n(f_\theta(x^r), n)) \quad (5)$$

where x^r signifies the input image that has been rotated by an angle associated with r from the set A_r.

After the training is completed, the backbone network will be frozen, and then the prototype classifier will be used to perform image classification tasks. The entire training process is shown in pseudocode 1.

D. FEATURE PREPROCESSING

To enhance the robustness and generalization ability of feature representations, we sequentially applied feature centering and feature normalization preprocessing steps to the features extracted by the feature extraction network. The detailed descriptions of these steps are as follows:

1) FEATURE CENTERING

Feature centering aims to standardize the features by removing the mean, thereby reducing bias among features. Suppose the training feature matrix is $\mathbf{X}_{\text{train}} \in \mathbb{R}^{N \times D}$, where N is the number of samples and D is the feature dimension. We first compute the mean of all feature vectors:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \mathbf{X}_{\text{train}}^{(i)} \quad (6)$$

Next, for each feature $\mathbf{X}$, we perform feature centering by subtracting the mean vector μ:

$$\mathbf{X}' = \mathbf{X} - \mu \quad (7)$$

Algorithm 1 Backbone Training

Require: Dataset $\mathbb{D}_{\text{base}}$, main classifier $C_n(\cdot)$, auxiliary classifier $C_r(\cdot)$, learning rate η
Ensure: Trained backbone network f_θ

```
Initialize f_θ
for epoch e = 1 to E do
    for each mini-batch (x, y) from D_base do
        Stage-I: PatchUp Augmentation
        Select k-th layer randomly from S
        Process (x_i, y_i) and (x_j, y_j) to obtain g_k(x_i) and g_k(x_j)
        Create binary mask M
        φ_hard(g_k(x_i), g_k(x_j)) ← M·g_k(x_i)+(1−M)·g_k(x_j)
        Obtain z̃ by forward propagation from layer k using φ_hard(g_k(x_i), g_k(x_j))
        L_patchup = p_u ℓ(C_n(z̃), y_i) + (1 − p_u) ℓ(C_n(z̃), y_j)
        θ ← θ − η∇_θ L_patchup
        Stage-II: Self-Supervised Rotation
        x_r ← Rotate(x, r) where r ∈ {0°, 90°, 180°, 270°}
        L_rot = Σ_{x∈D_base} Σ_{r∈A_r} ℓ(C_r(f_θ(x^r)), r)
        L_class = Σ_{x∈D_base} Σ_{n∈class(D_base)} ℓ(C_n(f_θ(x^r)), n)
        L_total = 0.5 * L_rot + 0.5 * L_class
        θ ← θ − η∇_θ L_total
    end for
end for
```

This processing ensures that the mean of all feature vectors is zero, thereby removing the bias in the data and improving the stability and consistency of the features.

Subtracting the mean μ of the training dataset is intended to ensure the consistency of feature distributions, so that training and testing features are compared and processed under the same mean baseline. This step effectively reduces bias caused by different data distributions, thereby improving the model's generalization ability.

2) FEATURE NORMALIZATION

After feature centering, apply feature normalization to further standardize the magnitude of the features. Specifically, we use the L2 norm for normalization. For each feature vector $\mathbf{X}^{(i)}$, we compute its L2 norm.

Then, we divide each feature vector by its L2 norm to ensure that each feature vector has a length of 1. In this way, all feature vectors are normalized to the unit sphere, thereby enhancing the robustness of the features.

$$\mathbf{X}^{(i)}_{\text{normalized}} = \frac{\mathbf{X}^{(i)}}{\|\mathbf{X}^{(i)}\|_2} \tag{8}$$

Combining the above two steps, we first apply centering to the training features, and then normalize the centered features. For the test features, we first apply the mean of the training features for centering, followed by normalization.

TABLE 1. The category split of three datasets.

Dataset	Split	Classes	Images
miniImageNet[43]	Train	64	38,400
	val	16	9,600
	Test	20	12,000
	Total	100	60,000
tieredImageNet[24]	Train	351	448,695
	val	97	124,261
	Test	160	206,209
	Total	602	779,165
CUB-200-2011[45]	Train	130	7,648
	val	20	1,182
	Test	50	2,958
	Total	200	11,788
FC-100[44]	Train	60	36,000
	val	20	12,000
	Test	20	12,000
	Total	100	60,000

This preprocessing process can be formally expressed as:

$$\mathbf{X}_{\text{preprocessed}} = \text{Normalization}(\mathbf{X} - \mu) \tag{9}$$

By applying centering, eliminate the mean bias of the features, making the feature distribution more uniform, thereby reducing the bias introduced by different feature means during training. Subsequently, by applying normalization, we standardize the magnitude of the features, ensuring that all feature vectors have an L2 norm of 1, thereby reducing the magnitude differences among features. These two steps not only improve the stability and consistency of the features but also enhance the robustness and generalization ability of the feature representations.

E. PROTOTYPE CLASSIFIER

We adopt a metric-based prototypical network as the fundamental few-shot Learning (FSL) classifier. This network consists of a feature extractor and a simple parameter-free classifier, where the training process only requires learning the parameters of the feature extractor. The prototypical network learns a nonlinear mapping from the input to the embedding space through a Convolutional Neural Network (CNN), projecting each sample into the same embedding space. For each class of samples in the support set, the average value in the embedding space is extracted as its class prototype, and the Euclidean distance is used as the metric. Through training, the distance between query set samples and the prototypes of the same class is minimized, while the distance to prototypes of other classes is maximized. For an N-way K-shot few-shot task, the specific steps of the prototypical classifier are as follows:

TABLE 2. Few-shot classification accuracy and 95% confidence interval on miniImageNet and FC-100 with the ResNet12 backbone.

Method	Backbone	miniImageNet		FC-100	
		1-shot	5-shot	1-shot	5-shot
MatchingNet[17]	ResNet12	63.08 ± 0.80	75.99 ± 0.60	-	-
ProtoNet[25]	ResNet12	60.37 ± 0.83	78.02 ± 0.57	37.53 ± 0.40	38.39 ± 0.40
TADAM[44]	ResNet12	58.50 ± 0.30	76.70 ± 0.30	40.10 ± 0.40	56.10 ± 0.40
Meta-Baseline[22]	ResNet12	63.17 ± 0.23	79.26 ± 0.17	-	-
DeepEMD[46]	ResNet12	65.91 ± 0.82	82.41 ± 0.56	46.60 ± 0.26	63.22 ± 0.71
Neg-cosine[47]	ResNet12	62.33 ± 0.82	80.94 ± 0.59	-	-
P-Transfer[48]	ResNet12	64.21 ± 0.77	80.38 ± 0.59	-	-
MetaOptNet-SVM[49]	ResNet12	64.09 ± 0.62	80.00 ± 0.45	**47.2 ± 0.60**	62.5 ± 0.60
AFHN[50]	ResNet18†	62.38 ± 0.72	78.16 ± 0.56	-	-
S2M2R[29]	ResNet18†	64.06 ± 0.18	80.58 ± 0.12	-	-
Our	ResNet12	**68.81 ± 0.20**	**84.61 ± 0.13**	46.63 ± 0.19	**63.57 ± 0.19**

Note: **Boldface** indicates the best results in each column, Underlined indicates the second-best results in each column, and '-' indicates that there are no experimental results available in the relevant literature. In the backbone network, † indicates a deeper network structure than ResNet12.

TABLE 3. Few-shot classification accuracy and 95% confidence interval on tieredImageNet and CUB-200-2011 with the ResNet12 backbone.

Method	Backbone	tieredImageNet		CUB-200-2011	
		1-shot	5-shot	1-shot	5-shot
MatchingNet[17]	ResNet12	68.50 ± 0.92	80.60 ± 0.71	71.87 ± 0.85	85.08 ± 0.57
ProtoNet[25]	ResNet12	65.65 ± 0.92	83.40 ± 0.65	66.09 ± 0.92	82.50 ± 0.58
Meta-Baseline[22]	ResNet12	68.62 ± 0.27	83.74 ± 0.18	-	-
DeepEMD[46]	ResNet12	71.16 ± 0.87	86.03 ± 0.58	75.65 ± 0.83	88.69 ± 0.50
Neg-cosine[47]	ResNet12	-	-	72.66 ± 0.85	89.40 ± 0.43
P-Transfer[48]	ResNet12	-	-	73.88 ± 0.87	87.81 ± 0.48
GLoFA[51]	ResNet12	69.75 ± 0.33	83.58 ± 0.42	-	-
AFHN[50]	ResNet18†	-	-	70.53 ± 1.01	83.95 ± 0.63
S2M2R[29]	ResNet18†	-	-	71.43 ± 0.43	85.55 ± 0.52
LEO[52]	WRN-28-10†	66.33 ± 0.05	81.44 ± 0.09	-	-
Our	ResNet12	**72.64 ± 0.22**	**86.99 ± 0.15**	**79.42 ± 0.60**	**90.17 ± 0.33**

Note: **Boldface** indicates the best results in each column, Underlined indicates the second-best results in each column, and '-' indicates that there are no experimental results available in the relevant literature. In the backbone network, † indicates a deeper network structure than ResNet12.

1) CALCULATE THE CLASS PROTOTYPES

For each class k in the few-shot learning task, its class prototype is calculated by taking the mean of the feature representations of all samples of that class in the support set. The class prototype can be represented as:

$$C_k = \frac{1}{|S_k|} \sum_{x \in S_k} f_\theta(x) \tag{10}$$

where $f_\theta(x)$ denotes the feature representation of the sample x, S_k is the set of all samples of class k in the support set, and N represents the total number of classes in the support set S.

2) CLASSIFY THE QUERY SET

For each sample q_i in the query set, the same feature embedding function is used to extract its feature representation to obtain $f_\theta(q_i)$. Calculate the distance between the feature vector of the query sample q_i and the prototypes of each class in the support set, and then use the softmax function to obtain the probability distribution of the query sample q_i belonging to a certain category, which is specifically calculated as:

$$p_\theta(y = k|q_i) = \frac{\exp(-d(f_\theta(q_i), C_k))}{\sum_{k'=1}^{N} \exp(-d(f_\theta(q_i), C_{k'}))} \tag{11}$$

where $d(\cdot, \cdot)$ is the distance metric function, using the Euclidean distance, k represents the true category of the query sample q_i, y represents the predicted category of the query sample q_i, and k' is the index of all classes in the support set.

In this way, the prototypical network can effectively handle few-shot classification problems by learning a mapping from the input to the feature space, and using class prototypes and distance metrics for classification decisions in this feature space.

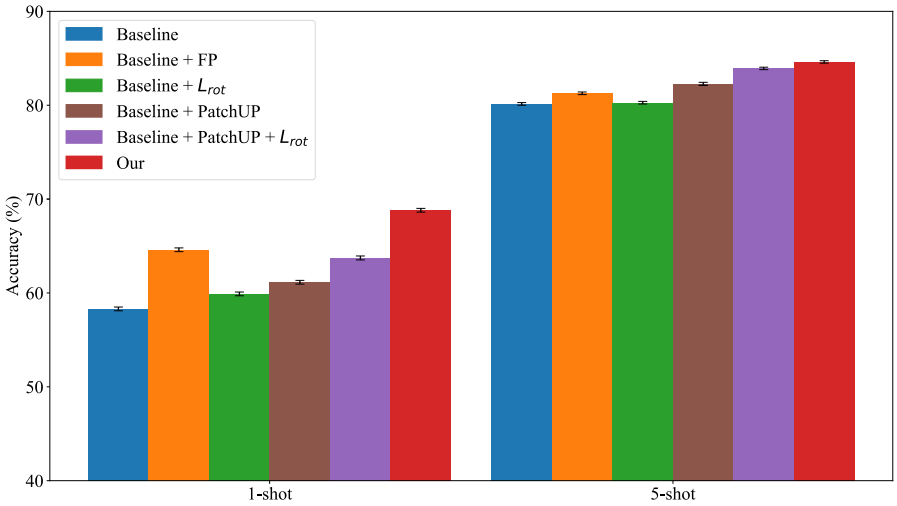
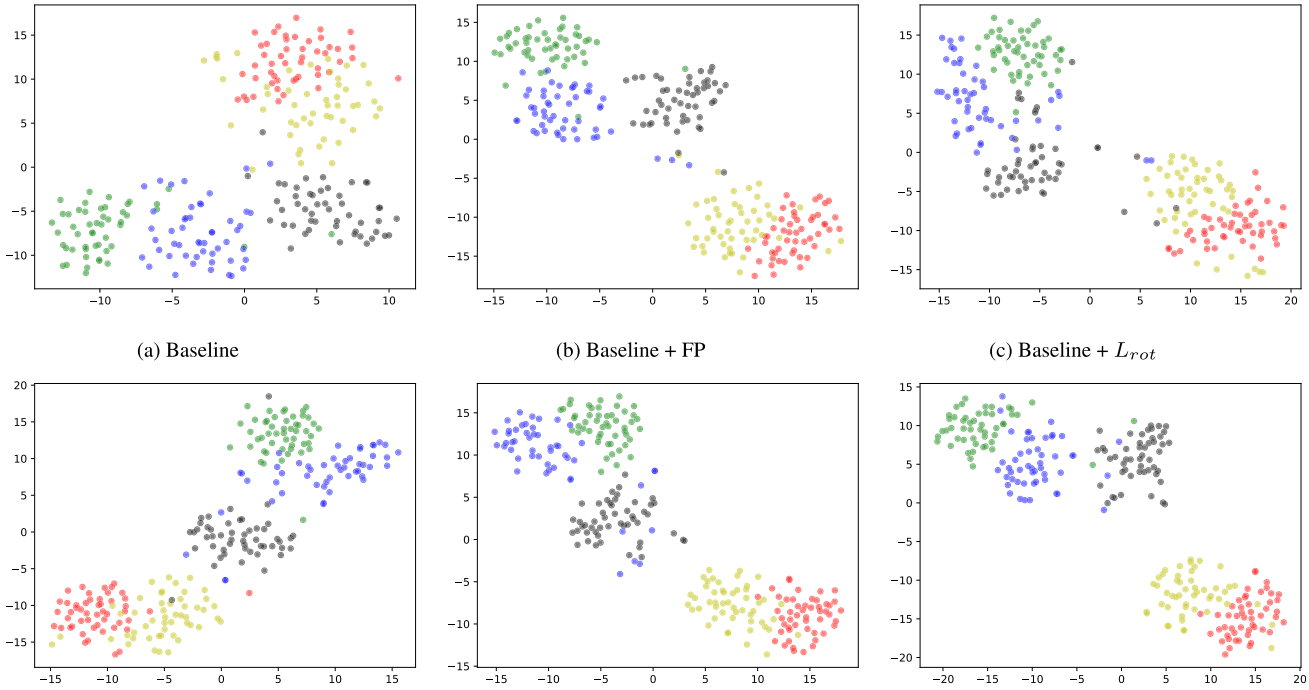


FIGURE 3. Comparison of different methods on 1-shot and 5-shot accuracy.



(a) Baseline (b) Baseline + FP (c) Baseline + L_{rot}

(d) Baseline + PatchUP (e) Baseline + PatchUP + L_{rot} (f) Baseline + PatchUP + L_{rot} + FP(our)

FIGURE 4. Visualization of t-SNE features for 50 images per category randomly selected from 5 categories in the test set of the MiniImageNet dataset.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL SETTING

To verify the feasibility of the proposed method, this paper tests and validates the model's performance on three classic few-shot public datasets: miniImageNet [43], tieredImageNet [24], FC-100 [44], and CUB-200-2011 [45], and compares it with other advanced methods. The division of the above dataset is shown in Table 1. Train indicates the number of base categories, Val indicates the number of validation categories, Test indicates the number of novel categories and Total indicates the original number of categories in the dataset. All images in the datasets are uniformly cropped to 84×84 for network training.

1) BACKBONE NETWORK

To make a fair comparison with current advanced methods, this paper uses the commonly used ResNet12 as the backbone network for feature extraction. It consists of 4 residual

blocks, each with 3 convolutional layers of 3 × 3 kernels, followed by a 2 × 2 max pooling layer at the end of each block. Intermediate layers of the backbone network use max pooling, ReLU nonlinear activation functions, and batch normalization operations. To generate dense features, a global average pooling layer is added at the end of the backbone network to produce a one-dimensional vector for each input image block.

2) EXPERIMENTAL DETAILS

During the model training phase, we follow the standard machine learning training paradigm to train our classification model. Specifically, we use ResNet12 as the backbone network, with the main classifier being an n-class linear classifier (n=total number of classes) for classifying all types in the training set. The auxiliary classifier predicts the rotation angle of the image. We use stochastic gradient descent (SGD) as the optimization algorithm, with the following parameters: initial learning rate of 0.1, momentum of 0.9, and weight decay of 5e-4. During training, we use a batch size of 128 and introduce a cosine annealing scheduler to adjust the learning rate, with the training process lasting for 150 epochs.

In the validation phase of training, we adopt the few-shot learning mode and perform classification testing on the validation set. Specifically, we randomly extract 500 episodes of 5-way k-shot (k=[1,5]), with k samples from each category as the support set and 15 samples as the query set for executing classification tasks.

In the testing phase, the model's performance is evaluated on 10,000 episodes, and accuracy with a 95% confidence interval is used as the evaluation metric.

B. EXPERIMENTAL RESULTS

In this study, our method was initially compared with the current state-of-the-art methods in terms of average accuracy for few-shot classification tasks. The comparative results for the 5-way 1-shot and 5-way 5-shot classification tasks are presented in Tables 2 and 3, respectively. The findings reveal that, in most cases, the performance of our method is significantly superior to that of existing methods. Specifically, on the MiniImageNet dataset, our method achieved an accuracy improvement of 2.90% and 2.20% over the second-best method for the 5-way 1-shot and 5-way 5-shot tasks, respectively. For the CUB dataset, the corresponding improvements in accuracy over the second-best method were 3.77% and 0.77%, respectively. On the tieredImageNet dataset, the respective improvements were 1.50% and 0.96%. Furthermore, on the FC-100 dataset, the results of our method were on par with the best-performing method, thereby demonstrating its competitive edge.

C. ABLATION EXPERIMENT

In order to further evaluate the contribution of each component in the proposed method, we conducted a series of ablation studies on the miniImageNet dataset. The results of the ablation experiment are summarized in Table 4 and Figure 3. The baseline model adopts ResNet12 as its backbone network for feature extraction, and is trained based on standard training paradigms. At the same time, a prototype classifier is used for small sample classification validation.

TABLE 4. Ablation study analysis on miniImageNet dataset by using ResNet12 as the backbone network.

Method	5-way 1-shot	5-way 5-shot
Baseline	58.30 ± 0.20	80.14 ± 0.14
Baseline + FP	64.60 ± 0.20	81.27 ± 0.14
Baseline + L_{rot}	59.90 ± 0.20	80.27 ± 0.14
Baseline + PatchUP	61.14 ± 0.20	82.28 ± 0.16
Baseline + PatchUP + L_{rot}	63.73 ± 0.21	83.92 ± 0.13
Baseline + PatchUP + L_{rot} + FP(Our)	$\mathbf{68.81 \pm 0.20}$	$\mathbf{84.61 \pm 0.13}$

Note: FP indicates the application of feature preprocessing. PatchUP refers to the use of PatchUp for data augmentation during training. L_{rot} denotes the inclusion of an angle-based self-supervised auxiliary loss during training.The best results are highlighted in bold.

TABLE 5. Comparison results of different data augmentation effects using ResNet-12 as the backbone network on miniImageNet.

Method	5-way 1-shot	5-way 5-shot
Cutmix	62.66 ± 0.21	81.62 ± 0.14
Mixup	61.06 ± 0.20	83.00 ± 0.13
Manifold mixup	61.65 ± 0.21	83.18 ± 0.13
PatchUP	$\mathbf{63.73 \pm 0.21}$	$\mathbf{83.92 \pm 0.13}$

Note: The model's performance is evaluated on 10,000 episodes using the miniImageNet test set and accuracy with a 95% confidence intervalis used as the evaluation metric. The best results are highlighted in bold.

1) ANALYSIS OF ABLATION EXPERIMENT RESULTS

In the 1-shot setting, the baseline model's accuracy significantly improved from 58.30% to 64.60% with the addition of feature preprocessing, marking an increase of 6.30%. In the 5-shot setting, accuracy increased from 80.14% to 81.27%, a rise of 1.13%. When angular loss was added to the baseline model, the accuracy in the 1-shot setting rose from 58.30% to 59.90%, an increase of 1.60%, while in the 5-shot setting, it slightly increased from 80.14% to 80.27%, a gain of 0.13%. Employing PatchUp for data augmentation resulted in the 1-shot setting accuracy improving from 58.30% to 61.14%, an increase of 2.84%, and in the 5-shot setting, accuracy rose from 80.14% to 82.28%, a gain of 2.14%. When both PatchUp data augmentation and angular loss were used simultaneously, the accuracy in the 1-shot setting increased from 58.30% to 63.73%, a gain of 5.43%, while in the 5-shot setting, it increased from 80.14% to 83.62%, an increase of 3.48%. Combining all optimization components resulted in a significant increase in the 1-shot setting accuracy to 68.81%, an improvement of 10.51% from the baseline, and in the 5-shot setting, accuracy increased to 84.61%, an improvement of 4.47% from the baseline. These results

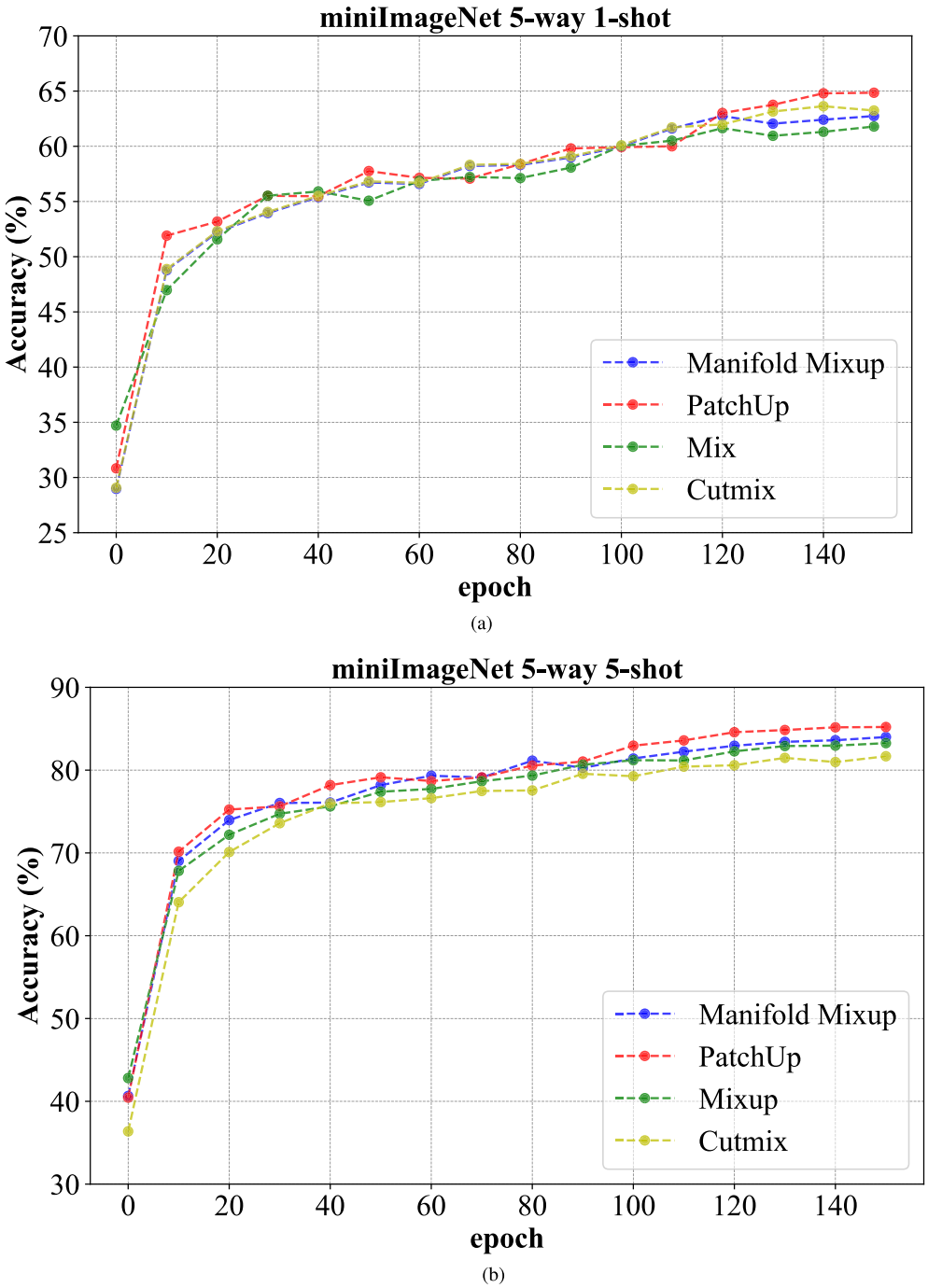


FIGURE 5. Accuracy variation with training for different data augmentation methods on 500 episodes in the miniImageNet val set.

highlight the importance of each component in enhancing the performance of few-shot classification tasks, especially when used together, demonstrating a significant synergistic effect.

2) VISUALIZATION OF ABLATION EXPERIMENT RESULTS

In order to comprehensively evaluate the effectiveness of the method proposed in this article, we employed t-SNE technology to conduct in-depth visual analysis of the generated features. We randomly selected 5 different categories from the test set of the MiniImageNet dataset, with 50 images selected for each category, and performed feature extraction and visualization processing. As shown in Figure 4, the visualization results of t-SNE clearly demonstrate the significant advantage of the features generated by our method in clustering performance. Specifically, these features exhibit smaller intra class distances and larger inter

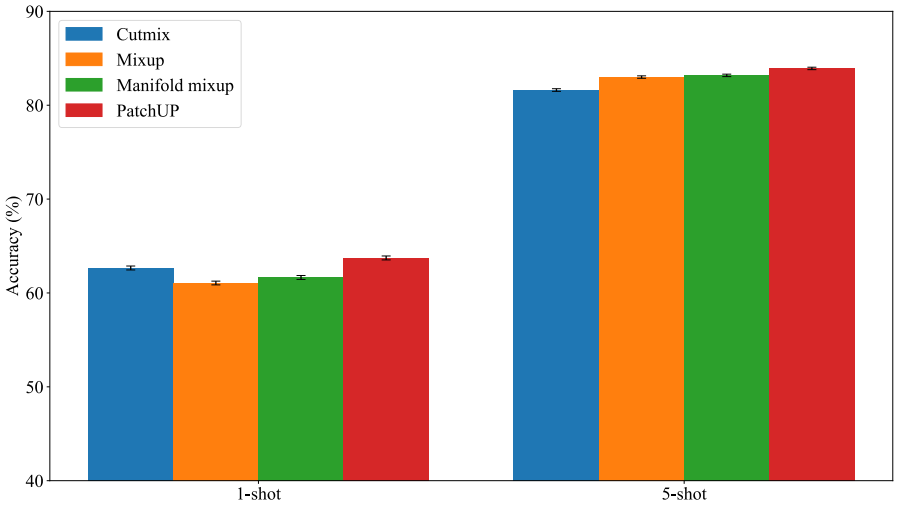


FIGURE 6. Comparison of different methods on 1-shot and 5-shot accuracy.

class distances, which enhance the discrimination between different categories and thus enhance the classification ability of the model.

D. THE EFFECTIVENESS OF PATCHUP

To evaluate the effectiveness of PatchUp [31] in few-shot image recognition, we conducted ablation studies on the miniImageNet dataset. We compared PatchUp with several other popular data augmentation techniques such as CutMix [33], Mixup [34], and Manifold Mixup [30]. To ensure fairness in comparison, all experiments were based on the same architecture, ResNet12 as the base network, and angle self supervised loss L_{rot} was added when applying other data augmentation strategies.

The convergence curves of the model on the miniImageNet dataset under the 5-way 1-shot and 5-way 5-shot settings are shown in Figure 5. The 95% confidence interval accuracy of the model on miniImageNet is presented in both Table 5 and Figure 6. The experimental results demonstrate that the PatchUP method performs the best among the data augmentation techniques. It achieved the highest accuracy of 63.73% in the 1-shot setting and 83.92% in the 5-shot setting. Cutmix performed well in the 1-shot setting but was slightly inferior to Mixup and Manifold Mixup in the 5-shot setting. Mixup excelled in the 5-shot setting with an accuracy of 83.00%, while Manifold Mixup showed consistent performance in both settings, particularly outperforming Mixup in the 5-shot setting. Overall, PatchUP significantly enhances the model's generalization ability, making it the most effective data augmentation method.

V. CONCLUSION

In conclusion, this study addresses the challenges of few-shot learning in image classification by proposing a novel approach based on metric learning. Our method leverages PatchUp techniques to perform block-level operations in the hidden layer feature space, enhancing feature diversity and expanding the range of data representations. This enables the model to form smoother classification decision boundaries, thereby improving its adaptability to variations in data. Additionally, the introduction of a self-supervised auxiliary loss function stabilizes the learning of deep semantic information, contributing to enhanced performance on new categories. Furthermore, centralizing and normalizing features extracted from the backbone network significantly bolster the model's performance in few-shot image classification tasks, particularly in challenging 1-shot learning scenarios. Through extensive experiments on multiple benchmark datasets including miniImageNet, tieredImageNet, FC-100, and CUB-200-2011, we validate the effectiveness and superior performance of our proposed method. While our approach has achieved significant advancements in the field of few-shot learning, there remains room for further optimization. Future work could explore more efficient data augmentation strategies and regularization techniques to enhance the model's generalization ability and robustness. Additionally, extending the application of this method to broader tasks and domains will validate its universality and applicability.

REFERENCES

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, Sep. 2018.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[3] Y. Lin, Z. Xie, T. Chen, X. Cheng, and H. Wen, "Image privacy protection scheme based on high-quality reconstruction DCT compression and nonlinear dynamics," *Expert Syst. Appl.*, vol. 257, Dec. 2024, Art. no. 124891.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[7] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–24.

[8] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," in *Proc. IEEE/CVF Int. Conf. Mach. Learn.*, Jul. 2018, pp. 3664–3673.

[9] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[10] A. Bär, N. Houlsby, M. Dehghani, and M. Kumar, "Frozen feature augmentation for few-shot image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2024, pp. 16046–16057.

[11] J. Zhou, Y. Zheng, J. Tang, J. Li, and Z. Yang, "FlipDA: Effective and robust data augmentation for few-shot learning," 2021, *arXiv:2108.06332*.

[12] J.-W. Seo, H.-G. Jung, and S.-W. Lee, "Self-augmentation: Generalizing deep networks to unseen classes for few-shot learning," *Neural Netw.*, vol. 138, pp. 140–149, Jun. 2021.

[13] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.

[14] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8805–8814.

[15] C. Zhang, H. Ding, G. Lin, R. Li, C. Wang, and C. Shen, "Meta navigator: Search for a good adaptation policy for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9415–9424.

[16] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, 2015, vol. 2, no. 1, pp. 1–24.

[17] O. Vinyals, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–29.

[18] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[19] U. Osahor and N. M. Nasrabadi, "Ortho-Shot: Low displacement rank regularization with data augmentation for few-shot learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2040–2049.

[20] Y. Meng, M. Michalski, J. Huang, Y. Zhang, T. Abdelzaher, and J. Han, "Tuning language models as training data generators for augmentation-enhanced few-shot learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 24457–24477.

[21] T. Qin, W. Li, Y. Shi, and Y. Gao, "Diversity helps: Unsupervised few-shot learning via distribution shift-based data augmentation," 2020, *arXiv:2004.05805*.

[22] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9042–9051.

[23] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "MetaFSCIL: A meta-learning approach for few-shot class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14146–14155.

[24] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," 2018, *arXiv:1803.00676*.

[25] J. Wang and Y. Zhai, "Prototypical Siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 178–181.

[26] S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales, "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2022, pp. 9068–9077.

[27] C. Fifty, D. Duan, R. G. Junkins, E. Amid, J. Leskovec, C. Re, and S. Thrun, "Context-aware meta-learning," 2023, *arXiv:2310.10971*.

[28] Y. Bendou, Y. Hu, R. Lafargue, G. Lioi, B. Pasdeloup, S. Pateux, and V. Gripon, "Easy—Ensemble augmented-shot-y-shaped learning: State-of-the-art few-shot classification with simple components," *J. Imag.*, vol. 8, no. 7, p. 179, 2022.

[29] P. Mangla, M. Singh, A. Sinha, N. Kumari, V. N. Balasubramanian, and B. Krishnamurthy, "Charting the right manifold: Manifold mixup for few-shot learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2207–2216.

[30] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6438–6447.

[31] M. Faramarzi, M. Amini, A. Badrinaaraayanan, V. Verma, and A. Chandar, "Patchup: A feature-space block-level regularization technique for convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 36, no. 1, pp. 589–597.

[32] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "SimpleShot: Revisiting nearest-neighbor classification for few-shot learning," 2019, *arXiv:1911.04623*.

[33] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.

[34] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[36] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–26.

[37] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[38] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5275–5285.

[39] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[40] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[41] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 649–666.

[42] K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J. Y. Zhang, Y. Liang, G. Pang, D. Song, and S. Pan, "Self-supervised learning for time series analysis: Taxonomy, progress, and prospects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 1, pp. 1–20, Sep. 2024.

[43] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, and C. Gagné, "Matching feature sets for few-shot image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9004–9014.

[44] B. Oreshkin, P. Rodríguez López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, 2011.

[46] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable Earth mover's distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12200–12210.

[47] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, "Negative margin matters: Understanding margin in few-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 438–455.

[48] Z. Shen, Z. Liu, J. Qin, M. Savvides, and K.-T. Cheng, "Partial is better than all: Revisiting fine-tuning strategy for few-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 11, 2021, pp. 9594–9602.

[49] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2019, pp. 1–26.

[50] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13467–13476.

[51] S. Lu, H.-J. Ye, and D.-C. Zhan, "Tailoring embedding function to heterogeneous few-shot tasks by global and local feature adaptors," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 10, pp. 8776–8783.

[52] Y. Wang, C. Huang, M. Li, Q. Huang, X. Wu, and J. Wu, "AG-meta: Adaptive graph meta-learning via representation consistency over local subgraphs," *Pattern Recognit.*, vol. 151, Jul. 2024, Art. no. 110387.

LEI ZHANG received the B.S. degree in engineering from Sichuan Agricultural University, in 2021. He is currently pursuing the master's degree in electronic information with the University of Electronic Science and Technology of China.

From 2023 to 2024, he was an outstanding employee with Beijing Dajia Internet Information Technology Company Ltd. He has published two articles and holds two invention patents. His research interests include computer science, image processing, and few-shot image classification.

YITING LIN (Member, IEEE) received the B.S. degree in computer science and technology from the University of Electronic Science and Technology of China Zhongshan Institute, China, in 2024.

From 2020 to 2024, he was a Research Assistant with the Intelligent IoT and Secure Communication Innovation Laboratory, University of Electronic Science and Technology of China. Since 2024, he has been a Research Professor with the School of Computer Science, University of Electronic Science and Technology of China Zhongshan Institute. He has published 16 articles, five invention patents, and more than ten computer software copyrights. His research interests include multimedia security, computer science, image processing, information security, signal processing, cryptography, and nonlinear dynamics. He is a reviewer of numerous international journals.

XINYU YANG received the bachelor's degree in communication engineering from Wuhan University of Technology, in 2023. He is currently pursuing the master's degree in electronic information with the University of Electronic Science and Technology of China.

His research interests include multimedia security, computer science, image processing, information security, signal processing, cryptography, and nonlinear dynamics.

TINGTING CHEN received the B.S. degree from the University of Electronic Science and Technology of China Zhongshan Institute, China, in 2024.

From 2021 to 2022, she was an outstanding employee with China United Network Communications Corporation (Zhongshan Branch). From 2022 to 2023, she was the Head of the Data Review Team, Zhongshan Environmental Protection Science Research Institute. From 2023 to 2024, she was a Research Assistant with the Intelligent IoT and Secure Communication Innovation Laboratory, University of Electronic Science and Technology of China. Her research interests include data science, information security, education, business administration, and economics.

XIYUAN CHENG received the B.S. degree in biomedical engineering from Northeastern University, Shenyang, China, in 2020.

She is currently working toward the M.S. degree with the School of Automation, Guangdong University of Technology. Her research interests include information security, image processing, domain adaptation, deep learning control systems, and electric engineering.

WENBIN CHENG received the B.S. degree from Information Engineering University, China, and the M.S. degree from the University of Electronic Science and Technology of China, China.

He is currently a Professor with the University of Electronic Science and Technology of China Zhongshan Institute and a Graduate Tutor with the University of Electronic Science and Technology of China. He is the Head of the Embedded Technology and IoT Laboratory. He has published 12 papers as the first author, obtained more than 20 patents, registered more than ten computer software copyrights, and participated in the compilation of the Guangdong provincial local technical standard "Ultra-high Frequency RFID Chip Test Method." He is currently a member of the Expert Technical Committee of the National Local Joint Engineering Research Center for RFID and IoT Tag Technology, and the Director of the Internet of Things Technology and Application Center, Zhongshan Industrial Research Institute. His research interests include computer science, the Internet of Things, artificial intelligence, information security, networked control systems and electric engineering.