



# Boosting few-shot rare skin disease classification via self-supervision and distribution calibration

Wen Fu<sup>1,2</sup> · Jie Chen<sup>1</sup> · Li Zhou<sup>1</sup>

Received: 10 January 2024 / Revised: 22 April 2024 / Accepted: 25 April 2024 / Published online: 20 May 2024  
© Korean Society of Medical and Biological Engineering 2024

## Abstract

Due to the difficulty in obtaining clinical samples and the high cost of labeling, rare skin diseases are characterized by data scarcity, making training deep neural networks for classification challenging. In recent years, few-shot learning has emerged as a promising solution, enabling models to recognize unseen disease classes by limited labeled samples. However, most existing methods ignored the fine-grained nature of rare skin diseases, resulting in poor performance when generalizing to highly similar classes. Moreover, the distributions learned from limited labeled data are biased, severely impairing the model's generalizability. This paper proposes a self-supervision distribution calibration network (SS-DCN) to address the above issues. Specifically, SS-DCN adopts a multi-task learning framework during pre-training. By introducing self-supervised tasks to aid in supervised learning, the model can learn more discriminative and transferable visual representations. Furthermore, SS-DCN applied an enhanced distribution calibration (EDC) strategy, which utilizes the statistics of base classes with sufficient samples to calibrate the bias distribution of novel classes with few-shot samples. By generating more samples from the calibrated distribution, EDC can provide sufficient supervision for subsequent classifier training. The proposed method is evaluated on three public skin disease datasets (i.e., ISIC2018, Derm7pt, and SD198), achieving significant performance improvements over state-of-the-art methods.

**Keywords** Rare skin disease classification · Deep learning · Few-shot learning · Self-supervised learning

## 1 Introduction

Skin diseases are a global health concern that can significantly impact individuals' quality of life. Early and accurate diagnosis is crucial for determining appropriate treatment strategies and improving patient outcomes [2]. Deep learning has emerged as a promising approach in dermatology by analyzing medical images to aid in identifying and classifying various skin conditions [3–5]. However, obtaining adequate annotated samples can be difficult due to patient

privacy concerns and labeling costs. Figure 1 shows that rare skin diseases are located in the tail of the distribution, which is characterized by a limited number of labeled samples. In this situation, deep learning models must possess strong generalization capabilities, enabling them to quickly adapt and accurately classify unseen disease classes with limited samples. The above issues make the classification of rare skin diseases still a challenge.

Few-shot learning (FSL) is a feasible approach to address the issues mentioned above. The methods in FSL for classifying rare skin diseases are divided into two categories: meta-learning and transfer learning. Meta-learning aims to train models to learn how to learn. In this direction, model-agnostic meta-learning (MAML) [6], and prototypical networks [7] are widely used and improved [1, 8–13]. As for transfer learning, it allows models to utilize knowledge learned from related tasks or datasets. By pre-training the model on the base classes containing sufficiently labeled samples, such methods [14–16] allows the model to be quickly adapted to rare disease classes with fewer samples. There are also methods [17, 18] that adopt a cross-domain

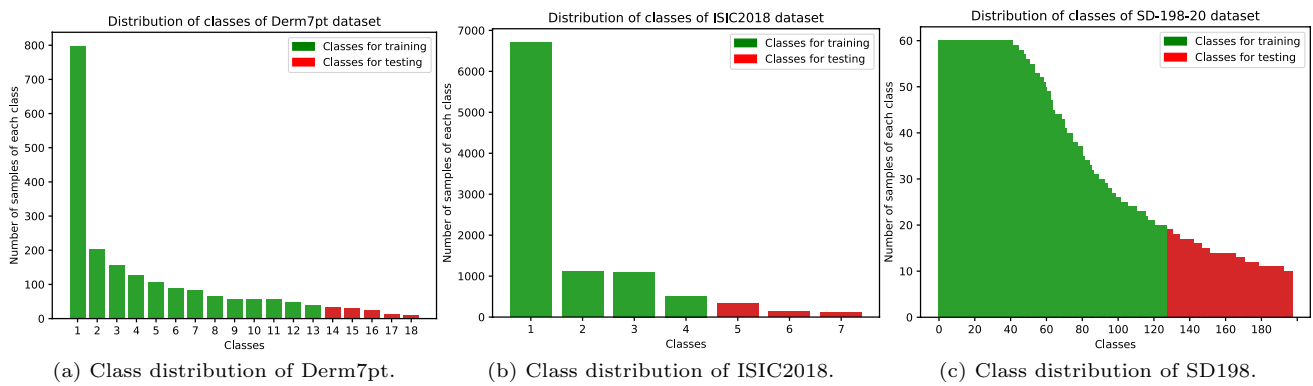
✉ Wen Fu  
fuwen2020@ime.ac.cn

Jie Chen  
jchen@ime.ac.cn

Li Zhou  
zhouli@ime.ac.cn

<sup>1</sup> Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China

<sup>2</sup> University of Chinese Academy of Sciences, Beijing 100049, China



**Fig. 1** Visualization of class distributions for three skin disease datasets. Following the setting of [1], use the categories at the head of the distribution (common diseases) as training/base classes and the

classes at the tail of the distribution (new/rare diseases) as testing/novel classes, marked with red and green, respectively

setting, i.e., the base and novel classes belong to different domains. However, the existence of a domain gap makes the classification performance unsatisfactory.

It should be noted that despite some advancements in current research, the classification of rare skin diseases through few-shot learning remains a challenge for two reasons. Firstly, most existing methods have not been optimized for the fine-grained nature of rare skin diseases, making them struggle to perform well when generalizing to highly similar classes. Secondly, the model is prone to overfitting when generalizing to few-shot novel classes. The data distribution derived from limited samples typically exhibits a bias compared to the actual distribution, leading to further degradation in the model's performance. Self-supervised learning (SSL) presents a viable solution for addressing fine-grained classification problems and has demonstrated promising results across various medical classification tasks [19]. By pre-training the model with pretext tasks defined by the visual information in large-scale unlabeled images, SSL alleviates the model's dependence on the labeled samples of the downstream task and significantly enhances the model's generalizability. Additionally, previous work has shown that similar classes exhibit similar statistics (i.e., mean and variance) when features adhere to the Gaussian distribution [20]. Consequently, the bias distribution of the few-shot novel classes can be calibrated by transferring the more accurate statistics of the similar base classes with sufficient samples.

This paper proposes a Self-Supervision Distribution Calibration Network (SS-DCN) to further improve the accuracy of rare skin disease diagnosis in the few-shot setting. SS-DCN adopts a multi-task learning framework during pre-training to help the model learn more discriminative and transferable visual representations. Specifically, the multi-task learning framework contains two auxiliary self-supervised tasks: rotation prediction and contrastive learning. As a powerful self-supervised learning paradigm, contrastive

learning aims to bring representations from the same class closer in the embedding space while representations from different classes are far away. The rotation prediction task is used to help the model learn rotational invariance, which has been proven beneficial for classifying dermatologic images [1]. Furthermore, an enhanced distribution calibration (EDC) strategy is applied to alleviate the biased distribution problem caused by a lack of data for novel classes. The biased distribution of novel classes is calibrated by reusing the statistics of base classes with sufficient samples, and more samples can be generated from the calibrated distribution to augment the classifier's input, further improving classification performance. Our main contributions are summarized as follows:

- A Self-Supervision Distribution Calibration network (SS-DCN) is proposed, and a multi-task learning framework is used during pre-training to help the model learn more discriminative and transferable visual representations.
- An Enhanced Distribution Calibration strategy (EDC) is proposed to address the biased distribution of few-shot novel classes. By generating more samples from the calibrated distribution, EDC can provide sufficient supervision for subsequent classifier training, improving the discrimination of the decision boundary.
- The proposed method is evaluated on three public skin disease datasets, and advanced results are achieved.

## 2 Related work

### 2.1 Few-shot image classification

Few-shot image classification (FSIC) has received considerable attention in recent years, which aims to recognize

unseen categories by limited labeled samples [21]. According to recent research, mainstream FSIC methods are divided into two paradigms [21]: meta-learning-based and non-meta-learning-based. Specifically, meta-learning-based methods contained three categories: metric-based, optimization-based, and hallucination-based. Metric-based methods focus on learning good embeddings as well as metrics [7, 22]. Optimization-based methods focus on learning a good optimizer or a well-initialized model that can quickly generalize to novel classes [23, 24]. Hallucination-based methods expand prior knowledge by generating more samples to alleviate the overfitting problem caused by the lack of labeled data [25, 26].

Non-meta-learning-based methods mainly refer to transfer learning [27, 28]. Transfer learning is a viable option in few-shot image classification scenarios where the data is too limited to train a deep model from scratch. Specifically, a model is pre-trained on base classes with a large amount of training data and then fine-tuned on few-shot unseen classes. Non-meta-learning-based methods have been proven to achieve good performance when downstream tasks lack sufficient samples and are explored in this paper.

## 2.2 FSIC for skin disease

Researchers have tried combining few-shot learning with specific application scenarios in recent years [29, 30]. In this paper, the task of rare skin disease classification is studied.

One mainstream method in FSL for rare skin disease classification is meta-learning, which trains models to learn how to learn. In this field, several methods have made improvements based on MAML [6]. ST-MetaDiagnosis [8] and Meta-Derm [1] enhance classification performance by learning transformation invariance through model structure design, allowing the models to handle skin disease image variations better. DAML [10] adjusts the task's weights based on its difficulty, allowing the model to adapt to novel classes more effectively. MetaMed [9] improves the model's generalizability by introducing advanced data augmentation technology. Some methods are making improvements based on prototypical networks [7]. QR loss [31] addresses the incompatibility of cross-entropy loss with episode training and proposes the query relative loss to better utilize cross-sample information. IPNet [12] proposes an influential prototypical network in which the influence weights of samples are calculated based on maximum mean discrepancy (MMD) between the mean embeddings of sample distributions that include and exclude the sample. Subspace [32] utilizes adaptive subspace to construct symmetric functions and effectively combines two similarity metrics to improve performance. Another mainstream method in FSL for rare skin disease classification is transfer learning. By pre-training the model on the classes containing a large number of labeled

samples, Meta-Rep [16], Med-tumor [17], and [18] can quickly adapt the model to rare classes with fewer samples.

Some recent work has been devoted to solving the fine-grained challenge. PCN [11] and SCAN [14] argue that when the same type of disease is located in different body parts, it will exhibit different appearances. By learning the sub-cluster structure contained in the class, they enhanced the model's ability to recognize fine-grained features. However, such methods rely heavily on the dataset's characteristics, making it difficult to distinguish between classes that are only slightly different in appearance. Pre-MocoDiagnosis [15], similar to our method, solves this problem by combining supervised and self-supervised learning. However, it trains the two targets separately, whereas our method trains them jointly.

## 2.3 Self-supervised learning

Self-supervised learning (SSL) is a promising technique that can effectively reduce the model's dependence on labeled data [33]. It can be categorized into three main paradigms: prediction-based, generative-based, and contrastive-based.

Specifically, prediction-based approaches involve defining pretext tasks that require the model to predict specific properties of the input data, such as image rotation [34] or patch relative position prediction [35]. Among generative approaches, the MIM method has recently gained significant attention. MIM utilizes the co-occurrence relationships between image patches as supervision signals to help the network learn meaningful feature representations [33]. As for contrastive-based approaches, early methods, such as MoCo v2 [36] and SimCLR [37], rely on negative samples. However, recent advancements have proposed methods that do not require explicit negative samples, i.e., SimSiam [38], and Barlow Twins [39]. Contrastive-based approaches have shown promising results across various vision tasks. Learning more discriminative and transferable visual representations through the contrastive target enables models to generalize to downstream tasks better and achieve higher performance even with limited labeled data.

# 3 Methodology

## 3.1 Problem formulation

A standard labeled dataset is further divided into a base and a novel dataset in the typical few-shot image classification setting. Denote the base dataset as  $D_{base} = \{(x, y)\} \subset I \times C_{base}$ , where  $x \in I$  are  $N_{base}$  images labeled  $y$  in the base class  $C_{base}$ . The label spaces of the novel dataset  $D_{novel} = \{(x, y)\} \subset I \times C_{novel}$  is mutually exclusive with the base dataset  $D_{base}$ .

The purpose of FSIC is to learn the transferable knowledge from the seen categories, which can be generalized to the unseen categories with limited labeled instances. Following previous work [1, 14, 16], the model is trained on  $D_{base}$ , and the episodic strategy is used on  $D_{novel}$  to evaluate the generalization ability. Specifically, each episode contains a support set  $S = \{(x_i, y_i)\}_{i=1}^{NK}$  for fast adaptation, and a query set  $Q = \{(x_j)\}_{j=1}^{NT}$  for evaluation, where  $N$  represents the number of categories randomly sampled from the  $D_{novel}$ ,  $K$  is the number of labeled instances in each category, and  $T$  represents the number of unlabeled instances randomly sampled from the same  $N$  categories. Such an episode is also called an  $N$ -way  $K$ -shot task. Under this setting, the model's performance is evaluated as the average classification results on batch tasks sampled from  $D_{novel}$ .

### 3.2 Overall framework description

As shown in Fig. 2, the proposed framework has three stages: pre-training, classifier training, and inference. In the pre-training stage, a multi-task learning framework enhances the discrimination and generalization of the backbone. The frozen pre-trained backbone is utilized in the classifier training phase, and simple classifiers are trained for each task. Furthermore, by transforming the extracted features into a Gaussian-like distribution, the statistical information from base classes is leveraged to calibrate the biased distribution of novel classes, and generate more samples to augment the classifier's input. Finally, in the inference phase, the backbone and the classifier are frozen to classify query samples efficiently.

### 3.3 Pre-training stage

#### 3.3.1 Multi-task learning

Multi-task learning is a powerful technique that allows a model to learn multiple tasks simultaneously [40]. In this paper, the focus is on improving the performance of the main task, i.e., the supervised classification task, by jointly training with self-supervised pretext tasks. As shown in Fig. 3, self-supervised losses are incorporated as regularizers into the pre-training stage's training loss. This paper integrates two auxiliary self-supervision tasks: rotation prediction and contrastive learning.

Specifically, four rotated copies of an image  $x$  are created, denoted as  $R = \{x_r | r \in (0^\circ, 90^\circ, 180^\circ, 270^\circ)\}$ , where  $x_r$  is  $x$  rotated by  $r$  degrees. The model performs supervised classification and rotation prediction on  $R$ . Meanwhile, two different views of  $x$ ,  $x_1$ , and  $x_2$ , are obtained for contrastive learning by applying two sets of data augmentations. The same backbone  $F_\theta(\cdot)$  is used for all pretext tasks to extract features, while different classifiers or predictors are defined for different pretext tasks. Finally, the training loss in the pre-training stage is as follows:

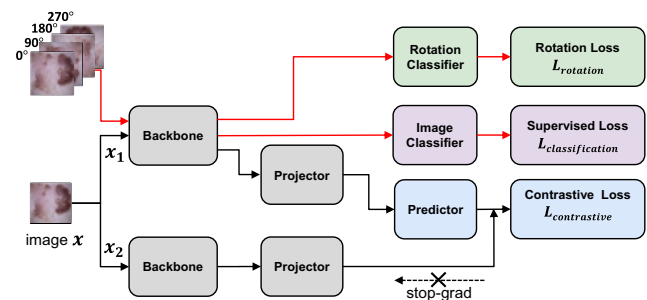


Fig. 3 Multi-task learning framework for pre-training

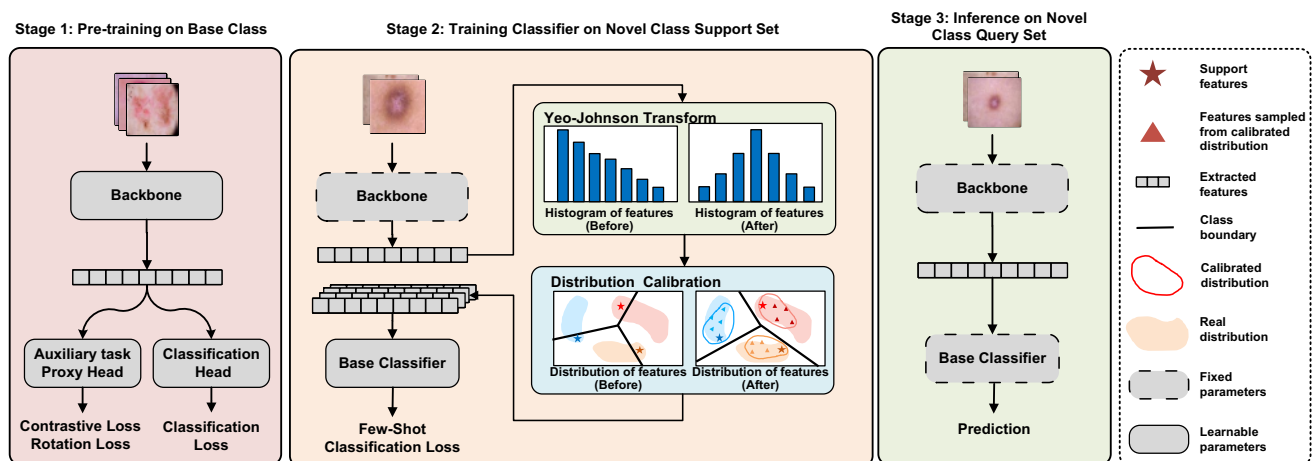


Fig. 2 The framework of the proposed method, where the corresponding components are shown in the dashed boxes on the right

$$L_{total} = L_{classification} + L_{contrastive} + wL_{rotation} \quad (1)$$

where  $w$  is a weighting factor used to adjust the proportion of  $L_{rotation}$  in the  $L_{total}$ .

### 3.3.2 Supervised learning

As shown in Fig. 3, a feature extractor  $F_\theta(\cdot)$  and a linear classifier  $C(\cdot|W_{class})$  are trained for classifying the base classes in the pre-training stage. Performing standard classification tasks on the base classes during pre-training has been proven beneficial for the learning of downstream base classifiers [41]. Specifically, the prediction category  $\hat{y}_i$  of the input image  $x_i \in R$  can be expressed as:

$$P(y = c|x_i) = \text{Softmax}(C(F_\theta(x_i)|W_{class})) \quad (2)$$

$$\hat{y}_i = \arg \max_c \mathcal{P}(y = c | x_i) \quad (3)$$

where  $c = 1, 2, \dots, C_{base}$ . The supervised classification loss  $L_{classification}$  is obtained by computing the cross-entropy loss between the predicted and true labels.

$$L_{classification} = -\frac{1}{R} \sum_{i=1}^R \log \mathcal{P}(y = y_i | x_i) \quad (4)$$

where  $y_i$  is the true label of the image  $x_i$ .

### 3.3.3 Rotation prediction

In the rotation prediction task, the model is trained to predict the 2D rotation angle of a given image. This process forces the model to accurately identify and localize salient targets in the image and recognize their orientation, significantly enhancing its learning of rotational invariance.

Specifically, the model first extracts features using  $F_\theta(\cdot)$  from the rotated images. These features contain essential information about the object's semantics and orientation. Then, the rotation classifier  $C(\cdot|W_{rotation})$  utilizes these features to predict the rotation angle  $r$ . Similar to  $L_{classification}$ , the rotation loss  $L_{rotation}$  can be expressed as:

$$L_{rotation} = -\frac{1}{R} \sum_{i=1}^R \log P(r = r_i | x_i) \quad (5)$$

where  $r_i$  is the true rotation label of the image  $x_i$ .

### 3.3.4 Contrastive learning

Contrastive learning is one of the Self-supervision techniques that enables the model to capture the relevant structures and differences in the instances, resulting in more discriminative and generalizable features. In recent years,

contrastive learning has shown promising results on various fine-grained image classification tasks [33]. SimSiam is a practical and straightforward solution in this field, as it has been demonstrated to learn effective feature representations without the need for (1) negative sample pairs, (2) a large batch size, and (3) a momentum encoder.

As shown in Fig. 3, two different augmented views  $x_1$  and  $x_2$  of  $x$  are fed into the same encoding network, which consists of a backbone and a projector. The purpose is to extract features and project them into a high-dimensional space. Additionally, a predictor  $h$  is introduced after the encoding network that transforms the output of one branch. The predictor  $h$ 's role is to introduce diversity and increase the difficulty of the self-supervised learning task. By transforming one of the branches, the model is encouraged to capture more fine-grained details and learn richer representations, improving the discriminative power and generalizability of the learned features. Denoting the two outputs as  $p_1 = h(F_\theta(x_1))$ ,  $z_2 = F_\theta(x_2)$ , the transformed output  $p_1$  with the output  $z_2$  of the other branch are matched by minimizing the negative cosine similarity between them:

$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \quad (6)$$

where  $\|\cdot\|_2$  represent  $L_2$ -norm. Following [38], the self-supervised loss for this task is defined as a symmetrized loss, which encourages consistency between the two branches and helps in learning meaningful representations:

$$L_{contrastive} = \frac{1}{2} (\mathcal{D}(p_1, \hat{z}_2) + \mathcal{D}(p_2, \hat{z}_1)) \quad (7)$$

where  $\hat{z}_1, \hat{z}_2$  represents  $\text{stopgrad}(z_1)$ ,  $\text{stopgrad}(z_2)$ , respectively. The  $\text{stopgrad}(\cdot)$  operation plays an essential role in alleviating collapsing [38].

## 3.4 Classifier training stage

As shown in Fig. 2, the backbone is frozen after pre-training, and simple classifiers are trained for each task sampled from novel classes. In addition, to further improve the model's performance, an enhanced distribution calibration strategy is introduced at this stage.

### 3.4.1 Enhanced distribution calibration

As mentioned before, the bias distribution of the few-shot novel classes can be calibrated by transferring the more accurate statistics of the similar base classes with sufficient samples. To achieve this, the Yeo-Johnson transform [42] is first applied to the features extracted from both the base and novel classes to make their distributions more Gaussian-like. The Yeo-Johnson transform is an extended version of the Box-Cox transform [43] that can handle a wider range

of data, including negative values. The formula for the Yeo–Johnson transform is as follows:

$$x_i^{(\lambda)} = \begin{cases} \frac{[(x_i+1)^\lambda - 1]}{\lambda} & \text{if } \lambda \neq 0, x_i \geq 0 \\ \ln(x_i + 1) & \text{if } \lambda = 0, x_i \geq 0 \\ -\frac{[(-x_i+1)^{2-\lambda} - 1]}{(2-\lambda)} & \text{if } \lambda \neq 2, x_i < 0 \\ -\ln(-x_i + 1) & \text{if } \lambda = 2, x_i < 0 \end{cases} \quad (8)$$

where  $x_i$  represents the features extracted by the backbone, and  $x_i^{(\lambda)}$  represents the features after Yeo–Johnson transform.  $\lambda$  is a hyperparameter that determines the way and degree of data transformation. Subsequently, the mean and variance of each base class are calculated as:

$$\mu_i = \frac{\sum_j^{n_i} x_j^{(\lambda)}}{n_i}, \Sigma_i = \frac{\sum_j^{n_i} (x_j^{(\lambda)} - \mu_i)(x_j^{(\lambda)} - \mu_i)^T}{n_i - 1} \quad (9)$$

where  $x_j^{(\lambda)}$  is the transformed feature of the  $j$ -th instance from  $i$ -th base class, and  $n_i$  is the total number of instances in class  $i$ .

To improve the accuracy of estimating the distribution of novel classes, statistics from base classes that are similar to the novel class are reused. Specifically, for an  $N$ -way 1-shot task sampled from  $D_{novel}$ , the Euclidean distance between the feature  $x_n^{(\lambda)}$  of the support set and the mean of each base class is calculated. The top  $k$  base classes that are closest to  $x_n^{(\lambda)}$  are then selected to correct the distribution of the novel class:

$$\mathbb{S}_n^{distance} = \left\{ -\|\mu_i - x_n^{(\lambda)}\|^2 \mid i \in C_{base} \right\} \quad (10)$$

$$\mathbb{S}_n^{selected} = \left\{ i \mid -\|\mu_i - x_n^{(\lambda)}\|^2 \in \text{topk}(\mathbb{S}_n^{distance}) \right\} \quad (11)$$

$$\mu'_n = \frac{\sum_{i \in \mathbb{S}_n^{selected}} \mu_i + x_n^{(\lambda)}}{k+1}, \Sigma'_n = \frac{\sum_{i \in \mathbb{S}_n^{selected}} \Sigma_i}{k} \quad (12)$$

where  $n = 1, 2, \dots, N$ , represents the  $n$ -th novel class.  $\mathbb{S}_n^{distance}$  is a set that contains the Euclidean distances between the feature  $x_n^{(\lambda)}$  of the support set and each base class.  $\mathbb{S}_n^{selected}$  is a set that contains the  $k$  base classes most similar to the support feature  $x_n^{(\lambda)}$ .  $\mu'_n$  and  $\Sigma'_n$  denote the calibrated mean and covariance of the  $n$ -th novel class, respectively.

In the case of multi-shot scenes, distribution calibration operations are performed multiple times, with each operation using one feature from the support set. For a class  $n \in C_{novel}$ , the calibrated statistics are denoted as  $\mathbb{S}_n^{calibrated} = \{(\mu'_1, \Sigma'_1), \dots, (\mu'_K, \Sigma'_K)\}$ , where  $K$  represent the number of shots,  $\mathbb{S}_n^{calibrated}$  contains the calibrated mean and covariance for each shot of class  $n$ .

### Algorithm 1 Procedure of SS-DCN

---

**input** : base dataset( $D_{base}$ ), novel dataset( $D_{novel}$ ), learning rate( $\xi$ ), weighting factor( $w$ ), the number of epochs ( $n_{epoch}$ ), the number of  $N$ -way  $K$ -shot few shot tasks( $n_{tasks}$ ), the number of generated feature( $N_{generate}$ )

**output**: Average accuracy and Average marco-AUC for  $n_{tasks}$  few shot tasks

---

```

1 /* Pre-training Stage;
2 Randomly initialize model weight  $\theta$ ;
3 for  $n \leftarrow 1$  to  $n_{epoch}$  do
4   for  $i \leftarrow 1$  to  $batchsize$  do
5     Randomly sample  $(x_i, y_i)$  from  $D_{base}$ ;
6     Obtain rotation set  $R_i$  and two different views of  $x_i$ ;
7     Obtain multi-task Loss:  $L_{total} += (L_{classification} + L_{contrastive} + wL_{rotation})$ ;
8   Updating model:  $\theta \leftarrow \theta - \xi \cdot \nabla_{\theta} L_{total}$ ;
9 /*Enhanced Distribution Calibration;
10 Frozen backbone's parameter  $\theta$ ;
11  $x_{base}^{(\lambda)} = F_{\theta}(D_{base})$ ,  $x_{novel}^{(\lambda)} = F_{\theta}(D_{novel})$ ;
12  $x_{base}^{(\lambda)} = \text{Yeo-Johnson transform}(x_{base})$ ;
13 Obtain the statistics of base classes;
14 for  $i \leftarrow 1$  to  $n_{tasks}$  do
15   Randomly sample  $(S_i, Q_i)$  from  $x_{novel}$ ;
16    $S_i^{(\lambda)} = \text{Yeo-Johnson transform}(S_i)$ ;
17    $Q_i^{(\lambda)} = \text{Yeo-Johnson transform}(Q_i)$ ;
18   Obtain the calibrated statistics:  $\mathbb{S}_{calibrated} = \text{Distribution Calibration}(S_i^{(\lambda)}, x_{base}^{(\lambda)})$ ;
19   /* Classifier-training Stage;
20   Generating  $N_{generate}$  samples for each class  $n$ , denoted as  $\mathbb{D}_n$ ;
21   Trained a simple classifier on  $\mathbb{D}_{aug} = S_i^{(\lambda)} \cup \mathbb{D}_1 \cup \dots \cup \mathbb{D}_N$ ;
22   /* Inference Stage;
23   Frozen classifier's parameter;
24   Calculate accuracy and marco-AUC on  $Q_i^{(\lambda)}$ ;
25 Calculate Average accuracy and Average marco-AUC for  $n_{tasks}$  few shot tasks;
```

---

### 3.4.2 Simple classifier training

To enhance the classifier's training, more samples are generated from the calibrated distribution. Specifically, for a novel class  $n$ , a set of features is sampled from the calibrated Gaussian distribution:

$$\mathbb{D}_n = \{(x, n) | x \sim \mathcal{N}(\mu, \Sigma + \alpha), \forall (\mu, \Sigma) \in \mathbb{S}_n^{\text{calibrated}}\} \quad (13)$$

where  $\alpha$  is a hyperparameter used to control the diversity and dispersion of generated features by adjusting the covariance matrix. It is worth noting that the total number of generated features  $N_{\text{generate}}$  for each class is set as a hyperparameter. In the  $K$ -shot scenario,  $N_{\text{generate}}/K$  features are generated for each shot's calibrated distribution. Finally, for each  $N$ -way  $K$ -shot task, the generated features and the transformed support set features are utilized as training data for a simple task-specific classifier:

$$\mathbb{D}_{\text{aug}} = S^{(\lambda)} \cup \mathbb{D}_1 \cup \dots \cup \mathbb{D}_N \quad (14)$$

$$P(y = c | x_i) = \text{Softmax}(C(F_\theta(x_i) | W_{\text{simple}})) \quad (15)$$

where  $S^{(\lambda)}$  denotes the support set features transformed by Yeo–Johnson transformation and  $x_i \in \mathbb{D}_{\text{aug}}$ .  $C(\cdot | W_{\text{simple}})$  denotes the simple classifier. In this paper, it is a logistic regression classifier. The classifier is trained for each task by minimizing the cross-entropy loss:

$$L_{\text{simple}} = -\frac{1}{|\mathbb{D}_{\text{aug}}|} \sum_{i=1}^{|\mathbb{D}_{\text{aug}}|} \log P(y = y_i | x_i) \quad (16)$$

where  $y_i$  is the true label of the image  $x_i$ .

### 3.5 Inference stage

Once the simple classifier is trained for the specific task, all parameters are frozen, and perform classification on the query set  $Q$ :

$$P(y = n | x_j) = \text{Softmax}(C(F_\theta(x_j) | W_{\text{simple}})), x_j \in Q \quad (17)$$

where  $F_\theta(x_j)$  represents the extracted query feature.  $C(\cdot | W_{\text{simple}})$  is the trained simple classifier.

The procedure of the proposed method is shown in Algorithm 1.

## 4 Experiments

### 4.1 Datasets and preprocessing

Experiments are conducted on the three publicly available skin disease datasets to evaluate the effectiveness of the proposed method. The details of each dataset and the corresponding preprocessing setup are given below.

**ISIC2018.** The ISIC2018 dataset [48] includes 10,015 dermoscopic images of 7 skin disease categories, each with an original size of  $600 \times 450$  pixels. As prior work [1, 8], the 4 categories with more samples form the base dataset, while the 3 categories with fewer samples form the novel dataset, as shown in Fig. 1b.

**Derm7pt.** The Derm7pt dataset [49] includes over 2000 skin disease images across 20 categories, each having an original size of  $768 \times 512$  pixels. Following previous work [1, 14], the miscellaneous and melanoma categories are excluded. As shown in Fig. 1a, out of the remaining

**Table 2** Average accuracy on 400 few-shot classification tasks for ISIC2018 dataset with 3-way setting. Bold is best/second

Method	Backbone	3-shot	5-shot	10-shot
MetaMed [9]	Conv4	58.50	61.25	71.00
Transfer [9]	Conv4	55.67	59.67	65.92
PT-MAP [45]	WRN28-10	53.17	55.61	59.57
Baseline [46]	WRN28-10	56.80	59.20	65.22
PFEMed [47]	WRN28-10	<b>66.94</b>	<b>69.78</b>	<b>73.81</b>
ST-Meta [8]	Conv4	59.79	64.59	–
SS-DCN (Ours)	Conv4	<b>66.34</b>	<b>70.69</b>	<b>74.79</b>

**Table 1** Performance comparison of AUC and Accuracy on the ISIC2018 dataset with 2-way setting. Bold is best/second

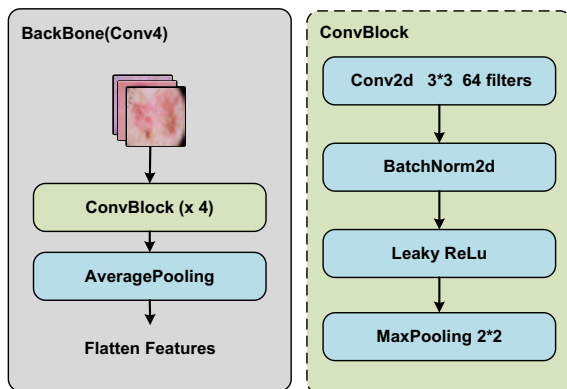
Method	Backbone	2-way 1-shot		2-way 3-shot		2-way 5-shot	
		Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc
Reptile [44]	Conv6	60.3	58.0	73.1	73.4	79.6	76.2
ProtoNet [7]	Conv6	61.6	59.3	70.2	67.9	75.4	73.0
Meta-Derm [1]	Conv6	68.1	64.3	<b>81.2</b>	<b>76.7</b>	<b>86.8</b>	<b>82.1</b>
Meta-Rep [16]	ResNet50	<b>72.6</b>	<b>65.9</b>	77.4	76.5	80.1	79.6
MAML [6]	Conv4	61.74	60.16	73.38	74.56	78.29	79.16
ST-Meta [8]	Conv4	65.27	65.78	76.32	76.38	80.59	81.38
SS-DCN (Ours)	Conv4	<b>75.77</b>	<b>68.45</b>	<b>85.94</b>	<b>79.22</b>	<b>88.88</b>	<b>82.63</b>

**Table 3** Performance comparison of AUC and Accuracy on the Derm7pt dataset with 2-way setting. Bold is best/second

Method	Backbone	2-way 1-shot		2-way 3-shot		2-way 5-shot	
		Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc
Reptile [44]	Conv6	59.7	60.2	64.1	65.7	71.4	70.5
ProtoNet [7]	Conv6	60.6	62.5	65.8	63.9	68.2	66.7
Meta-Derm [1]	Conv6	62.1	61.8	68.7	69.9	77.2	76.9
Meta-Rep [16]	ResNet50	<b>72.9</b>	<b>64.0</b>	<b>78.6</b>	<b>74.3</b>	<b>83.2</b>	<b>78.1</b>
PCN [11]	Conv4	–	59.98	–	–	–	70.62
SCAN [14]	Conv4	–	61.42	–	–	–	72.58
SS-DCN(Ours)	Conv4	<b>74.89</b>	<b>67.32</b>	<b>86.83</b>	<b>78.13</b>	<b>92.87</b>	<b>84.60</b>

**Table 4** Performance comparison of AUC and Accuracy on the SD-198 dataset for 2-way classification tasks. Bold is best/second

Method	Backbone	2-way 1-shot		2-way 3-shot		2-way 5-shot	
		Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc
Reptile [44]	Conv6	64.1	63.0	77.4	72.9	84.6	80.4
ProtoNet [7]	Conv6	59.4	59.8	70.6	66.6	80.7	78.3
Meta-Derm [1]	Conv6	<b>68.6</b>	65.3	79.1	75.8	<b>89.5</b>	83.7
IPNet [12]	Conv6	–	–	<b>83.00</b>	<b>78.41</b>	87.00	84.20
Baseline [46]	Conv4	–	73.98	–	–	–	88.67
SCAN [14]	Conv4	–	<b>77.12</b>	–	–	–	<b>90.22</b>
SS-DCN(Ours)	Conv4	<b>85.63</b>	<b>78.52</b>	<b>93.69</b>	<b>87.53</b>	<b>95.81</b>	<b>90.43</b>

**Fig. 4** Structure of Backbone(Conv4)

18 categories, 13 with more samples form the base dataset, while the 5 categories with fewer samples form the novel dataset.

**SD-198.** The SD-198 dataset [50] includes 6,584 images from 198 fine-grained skin disease categories, each with an original size of  $1640 \times 1130$  pixels. As illustrated in Fig. 1c, 128 classes with more samples are selected as the base dataset, while the remaining 70 classes with fewer samples are selected as the novel dataset.

**Preprocessing.** All input images are uniformly resized to  $80 \times 80$ . Standard data augmentation techniques, i.e., scaling

and horizontal flipping, are utilized for the supervised classification and rotation prediction branch. Meanwhile, the data augmentation techniques recommended in previous work [36, 38] are applied for the contrastive learning branch, enabling the creation of different views of the same sample.

## 4.2 Implementations

**Networks.** In this paper, Conv4 is used as the backbone, comprising four convolutional blocks, each with 64 channels. As shown in Fig. 4, each convolutional block consists of a  $3 \times 3$  convolutional layer, a Batch-Norm layer, a Leaky-Relu layer, and a Max-pooling layer. The output feature maps undergo global average pooling and are subsequently flattened, yielding a 64-dimensional feature embedding. The image classifier  $C(\cdot|W_{class})$  is implemented by a linear layer. The rotation classifier  $C(\cdot|W_{rotation})$  is implemented by a three-layer MLP. The contrastive learning branch follows the structure proposed in [38].

**Training and evaluation.** During the pre-training, Adam [51] is used as the optimizer, the initial learning rate is set to 0.001, and the weight decay is set to  $1e-4$ . The weighting factor  $w$  is set to 0.5. The model is trained for 100 epochs with a batch size of 32. To make a fair comparison, following [1, 14], 600 tasks are randomly sampled from the  $D_{novel}$  for testing. The average accuracy and average macro-AUC are calculated to evaluate the model's classification

performance. The code is implemented in PyTorch [52] and runs on a server with two GeForce RTX 2080 Ti GPUs.

### 4.3 Results

#### 4.3.1 Results on ISIC2018

Performance comparisons for 2-way classification tasks on the ISIC2018 dataset are detailed in Table 1. Our SS-DCN surpasses current state-of-the-art methods, showing significant improvements in AUC and accuracy. Specifically, for 2-way 1-shot, 3-shot, and 5-shot tasks, the AUC and accuracy increased by 3.17%/2.55%, 4.74%/2.52% and 2.08%/0.53%, respectively. Furthermore, Table 2 provides results for the more challenging 3-way setting. To ensure fairness, following previous work [9, 47], 400 tasks are randomly sampled from the  $D_{novel}$  for testing, and accuracy is used as the performance metric. Compared to PEFMed, which had the best results so far, SS-DCN's accuracy improves by 0.91% and 0.98% for 5-shot and 10-shot tasks, respectively, and falls behind by 0.6% for 1-shot tasks. However, the backbone used by PEFMed is far more complex than SS-DCN.

#### 4.3.2 Results on Derm7pt

The performance comparisons on the Derm7pt dataset for 2-way classification tasks are shown in Table 3. The

proposed SS-DCN achieved the best results in all tasks. In the 2-way 1-shot, 3-shot, and 5-shot settings, the AUC and accuracy increased by 1.99%/3.32%, 8.23%/3.83% and 9.67%/6.50%, respectively. We attribute this improvement to the small scale of the Derm7pt dataset, which makes deep models prone to overfitting. SS-DCN effectively mitigates this issue by introducing self-supervised tasks as regularizers. Compared to SCAN, SS-DCN achieves higher accuracy for 2-way 1-shot and 5-shot tasks, with improvements of 5.9% and 12.02%, respectively. SCAN improves performance by learning the sub-cluster structure in the dataset, which is not apparent for Derm7pt.

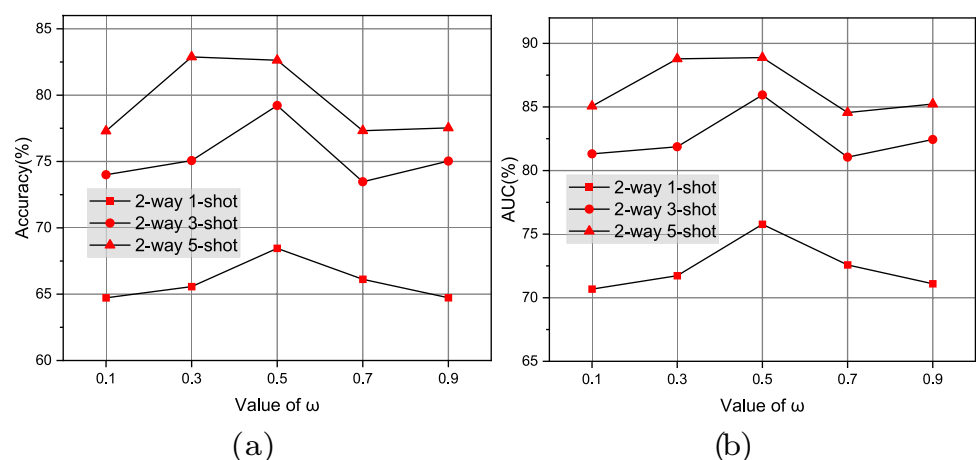
#### 4.3.3 Results on SD-198

The performance comparisons on the SD-198 skin disease dataset for 2-way classification tasks are shown in Table 4. It can be seen that SS-DCN improved AUC and accuracy over the previous best results. Specifically, for the 2-way 1-shot, 3-shot, and 5-shot tasks, the AUC and accuracy increased by 17.03%/1.4%, 10.69%/9.12%, and 6.31%/0.21%, respectively. Moreover, SS-DCN achieves higher accuracy than SCAN, with improvements of 1.4% and 0.21% for 2-way 1-shot and 5-shot tasks, respectively. As mentioned earlier, SCAN focuses primarily on exploring the sub-clustering structure of the dataset and thus relies heavily on the dataset's characteristics. In contrast, SS-DCN is designed to be more versatile and applicable across various datasets.

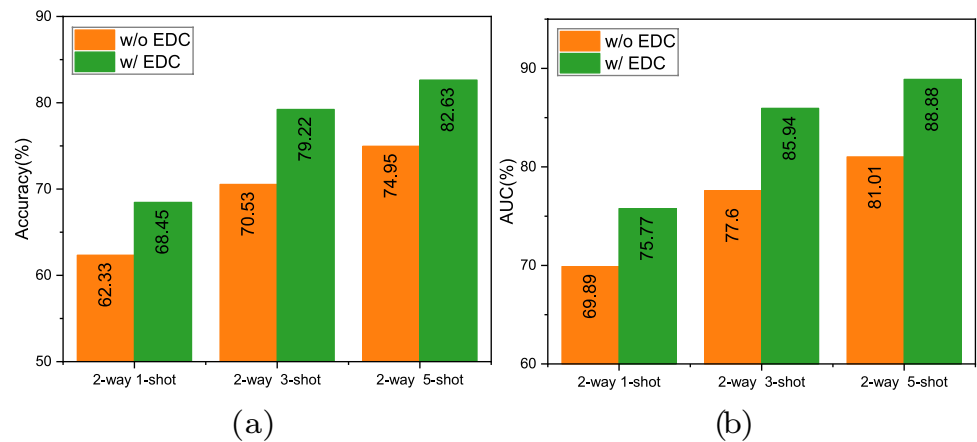
**Table 5** Ablation study on the effectiveness of three pretext tasks on the ISIC2018 dataset. Bold is best

$L_{supervised}$	$L_{rotation}$	$L_{contrastive}$	2-way 1-shot		2-way 3-shot		2-way 5-shot	
			Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc	Avg. AUC	Avg. Acc
✓	✗	✗	69.21	64.18	77.83	71.02	81.01	74.58
✓	✓	✗	74.17	68.07	83.03	76.80	84.47	77.95
✓	✗	✓	70.84	65.22	79.77	73.52	84.16	77.18
✓	✓	✓	<b>75.77</b>	<b>68.45</b>	<b>85.94</b>	<b>79.22</b>	<b>88.88</b>	<b>82.63</b>

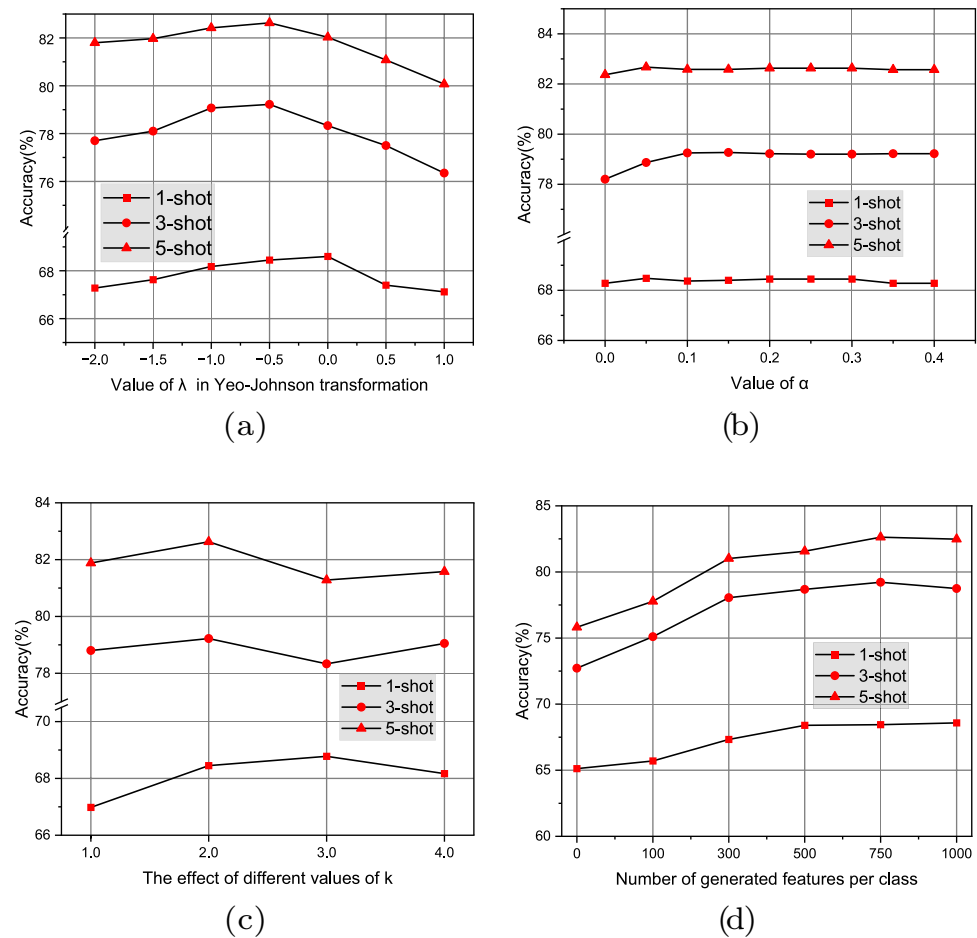
**Fig. 5** Ablation study of weighting factor  $w$  on the ISIC2018 dataset



**Fig. 6** Ablation study on the effectiveness of the EDC on the ISIC2018 dataset



**Fig. 7** Ablation study on the hyperparameters selection of EDC on the ISIC2018 dataset

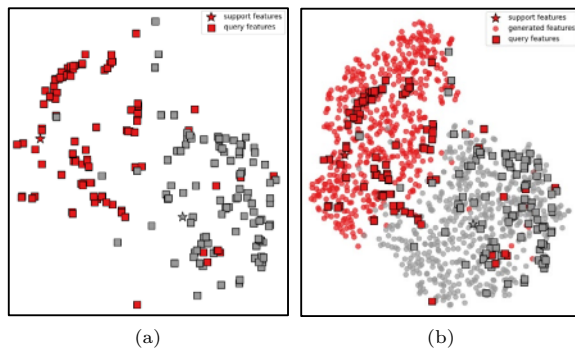


#### 4.4 Ablation study

This section presents comprehensive ablation experiments conducted on the ISIC2018 dataset. The experiments cover various aspects, including hyperparameter selection, verification of module effectiveness, and feature visualization.

##### 4.4.1 Effectiveness of pretext task

To verify the effectiveness of each pretext task, three cases are set up: (1) Use only the supervised branch, which also serves as the baseline; (2) using the supervised and rotation prediction branch; (3) using the supervised, rotation prediction, and the contrastive branch, i.e., SS-DCN. Table 5 summarizes the results for these three cases. The introduction



**Fig. 8** t-SNE visualization of features on the ISIC2018 dataset. Different colors represent different classes. Stars represent support set features, squares represent query set features, and circles represent generated features

of the contrast learning branch enhances the classification performance, indicating that it helps to improve the model's generalizability. Introducing the rotation prediction branch leads to a more significant improvement in model performance. This aligns with previous research [1], which suggests that orientation is not a primary feature in skin disease classification tasks. By enhancing the learning of rotation invariance, the model's performance can be effectively improved. Furthermore, the model that combines supervision, rotation prediction, and contrastive branches performs the best, demonstrating their complementary nature. A predefined weighting factor is introduced, denoted as  $w$ , which influences the classification performance. Experimental results in Fig. 5 show that SS-DCN achieves the best performance when  $w$  is set to 0.5.

#### 4.4.2 Effectiveness of EDC

Two cases are set up to verify the effectiveness of EDC: (1) with EDC and (2) without EDC. As shown in Fig. 6, EDC positively impacts performance. Additionally, detailed ablation experiments are conducted on selecting hyperparameters for this module. Figure 7a shows the impact of the value of  $\lambda$  in Yeo-Johnson transformation on accuracy. This paper sets  $\lambda$  to  $-0.5$ , which achieves better performance in various tasks. Figure 7b shows that  $\alpha$  has little impact on the proposed method, and accuracy tends to be stable after  $\alpha$  is greater than 0.1. Figure 7c shows the effect of different values of  $k$ , which represents the number of base class statistics retrieved. When  $k$  is set to 2, better performance is achieved in various tasks. Figure 7d shows the impact of the number of generated features on performance. The model can benefit from more generated features when the number is below 750. However, when the sampled features further increase, the performance of the model begins to decrease.

## 4.5 Visualizing the generated features

t-SNE [53] is utilized to visualize the generated features sampled from the calibrated distribution. Figure 8a displays the features of support and query set, while Fig. 8b exhibits the features of support set, query set, and generated samples. Due to the limited number of samples in the support set, the samples in the query set tend to cover a larger area, leading to a less obvious decision boundary. SS-DCN addresses this issue by generating features. Specifically, the generated features can overlap with the distribution of the query set, and incorporating these generated features into classifier training can result in a better decision boundary.

## 5 Conclusion

This paper proposes a self-supervision distribution calibration network (SS-DCN) to enhance the accuracy of rare skin disease diagnosis in the few-shot setting. Specifically, a multi-task learning framework is used during pre-training to help the model learn more discriminative and transferable visual representations. Additionally, an enhanced distribution calibration (EDC) strategy is proposed to calibrate the biased distribution of novel classes, and more samples are generated to improve the discrimination of the decision boundary. Compared with existing works, SS-DCN achieves advanced performance.

**Funding** This work was supported by the Science and Technology Cooperation Project between Jilin Province and the Chinese Academy of Sciences (2022SYHZ0030).

## Declarations

**Conflict of interest** The authors have not disclosed any Conflict of interest.

**Ethical approval** The study utilized publicly available datasets, and therefore, ethical review and approval were not required in accordance with the local legislation and institutional requirements.

## References

1. Mahajan K, Sharma M, Vig L. Meta-dermdiagnosis: few-shot skin disease identification using meta-learning. In: 2020 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW) 2020; pp. 3142–3151.
2. Grignaffini F, et al. Machine learning approaches for skin cancer classification from dermoscopic images: a systematic review. *Algorithms*. 2022;15:438.

3. Hosny KM, Kassem MA. Refined residual deep convolutional network for skin lesion classification. *J Digit Imaging*. 2022;35:258–80.
4. Alsahafi YS, Kassem MA, Hosny KM. Skin-net: a novel deep residual network for skin lesions classification using multilevel feature extraction and cross-channel correlation with detection of outlier. *J Big Data*. 2023;10:1–23.
5. Hosny KM, Said W, Elmezain M, et al. Explainable deep inherent learning for multi-classes skin lesion classification[J]. *Appl Soft Comput*. 2024;111624.
6. Finn C, Abbeel P, Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* 2017.
7. Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[J]. *Adv Neural Inform Process Syst* 2017;30.
8. Zhang D, Jin M, Cao P. St-metadiagnosis: meta learning with spatial transform for rare skin disease diagnosis. In *2020 IEEE international conference on bioinformatics and biomedicine (BIBM)* 2020;2153–2160.
9. Singh R, et al. Metamed: few-shot medical image classification using gradient-based meta-learning. *Pattern Recognit*. 2021;120: 108111.
10. Li X, et al. Difficulty-aware meta-learning for rare disease diagnosis. In *Medical image computing and computer assisted intervention—MICCAI 2020: 23rd international conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I* 23 2020;357–366.
11. Prabhu V, Kannan A, Ravuri M, et al. Few-shot learning for dermatological disease diagnosis[C]//*Machine Learning for Healthcare Conference*. PMLR, 2019: 532–552.
12. Chowdhury RR, Bathula DR. Influential prototypical networks for few shot learning: a dermatological case study. In *2022 IEEE 19th international symposium on biomedical imaging (ISBI)* 2021;1–4.
13. Zhu W, Li W, Liao H, Luo J. Temperature network for few-shot learning with distribution-aware large-margin metric. *Pattern Recognit*. 2021;112: 107797.
14. Li S, Li X, Xu X, et al. Dynamic Subcluster-Aware Network for Few-Shot Skin Disease Classification[J]. *IEEE Trans on Neural Net Learn Syst*. 2023.
15. Cai A, et al. Pre-mocodiagnosis: few-shot ophthalmic diseases recognition using contrastive learning. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)* 2022;2059–2066.
16. Desingu K, Mirunalini P, Chandrabose A. Few-shot classification of skin lesions from dermoscopic images by meta-learning representative embeddings. 2022 **abs/2210.16954**.
17. Wang W, Li Y, Lu K, et al. Medical tumor image classification based on Few-shot learning[J]. *IEEE/ACM Trans Comput Biol Bioinf*. 2023.
18. Li P, et al. Knowledge transduction for cross-domain few-shot learning. *Pattern Recognit*. 2023;141: 109652.
19. Zhang C, Gu, Y. Dive into self-supervised learning for medical image analysis: data, models and tasks. **abs/2209.12157** 2022.
20. Yang S, Liu L, Xu M. Free lunch for few-shot learning: distribution calibration. **abs/2101.06395** 2021.
21. Parnami A, Lee M. Learning from few examples: a summary of approaches to few-shot learning. **abs/2203.04291** 2022.
22. Sung F, et al. Learning to compare: relation network for few-shot learning. In *2018 IEEE/CVF conference on computer vision and pattern recognition* 2017;1199–1208.
23. Rusu AA, et al. Meta-learning with latent embedding optimization. **abs/1807.05960** 2018.
24. Li Z, Zhou F, Chen F, Li H. Meta-sgd: learning to learn quickly for few shot learning. **abs/1707.09835** 2017.
25. Wang Y-X, Girshick RB, Hebert M, Hariharan B. Low-shot learning from imaginary data. In *2018 IEEE/CVF conference on computer vision and pattern recognition* 2018;7278–7286.
26. Hariharan B, Girshick RB. Low-shot visual recognition by shrinking and hallucinating features. In *2017 IEEE international conference on computer vision (ICCV)* 2016;3037–3046.
27. Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P. Rethinking few-shot image classification: a good embedding is all you need? In *European conference on computer vision* 2020.
28. Wang Y, Chao W-L, Weinberger KQ, van der Maaten L. Simple-shot: revisiting nearest-neighbor classification for few-shot learning. **abs/1911.04623** 2019.
29. Hu Y, Liu R, Li X, Chen D, Hu Q. Task-sequencing meta learning for intelligent few-shot fault diagnosis with limited data. *IEEE Trans Industr Inf*. 2022;18:3894–904.
30. Khadka R, et al. Meta-learning with implicit gradients in a few-shot setting for medical image segmentation. *Comput Biol Med*. 2021;143: 105227.
31. Zhu W, Liao H, Li W, Li W, Luo J. Alleviating the incompatibility between cross entropy loss and episode training for few-shot skin disease classification. In *International conference on medical image computing and computer-assisted intervention* 2020.
32. Zhou C, Sun M, Chen L, Cai A, Fang J. Few-shot learning framework based on adaptive subspace for skin disease classification. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)* 2022;2231–2237.
33. Gui J, et al. A survey of self-supervised learning from multiple perspectives: algorithms, theory, applications and future trends. **abs/2301.05712** 2023.
34. Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations. **abs/1803.07728** 2018.
35. Doersch C, Gupta AK, Efros AA. Unsupervised visual representation learning by context prediction. In *2015 IEEE international conference on computer vision (ICCV)* 2015;1422–1430.
36. Chen X, Fan H, Girshick RB, He K. Improved baselines with momentum contrastive learning. **abs/2003.04297** 2020.
37. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning* 2020;1597–1607.
38. Chen X, He K. Exploring simple siamese representation learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* 2020;15745–15753.
39. Zbontar J, Jing L, Misra I, LeCun Y, Deny S. Barlow twins: self-supervised learning via redundancy reduction. In *International conference on machine learning* 2021;12310–12320.
40. Doersch C, Zisserman A. Multi-task self-supervised visual learning. In *2017 IEEE international conference on computer vision (ICCV)* 2017;2070–2079.
41. Simard N, Lagrange G. Improving few-shot learning with auxiliary self-supervised pretext tasks. **abs/2101.09825** 2021.
42. Kwon YI, Johnson RA. A new family of power transformations to improve normality or symmetry. *Biometrika*. 2000;87:954–9.
43. Box GEP, Cox DR. An analysis of transformations. *J R Stat Soc Ser B-Methodol*. 1964;26:211–43.
44. Nichol A, Schulman J. Reptile: a scalable metalearning algorithm. [arXiv: Learning](https://arxiv.org/abs/1803.03300) 2018.
45. Hu Y, Gripon V, Pateux S. Leveraging the feature distribution in transfer-based few-shot learning. In *International conference on artificial neural networks* 2020.
46. Chen W-Y, Liu Y-C, Kira Z, Wang YCF, Huang J-B. A closer look at few-shot classification. In *International conference on learning representations* 2019.
47. Dai Z, et al. Pfemed: few-shot medical image classification using prior guided feature enhancement. *Pattern Recognit*. 2023;134: 109108.
48. Codella NCF, et al. Skin lesion analysis toward melanoma detection 2018: a challenge hosted by the international skin imaging collaboration (isic). **abs/1902.03368** 2019.

49. Kawahara J, Daneshvar S, Argenziano G, Hamarneh G. Seven-point checklist and skin lesion classification using multi-task multimodal neural nets. *IEEE J Biomed Health Inform.* 2019;23:538–46.
50. Sun X, Yang J, Sun M, Wang K. A benchmark for automatic visual classification of clinical skin disease images. In *European conference on computer vision* 2016.
51. Kingma DP, Ba J. Adam: a method for stochastic optimization. **abs/1412.6980** 2014.
52. Paszke A, et al. Pytorch: an imperative style, high-performance deep learning library. In *Neural information processing systems* 2019.
53. van der Maaten L, Hinton GE. Visualizing data using t-sne. *J Mach Learn Res.* 2008;9:2579–605.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.