Applied Intelligence (2023) 53:20661–20674
https://doi.org/10.1007/s10489-023-04525-4

Self-supervised pairwise-sample resistance model for few-shot classification

Weigang Li[1,2] · **Lu Xie**[2] · **Ping Gan**[2,3] · **Yuntao Zhao**[2]

Accepted: 10 February 2023 / Published online: 19 April 2023

Abstract
The traditional supervised learning models rely on high-quality labeled samples heavily. In many fields, training the model on limited labeled samples will result in a weak generalization ability of the model. To address this problem, we propose a novel few-shot image classification method by self-supervised and metric learning, which contains two training steps: (1) Training the feature extractor and projection head with strong representational ability by self-supervised technology; (2) taking the trained feature extractor and projection head as the initialization meta-learning model, and fine-tuning the meta-learning model by the proposed loss functions. Specifically, we construct the pairwise-sample meta loss (ML) to consider the influence of each sample on the target sample in the feature space, and propose a novel regularization technique named resistance regularization based on pairwise-samples which is utilized as an auxiliary loss in the meta-learning model. The model performance is evaluated on the 5-way 1-shot and 5-way 5-shot classification tasks of mini-ImageNet and tired-ImageNet. The results demonstrate that the proposed method achieves the state-of-the-art performance.

Keywords Few-shot · Self-supervised · Meta-learning · Pairwise-sample · Regularization

1 Introduction

The supervised learning methods rely on a large number of manually labeled samples. In many fields, the lack of labeled samples limits the reliability and generalization ability of the model. Few-shot learning is proposed to address this problem, which aims to enable the model to classify the new classes that do not appear in the training set with only a few annotations [1].

The essence of few-shot learning tasks are to solve cross-domain problems. Chen et al. [2] find that the meta-learning methods lose their advantages when the domain difference is too large. The same view is also put forward in [3], Guo et al. find that there is a little difference in the performance of different meta-learning models in the same domain, but the performance of one meta-learning model in different domains is significantly different. This phenomenon is called a cross-domain problem [4]. At present, many researchers have discovered that learning an excellent feature encoder can greatly improve the model performance [5]. Therefore, more and more researchers focus on learning an initialization feature encoder with strong generalization ability to solve the cross-domain problem.

In this work, we propose a novel few-shot classification framework based on metric learning and self-supervised learning, which consists of a classification model and a meta-learning model. Specifically, the classification model is trained on the base class set to obtain the feature extractor with strong extraction ability, and then this feature extractor is utilized as the initial feature encoder of the meta-learning

✉ Ping Gan
pinggan.gan@qunar.com

Weigang Li
liweigang.luck@foxmail.com

Lu Xie
1468596295@qq.com

Yuntao Zhao
zhyt@wust.edu.cn

[1] Engineering Research Center of Metallurgical Automation and Measurement Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan, 430081, Hubei, China

[2] College of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, 430081, Hubei, China

[3] Qunar, Beijing, 100080, Beijing, China

model to evaluate on the new classes set [6, 7]. The contributions are as follows:

- We add the rotation self-supervised auxiliary loss into the classification network, which aims to improve the feature representation ability of the model.
- We propose the meta loss (ML) based on pairwise-samples, which aims to reduce the intra-class difference and increase the inter-class difference.
- We propose a new regularization technique named resistance regularization, which could improve the generalization ability of the model. The resistance regularization includes the exchange processing and NT-Xent (normalized temperature-scaled cross entropy) loss.

2 Related work

2.1 Meta-learning

Meta-learning could generalize the previously learned knowledge or experience to many new tasks autonomously and quickly [8]. For example, the meta transfer learning (MTL) is proposed in [6], which combines the hard task (HT) meta-batch scheme to force the meta-learner to "grow up in difficulties". Task-aware feature embedding network (TAFE-Net) is proposed in [9] to obtain the task aware embedding for few-shot classification tasks. Latent embedding optimization (LEO) is introduced in [10], which applies a parameter generation model to capture useful parameters for the tasks. Chen et al. [2] propose that the performance of the model is related to the domain difference, and the performance of the shallow network is better than that of other deep backbones when the domain difference is small. In addition, a new meta-learning method is proposed by dual formulation and KKT conditions in [11] to improve the computational efficiency.

2.2 Metric learning

Metric learning aims to reduce the intra-class difference and increase the inter-class difference, it is widely used in many fileds. Recently, the deep metric learning losses are built on pairwise-samples. For example, a novel hierarchical triplet loss (HTL) is proposed to automatically collect informative training samples in [12]. Riplet center loss is proposed in [13], which could further enhance the distinctiveness of features. Multi-class n-pair loss is proposed in [14] to solve the slow convergence of the contrastive loss and triplet loss. A new angle loss is proposed in [15], which aims to learn valuable features by considering the angle relationship of samples. Wu et al. [16] put forward that the selection of training samples plays an equally important role in the training of the model, and propose a novel sampling method by distance weighting. In our work, we propose the meta loss (ML) based on the pairwise-samples, which is utilized in the meta-learning model to consider the influence of other samples on the target sample in the feature space.

Classification by metric learning is performed in two steps. First, the eigenvector centroid β_i of the class i is calculated by formula (1), where K is the number of input samples; secondly, calculating the similarity score p_i between the predicted sample x and the centroid in formula (2). The category with the highest similarity score is regarded as the category of the target sample x.

$$\beta_i = \frac{1}{K}\sum_{j=1}^{K} f_\theta(x_{i,j}) \tag{1}$$

$$p_i = < f_\theta(x), \beta_i > = \frac{f_\theta(x)\cdot\beta_i}{\|f_\theta(x)\|\|\beta_i\|} \tag{2}$$

2.3 Self-supervised learning

Self-supervised learning aims to mine the supervision information from large-scale unlabeled data by many auxiliary tasks, which could help the model capture more valuable features. Doersch et al. [17] construct a new auxiliary loss by predicting the context of the image, and the same work has been carried out in [18] and [19]. The auxiliary task of predicting the color of an image is designed in [20, 21], which is utilized to extract the semantic information. Gidaris et al. [22] propose a rotation loss to predict the rotation angle of the image, which could improve the robustness of the model. Hjelm et al. [23] design an auxiliary task to distinguish between the global feature and local feature of the image. Tian et al. [24] propose to construct samples by multi-perspective information. Chen et al. propose SimCLR in [25], it designs the auxiliary tasks by augmenting the input samples. At present, many researchers [26, 27] put forward to combine these self-supervised auxiliary tasks into the classification networks, which could greatly improve the performance of the models. In view of the above mentioned, the rotation self-supervised loss is applied in our classification network to obtain an initialized feature encoder with strong representational ability.

3 Method

3.1 The overall framework

The framework of our self-supervised pairwise-sample resistance model (SPRM) is shown in Fig. 1, which consists of a classification model and a meta-learning model. The

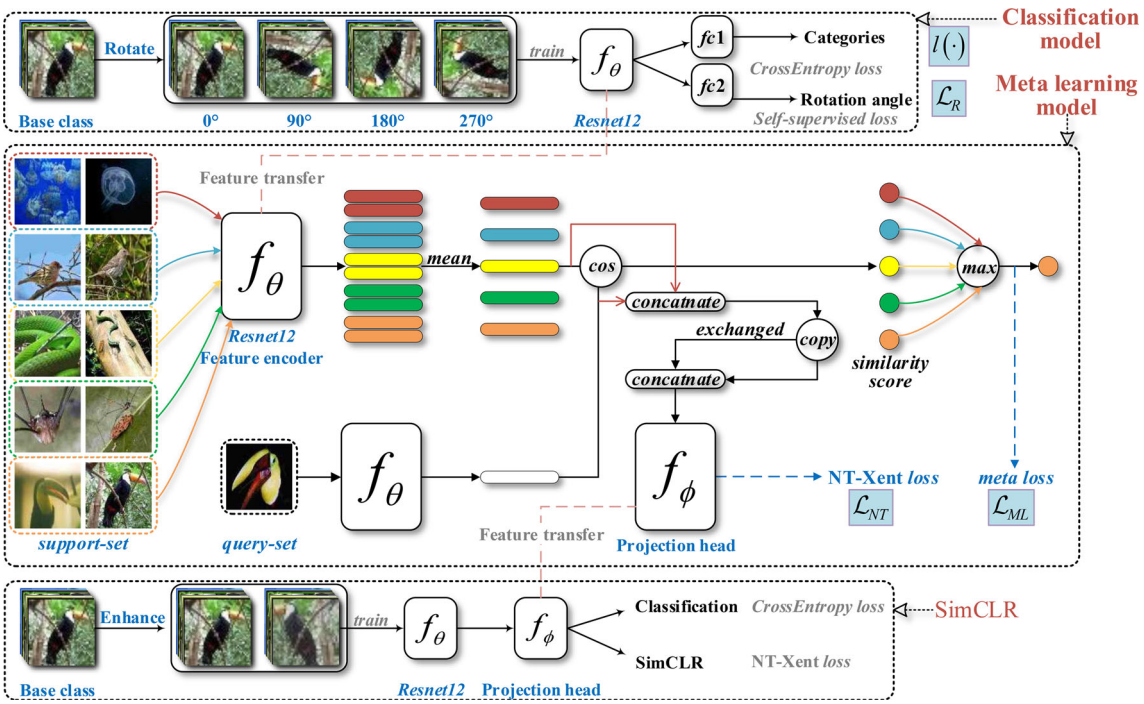

Fig. 1 The framework of SPRM

f_{c1} and f_{c2} are the two classification layers, f_θ is the feature encoder, and f_ϕ is the projection head. $l(\cdot)$ is the cross-entropy loss, and the rotation self-supervised loss ($\mathcal{L}_R$) is added to the classification model as an auxiliary loss, and the trained feature encoder of the classification model is used as the initial feature encoder of the meta-learning model; the meta loss ($\mathcal{L}_{ML}$) is proposed based on the pairwise-samples, and it is applied in the meta-learning model; inspired by SimCLR, a new regularization technique called resistance regularization ($\mathcal{L}_{NT}$) is proposed for few-shot learning, and it is applied into the ML as a regularization term.

The algorithm flow of SPRM is shown as Algorithm 1, it is divided into two training steps as follows:

Step 1: Training the classification model by $l(\cdot) + \mathcal{L}_R$; training the projection head by $l(\cdot) + \mathcal{L}_{NT}$.

Step 2: Training the meta-learning model by $\mathcal{L}_{ML} + \mathcal{L}_{NT}$. The resistance regularization $\mathcal{L}_{NT}$ is utilized as the regularization term in the meta loss $\mathcal{L}_{ML}$.

3.2 Rotation self-supervised loss

The rotation self-supervised loss [22] is utilized to increase the feature extraction ability and robustness of the classification model. Specifically, rotating the input image at four angles of 0°, 90°, 180° and 270°, so the four images can be obtained by one image. The task of the model is to predict the rotation angles of these rotated images. The self-supervised loss function is defined as:

$$\mathcal{L}_R = \min_\theta \frac{1}{MC} \sum_{i=1}^{M} \sum_{j=1}^{C} l(f_\theta(x_{i,j}), y) \tag{3}$$

where M is the number of input samples and C is the number of the rotation angles, and C is 4 in our network. $x_{i,j}$ represents the i-th sample with the rotation angle j.

3.3 Meta loss

Wang et al. [28] propose the MS loss combined with the self-similarity S, positive relative similarity P and negative relative similarity N, the calculation is shown as:

$$\mathcal{L}_{MS} = \frac{1}{M} \sum_{i=1}^{M} \left\{ \frac{1}{\alpha} \log \left[1 + \sum_{k \in \mathcal{P}} \exp^{-\alpha(S_{ik} - \lambda)} \right] + \frac{1}{\beta} \log \left[1 + \sum_{k \in \mathcal{N}} \exp^{\beta(S_{ik} - \lambda)} \right] \right\} \tag{4}$$

$$S_{ik} = < x_i, x_k > = \frac{x_i \cdot x_k}{\|x_i\| \|x_k\|} \tag{5}$$

where $\mathcal{P}$ and $\mathcal{N}$ are the positive and negative samples respectively, x_i is the anchor, x_k is the k-th sample which needs to be predicted, λ is the similarity threshold, and α and β are the hyperparameters, which are set by experience. α controls the compactness of the positive samples and penalizes the positive samples whose cosine similarity is less than λ; β controls the compactness of negative samples and penalizes the positive samples whose cosine similarity is greater than λ.

Input: $\{x, y\} \in \mathbb{B}$
Output: Backbone model f_θ, projection head modelv f_ϕ

begin
  ▷ Classification model feature extractor backbone f_θ and projection head f_ϕ training
  Initialize f_θ, f_ϕ
  for $epochs \in \{1, 2, ..., n\}$ **do**
    while $\mathcal{B} \sim \mathbb{B}$ **do**
      rotate $(x) \rightarrow$
      $x^{90^o}, x^{180^o}, x^{270^o}$, augment $(x) \rightarrow x^o$
      $\left\{x, x^{90^o}, x^{180^o}, x^{270^o}\right\} \rightarrow \hat{x}, \{x, x^o\} \rightarrow \tilde{x}$,
      $\{y, y, y, y\} \rightarrow \hat{y}, \{0, 1, 2, 3\} \rightarrow \hat{r}$
      $\mathcal{L}\left(\theta, \hat{x}, \hat{y}, \hat{r}\right) = \mathcal{L}_R + l(\cdot)$
      $\theta \rightarrow \theta - \eta * \nabla \mathcal{L}\left(\theta, \hat{x}, \hat{y}, \hat{r}\right)$
      $\mathcal{L}(\phi, \tilde{x}) = \mathcal{L}^{NX-Xent} + l(\cdot)$
      $\phi \rightarrow \phi - \xi * \nabla \mathcal{L}(\phi, \tilde{x})$
    end
  end
  ▷ Meta-learning model feature extractor backbone f_θ and projection head f_ϕ fine-tuning
  for $epochs \in \{1, 2, ..., m\}$ **do**
    for $task \in \{1, 2, ..., h\}$ **do**
      copy $(x) \rightarrow x^c$
      exchange $(x^c) \rightarrow x^c$
      $\{x, x^c\} \rightarrow \bar{x}$
      $\mathcal{L}(\theta, \phi, x, y, \bar{x}) =$
      $\mathcal{L}_{ML}(\theta, x, y) + \mathcal{L}^{NX-Xent}(\theta, \phi, \bar{x})$
      $(\theta, \phi) \rightarrow (\theta, \phi) - \eta * \mathcal{L}(\theta, \phi, x, y, \bar{x})$
    end
  end
  return fine-tuned backbone f_θ
end

Algorithm 1 *SPRM* feature backbone training.

Meta-learning aims to predict the query set samples by using the support set samples. However, MS loss sets each sample as the anchor in turn, which will lead to the situation that the samples in the query set are used to predict the other samples, it is incompatible with the meta-learning training paradigm.

According to all the above, we propose the meta loss (ML) for few-shot learning, which only takes the centroid of the support set samples as the anchor. The calculation formula of ML is shown as:

$$\mathcal{L}_{ML} = \frac{1}{N}\sum_{a=1}^{N}\left\{\frac{1}{\alpha}\log\left[\mu + \sum_{k\in\mathcal{P}}\exp^{-\alpha(S_{ak}-\eta)}\right] + \frac{1}{\beta}\log\left[\mu + \sum_{k\in\mathcal{N}}\exp^{\beta(S_{ak}-\lambda)}\right]\right\} \tag{6}$$

where the N is the number of categories, and μ and η are added in ML to constrain the positive and negative pairwise-samples, see Section 3.3.2 for details. It can be seen from formula (4) and formula (6) that the i of MS traverses all samples from 1 to M, while the a of ML traverses from 1 to N. That is, in the N-way tasks, ML only uses the centroid of the support set samples as the anchor, which not only improves the calculation efficiency of the model, but also satisfies the principle that only the support set samples are used as the anchors in the meta-learning model. ML contains two steps: mining and weighting the pairwise-samples, which are shown in Fig. 2.

3.3.1 Mining the pairwise-samples

ML aims to reduce the computational effort of the model by mining more valuable pairwise-samples, which are the negative pairwise-samples (different categories) with large similarity score and the positive pairwise-samples (the same category) with small similarity score.

Inspired by LMNN [29] and MS loss [28], the positive relative similarity P is used to mine the difficult pairwise-samples by formulas 7 and 8, and the other pairwise-samples with less information are discarded.

$$S_{ai}^{-} > \min_{i\notin\mathcal{P}} S_{a\mathcal{P}} - \epsilon \tag{7}$$

$$S_{aj}^{+} < \max_{j\notin\mathcal{N}} S_{a\mathcal{N}} + \epsilon \tag{8}$$

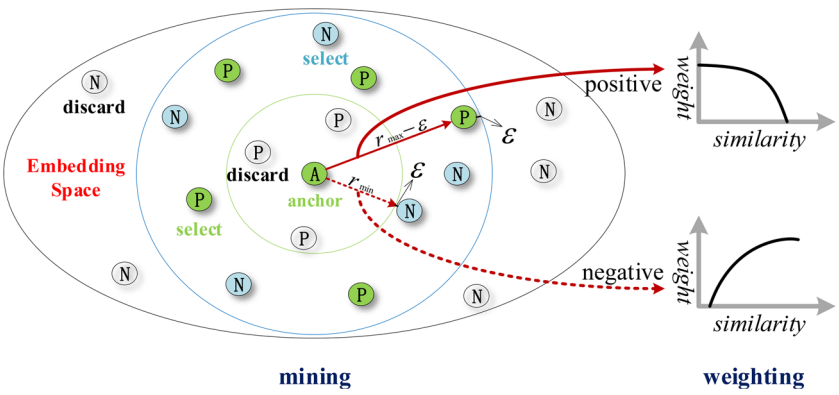

Fig. 2 Mining and weighting of pairwise-samples in ML. Draw a circle by the distance of the negative pairwise-sample which nearest to the anchor, the radius of this circle is $r_{\min}$, where $\min(d_{a\mathcal{N}}) = r_{\min} + \epsilon$; and draw a circle by the distance of the pairwise-sample which farthest from the anchor, the radius of this circle is $r_{\max}$, where $\max(d_{a\mathcal{P}}) = r_{\max} - \epsilon$

Table 1 Mining process of ML

Input: All samples in one task
1. The centroid set of the support set x_a is calculated by formula 1;
2. x_a is paired with all eigenvectors of the query set to obtain the pairwise-sample set $\mathcal{F}$;
3. The cosine similarity map S of each pairwise-sample is obtained by formula 4;
4. Finding the pairwise-sample with the minimum cosine similarity in the positive pairwise-sample set $\mathcal{P}$. $\{a, \mathcal{P}\}_{min}, \mathcal{P} \in \mathcal{F}$;
5. Finding the pairwise-sample with the maximum cosine similarity in the negative pairwise-sample set $\mathcal{N}$. $\{a, \mathcal{N}\}_{max}, \mathcal{N} \in \mathcal{F}$;
6. Pairwise-samples are retained if the cosine similarity S_{ai}^{-} of negative pairwise-samples $\{x_a, x_i\}$ satisfies formula 7;
7. Pairwise-samples are retained if the cosine similarity S_{aj}^{+} of positive pairwise-samples $\{x_a, x_j\}$ satisfies formula 8;
8. Discarding the remaining pairwise-samples;
Output: The set of reserved pairwise-samples

where $S_{a\mathcal{P}}$ and $S_{a\mathcal{N}}$ are the similarity score maps of positive and negative pairwise-samples, respectively, ϵ is the threshold for mining, and S_{ai}^{-} is the cosine similarity range of the mined negative pairwise-samples, S_{aj}^{+} is the cosine similarity range of the mined positive pairwise-samples. The mining process of ML is shown in Table 1.

3.3.2 Weighting the pairwise-samples

The valuable pairwise-samples can be roughly mined by positive relative similarity P, and then these valuable pairwise-samples can be further weighted by self-similarity S and negative relative similarity N. Specifically, given a negative pairwise-sample $\{x_a, x_i\}, i \in \mathcal{N}$, the weight w_{ai} (partial derivative of S_{ai} in formula 7) is calculated by formula 9, which penalizes the negative samples with cosine similarity $> \lambda$, given the positive pairwise-samples $\{x_a, x_j\}, j \in \mathcal{P}$, the weight calculation is shown in formula 10, which penalizes the positive samples with cosine similarity $< \eta$. The parameter μ is utilized to adjust the proportion of self-similarity S.

$$w_{ai}^{-} = \frac{1}{\mu \exp^{\beta(\lambda - S_{ai})} + \sum_{k \in \mathcal{N}} \exp^{\beta(S_{ak} - S_{ai})}} = \frac{\exp^{\beta(S_{ak} - \lambda)}}{\mu + \sum_{k \in \mathcal{N}} \exp^{\beta(S_{ak} - \lambda)}} \tag{9}$$

$$w_{aj}^{+} = \frac{1}{\mu \exp^{-\alpha(\eta - S_{aj})} + \sum_{k \in \mathcal{P}} \exp^{-\alpha(S_{ak} - S_{aj})}} \tag{10}$$

3.4 Resistance regularization

Resistance regularization contains the exchange processing and NT-Xent loss. Before that, we need to construct the pairwise-sample labels.

In a N-way K-shot M-query task, there are N classes, each with $K + M$ images, and a total of $N \times (K + M)$ input images. Coping each image, then the $2N \times (K + M)$ input images will be obtained. Regarding the real label of each image, we suppose the $N \times (K + M)$ original images are the $N \times (K + M)$ different categories. In fact, there are $2 \times (K + M)$ images in each category after copying, but in the pairwise-sample scenario, there are only two images (the original and its copied image) in each category. For example, there is a 3-way 1-shot 1-query meta-learning task, the original input samples are [dog1, dog2, cat1, cat2, pig1, pig2], their coped samples are [Dog1, Dog2, Cat1, Cat2, Pig1, Pig2], so the input data is expanded to [dog1, dog2, cat1, cat2, pig1, pig2, Dog1, Dog2, Cat1, Cat2, Pig1, Pig2] after copying. Pairwise-sample labels are constructed in Fig. 3 (a), we mark the pairwise-sample with the same label as 1, otherwise 0.

3.4.1 Exchange processing

After the pairwise-sample labels are constructed, the labels are fixed and the positions of the $2N \times (K + M)$ input images are exchanged, as shown in Fig. 3 (b). The self-pairwise-sample labels are deleted. After exchanging, for the areas where the pairwise-sample label are not zero, if the real labels of the two samples in one pairwise-sample are the same, it is called soft exchanging (represented as +1), otherwise it is called hard exchanging (represented as -1). "+1" and "-1" are for the convenience of differentiation, and they are regarded as 1 when calculating the NT-Xent loss. In Fig. 3 (b), an example of soft exchanging is the pairwise-sample {dog2, Dog1}; An example of hard exchanging is the pairwise-sample {dog1, Pig2}.

To sum up, both soft and hard exchanging are designed to hinder the further learning of the model. Specifically, taking the pairwise-sample {dog2, Dog1} as an example, the pairwise-sample label of them is "+1" and the real labels of them are the same, which allows the model to narrow the intra-class difference. But for the pairwise-sample {dog2, Dog2}, the pairwise-sample label of them is "0" and the real labels of them are the same, which will prevent the model from learning similar characteristics. Therefore, the soft exchanging will correctly increase the similarity of pairwise-samples {dog2, Dog1}; incorrectly decrease the similarity of pairwise-samples {dog2, Dog2}, {dog2, dog1}. Similarly, taking the pairwise-sample {dog1, Pig2}

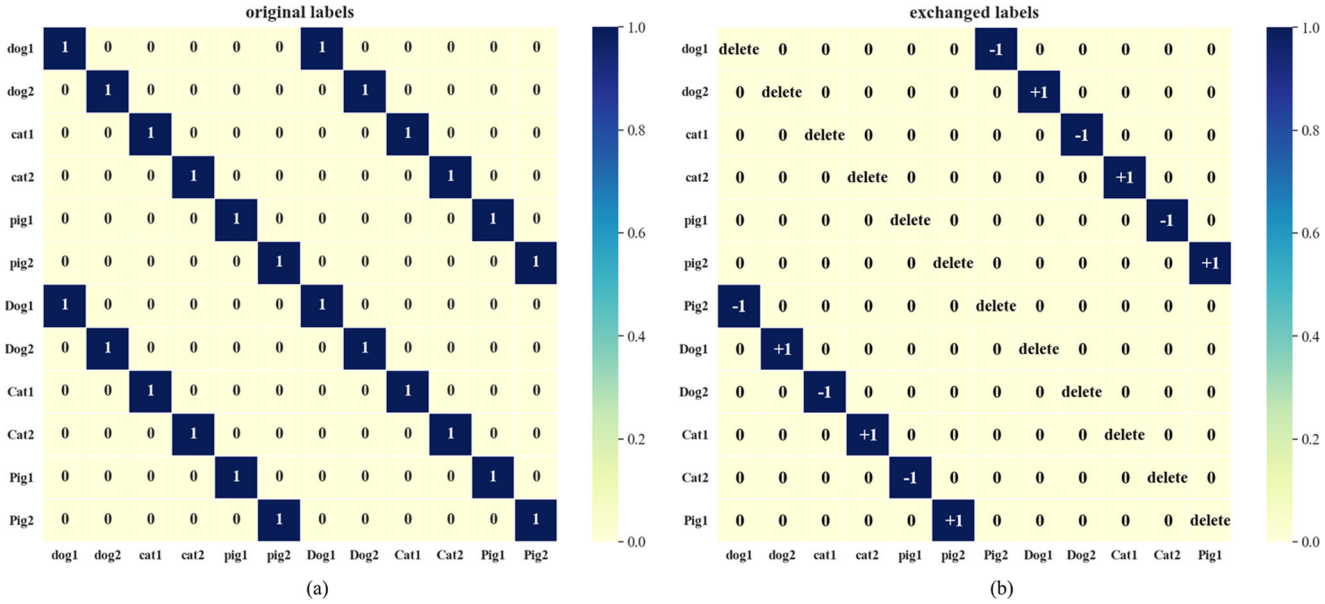

Fig. 3 Pairwise-sample labels for resistance regularization

as an example, hard exchanging will correctly decrease the similarity of the pairwise-samples {pig2, cat1}, and incorrectly increase the similarity of the pairwise-samples {dog1, Pig2}, {cat1, Dog2} and {pig1, Cat2}. Both the soft and hard exchanging could correctly decrease the similarity between the two samples whose the real labels are different, but the hard exchanging has a greater hindrance to the training of the model compared with soft exchanging.

The effects of the proportion of the soft and hard exchanging on the model performance are explored in Table 2 by comparative experiments, where "Soft%" represents the proportion of the soft exchanging.

Considering that the appropriate range of soft exchanging proportion can enhance the generalization ability of the model, the soft exchanging proportion of our method is randomly selected between 52.5% and 86.25%.

Table 2 The experimental results of SPRM with different exchanging proportion on mini-ImageNet

Soft %	1-shot(%)	5-shot(%)
0	64.99 ± 0.34	81.65 ± 0.28
20	65.49 ± 0.34	82.13 ± 0.27
40	65.53 ± 0.34	$\mathbf{82.28 \pm 0.27}$
60	65.49 ± 0.34	81.92 ± 0.28
80	65.46 ± 0.34	82.19 ± 0.27
100	65.39 ± 0.34	81.84 ± 0.28
Random	65.46 ± 0.34	81.91 ± 0.28
Our	$\mathbf{66.35 \pm 0.34}$	82.24 ± 0.27

The best results are shown in bold

3.4.2 NT-Xent loss

NT-Xent loss is proposed in SimCLR, the calculation is shown in formula 11. Z_i and Z_j represent the eigenvectors of an original and its copied image obtained from the feature extractor, respectively; Z_k is the eigenvector of the k-th image ($k \neq i$) obtained from the feature extractor. The NT-Xent loss is added as the auxiliary term in the ML.

$$\mathcal{L}_{NT}^{i,j} = -\frac{1}{2M}\sum_{i=1}^{2M}\sum_{j=1}^{2M} y_{ij} \log \frac{\exp^{sim(Z_i,Z_j)/\tau}}{\sum_{k=1[k\neq i]}^{2M} \exp^{sim(Z_i,Z_k)/\tau}}, i \neq j \tag{11}$$

$$sim(Z_i, Z_j) = \frac{Z_i \cdot Z_j}{\|Z_i\|\|Z_j\|} \tag{12}$$

4 Experiments

4.1 Datasets

Mini-ImageNet mini-ImageNet [1] dataset is composed of 60000 images selected from ImageNet, a total of 100 categories. There are 600 images in each category, and the size of each image is 84×84. It is usually divided into the base class set (64 categories), validation set (16 categories) and new class set (20 categories).

Tiered-ImageNet tiered-ImageNet [30] dataset is also selected from ImageNet. It contains 34 super-categories,

Table 3 Some hyperparameters of the models

	Classification model		Meta-learning model	
	mini-ImageNet	tiered-ImageNet	mini-ImageNet	tiered-ImageNet
Optimizer	$Momentum = 0.9$ $Learning\ rate = 0.1$ $Attenuation\ factor = 0.1$ $Weight\ decay = 5 \times 10^{-4}$		$Momentum = 0.9$ $Learning\ rate = 5 \times 10^{-3}$ $Attenuation\ factor = 0.1$ $Weight\ decay = 5 \times 10^{-4}$	
Epoch	100	120	50	
Batch size	128	512	$Support\ set : 5 \times 1/5 \times 5$ $(5way\ 1shot/5way\ 5shot)$ $Query\ Set : 5 \times 15$	
Decay epoch	90	40, 80	25	

each super-category contains 10-30 classes, a total of 608 classes and 779,165 images. The 34 super-categories can be divided into the base class set (20 super-categories), validation set (6 super-categories) and new class set (8 super-categories).

4.2 Implementation details

The models are trained on the base class set, and then evaluated on the new class set. The model is implemented by Python 3.8 with CUDA 11.0. The two NVIDIA GeForce

Table 4 Average accuracy confidence intervals (%) of different meta-learning methods on the mini-ImageNet dataset

				5-way	
Method	Venue	Year	Backbone	1-shot(%)	5-shot(%)
AFHN [31]	CVPR	2020	ResNet-18	62.38 ± 0.72	78.16 ± 0.56
TransMatch [32]	CVPR	2020	WRN-28-10	63.02 ± 1.07	82.24 ± 0.59
E^3BM [33]	ECCV	2020	ResNet-12	63.80 ± 0.40	80.10 ± 0.30
Neg-Cosine [34]	ECCV	2020	ResNet-12	63.85 ± 0.81	81.57 ± 0.56
DSN-MR [35]	CVPR	2020	ResNet-12	64.60 ± 0.72	79.51 ± 0.50
Y. Tian et al. [36]	ECCV	2020	ResNet-12	64.82 ± 0.60	82.14 ± 0.43
J. Kim et al. [37]	ECCV	2020	ResNet-12	65.08 ± 0.86	82.70 ± 0.54
FEAT [38]	CVPR	2020	WRN-28-10	65.10 ± 0.20	81.11 ± 0.14
Dhillon et al. [39]	ICLR	2020	WRN-28-10	65.73 ± 0.68	78.40 ± 0.52
DeepEMD [40]	CVPR	2020	ResNet-12	65.91 ± 0.82	82.41 ± 0.56
Centroid alignment [41]	ECCV	2020	WRN-28-10	65.92 ± 0.60	82.85 ± 0.55
NCA nearest centroid [42]	NeurIPS	2021	ResNet-12	62.55 ± 0.12	78.27 ± 0.09
Meta-Baseline [7]	ICCV	2021	ResNet-12	63.17 ± 0.23	79.26 ± 0.17
MixtFSL [43]	ICCV	2021	ResNet-12	63.98 ± 0.79	82.04 ± 0.49
PSST [44]	CVPR	2021	ResNet-12	64.05 ± 0.49	80.24 ± 0.49
PSST [44]	CVPR	2021	WRN-28-10	64.16 ± 0.44	80.64 ± 0.32
P-Transfer [45]	AAAI	2021	ResNet-12	64.21 ± 0.77	80.38 ± 0.59
Constellation Net [46]	ICLR	2021	ResNet-12	64.89 ± 0.23	79.95 ± 0.37
ArL [47]	CVPR	2021	ResNet-12	65.21 ± 0.58	80.41 ± 0.49
PT+MAP [48]	arXiv	2021	WRN-28-10	65.35 ± 0.20	$\mathbf{83.87 \pm 0.13}$
Our-Self	/	/	ResNet-12	63.73 ± 0.34	81.28 ± 0.28
Our-Self-ML	/	/	ResNet-12	65.80 ± 0.34	81.83 ± 0.28
SPRM	/	/	ResNet-12	$\mathbf{66.35 \pm 0.34}$	82.24 ± 0.27

is the DropBlock application. The best results are shown in bold

Table 5 Average accuracy confidence intervals (%) of different meta-learning methods on the tired-ImageNet dataset

Method	Venue	Year	Backbone	5-way	
				1-shot(%)	5-shot(%)
PPA [49]	CVPR	2018	WRN-28-10	65.65 ± 0.92	83.40 ± 0.65
CC+rot [26]	ICCV	2019	WRN-28-10	62.93 ± 0.45	79.87 ± 0.33
MetaOpt Net [11]	CVPR	2019	ResNet-12	65.99 ± ±0.72	81.56 ± 0.53
LEO [10]	ICLR	2019	WRN-28-10	66.33 ± 0.05	81.44 ± 0.09
Ravichandran et al. [50]	ICCV	2019	ResNet-12	66.87	82.64
wDAE-GNN [51]	CVPR	2019	WRN-28-10	68.18 ± 0.16	83.09 ± 0.12
CTM [52]	CVPR	2019	ResNet-18	68.41 ± 0.39	84.28 ± 1.73
AM3 [53]	NeurIPS	2019	ResNet-12	69.08 ± ±0.47	82.58 ± 0.31
DSN-MR [35]	CVPR	2020	ResNet-12	67.39 ± 0.82	82.85 ± 0.56
F. Wu et al. [54]	ECCV	2020	Capsule Net	69.87 ± 0.32	86.35 ± 0.41
FEAT [38]	CVPR	2020	WRN-28-10	70.41 ± 0.23	84.38 ± 0.16
NCA nearest centroid [42]	NeurIPS	2021	ResNet-12	68.35 ± 0.13	83.20 ± 0.10
Meta-Baseline [7]	ICCV	2021	ResNet-12	68.62 ± 0.27	83.74 ± 0.18
BML [55]	ICCV	2021	ResNet-12	68.99 ± 0.50	85.49 ± 0.34
PT+MAP [48]	arXiv	2021	DenseNet121	69.96 ± 0.22	**86.45 ± 0.10**
Our-Self	/	/	ResNet-12	69.93 ± 0.33	85.28 ± 0.25
Our-Self-ML	/	/	ResNet-12	70.46 ± ±0.33	85.52 ± 0.25
SPRM	/	/	ResNet-12	**70.70 ± 0.33**	85.40 ± ±0.25

The best results are shown in bold

RTX 2080 Ti GPUs are utilized. Some hyperparameters of the models are shown in Table 3.

The evaluation indicator of this experiment is the confidence interval (z=1.96) of the average precision P of M samples at the 95% confidence level, i.e. $P \pm R_{interval}$. The calculation of the confidence interval radius $R_{interval}$ is shown as:

$$R_{interval} = Z\sqrt{\frac{P(1-P)}{M}} \tag{13}$$

Table 6 Ablation experiment results

Self	m/c	rr	mini-ImageNet 5-way		tired-ImageNet 5-way	
			1-shot(%)	5-shot(%)	1-shot(%)	5-shot(%)
			58.89 ± 0.35	77.99 ± 0.30	68.00 ± 0.33	83.47 ± 0.27
	✓		60.37 ± 0.35	77.60 ± 0.30	68.13 ± 0.33	83.51 ± 0.27
	✓ ✓		62.62 ± 0.35	80.01 ± 0.29	68.35 ± 0.33	84.47 ± 0.26
		✓	62.59 ± 0.35	79.35 ± 0.29	68.32 ± 0.33	84.01 ± 0.26
	✓	✓	61.88 ± 0.35	78.39 ± ±0.29	68.41 ± 0.33	83.85 ± 0.26
	✓ ✓	✓	62.94 ± 0.35	80.00 ± 0.29	69.41 ± 0.33	84.57 ± 0.26
✓			63.73 ± 0.34	81.28 ± 0.28	69.93 ± 0.33	85.28 ± 0.25
✓	✓		64.41 ± 0.34	80.60 ± ±0.28	70.18 ± 0.33	85.17 ± 0.25
✓	✓ ✓		65.80 ± 0.34	81.83 ± 0.28	70.46 ± 0.33	**85.52 ± 0.25**
✓		✓	65.15 ± 0.34	81.23 ± 0.28	70.41 ± 0.33	85.16 ± 0.25
✓	✓	✓	65.32 ± 0.34	81.16 ± 0.28	70.36 ± 0.33	85.30 ± 0.25
✓	✓ ✓	✓	**66.35 ± 0.34**	**82.24 ± 0.27**	**70.70 ± 0.33**	85.40 ± 0.25

"Self" indicates whether to use self-supervised loss; "rr" indicates whether to use resistance regularization; in "m/c", "✓" represents the loss function of the meta-learning model is the cross-entropy loss, and "✓ ✓" represents the loss function of the meta-learning model is ML. The best results are shown in bold

Table 7 The evaluation results of different loss function on mini-ImageNet

Index	Loss	1-shot(%)	5-shot(%)	Time
①	Cross-entropy Loss	64.41 ± 0.34	80.60 ± 0.28	1.4×10^{-4}s
②	MS	58.74 ± 0.35	79.10 ± 0.29	4.7×10^{-1}s
③	MS weighting + Our mining	66.24 ± 0.34	81.71 ± 0.28	3.0×10^{-2}s
④	Our weighting + MS mining	59.06 ± 0.35	80.01 ± 0.29	5.1×10^{-1}s
⑤	ML(our)	$\mathbf{66.35 \pm 0.34}$	$\mathbf{82.24 \pm 0.27}$	2.9×10^{-2}s

The best results are shown in bold

Table 8 The evaluation results of different loss function on tired-ImageNet

Index	Loss	1-shot(%)	5-shot(%)	Time
①	Cross-entropy Loss	70.18 ± 0.33	85.17 ± 0.25	1.4×10^{-4}s
②	MS	69.75 ± 0.33	$85.15 \pm \pm 0.25$	4.7×10^{-1}s
③	MS weighting + Our mining	70.55 ± 0.33	85.19 ± 0.25	3.0×10^{-2}s
④	Our weighting + MS mining	69.76 ± 0.33	85.17 ± 0.25	5.1×10^{-1}s
⑤	ML(our)	$\mathbf{70.70 \pm 0.33}$	$\mathbf{85.40 \pm 0.25}$	2.9×10^{-2}s

The best results are shown in bold

5 Results and discussion

5.1 The performance evaluation of SPRM

In this paper, different few-shot learning methods are compared on mini-ImageNet and tiered-ImageNet dataset. The experimental results are shown in Tables 4 and 5, where "Our-self" is only the classification model with the self-supervised technology, and "Our-self-ML" is the meta-learning model combined with the self-supervised classification model, which uses the ML without adding resistance regularization term.

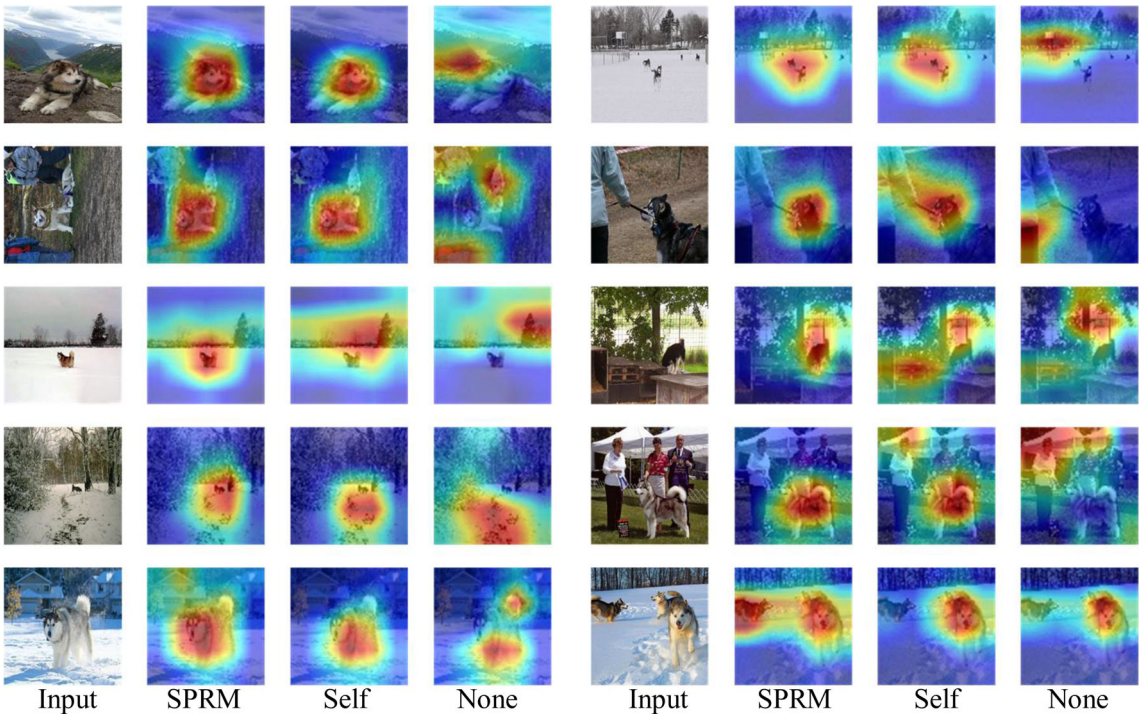

Fig. 4 The heat map visualization of different methods by Grad-CAM

Table 9 The different methods for visualization analysis

Method	Classification model Self-supervised	Meta-learning model ML	Resistance regularization
None			
Self	✓		
ML+Rr		✓	✓
SPRM	✓	✓	✓

In Tables 4 and 5, the classification accuracy of SPRM on the 5-way 1-shot task of mini-ImageNet reaches 66.35%, and it reaches 82.24% on the 5-way 5-shot task, which demonstrates the model has better performance than other few-shot learning methods. The classification accuracy of SPRM on the 5-way 1-shot task of tired-ImageNet reaches 70.70%, and it reaches 85.40% on the 5-way 5-shot task. These results show that the SPRM has excellent performance and generalization.

5.2 Ablation study

The effect of the three technologies (including the rotation self-supervised loss, ML and resistance regularization) are studied in the ablation experiments. The results are shown in Table 6. When the resistance regularization is used alone, it can be regarded as a loss function; when the ML and resistance regularization are not used, the cross-entropy loss function is used as the loss function; when these three technologies are not used, the framework does not contain the meta-learning model.

In Table 6, the performance of our proposed method is the best. The application of the rotation self-supervised loss in the classification model can greatly improve the model performance. Compared with the cross-entropy loss, ML has obvious improvement on the classification tasks. When the resistance regularization is used as a loss function alone, it can also increase the prediction accuracy of the

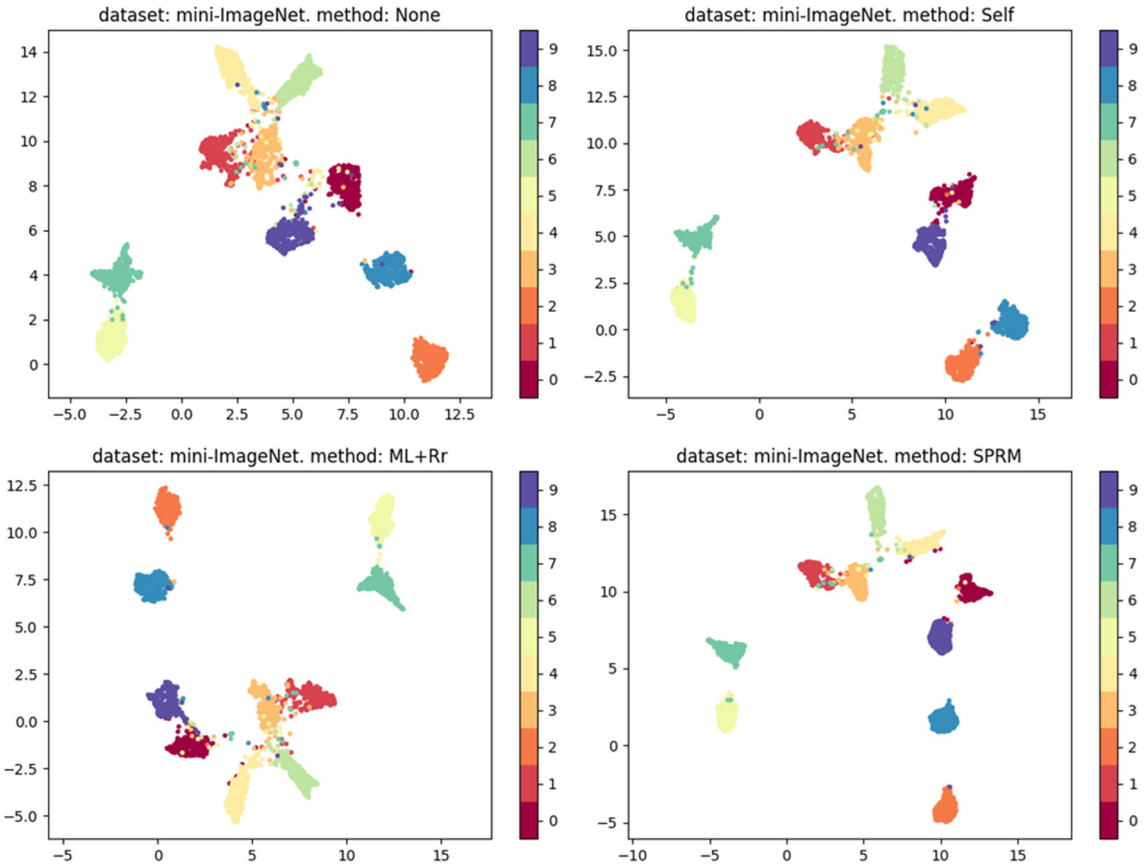

Fig. 5 The feature vector visualization of several categories in mini-ImageNet by UMAP(2-dim)

few-shot classification model. The ML with the resistance regularization has the best performance when the self-supervised technique is not considered. To sum up, the three proposed techniques all have improved the performance of the model.

In addition, the effect of the mining and weighting technologies of the MS loss and ML are also explored in the ablation study. The evaluation results on mini-ImageNet and tiered-ImageNet dataset are shown in Tables 7 and 8, respectively.

In Tables 7 and 8, the prediction accuracy of ML is the highest, and both the mining and weighting strategy in ML have a positive gain on the model performance. Comparing the results of ②③ and ④⑤ in these two tables, the effect of the mining strategy in ML is better than that in MS loss. Comparing the experimental results of ②④ and ③⑤ in these two tables, the weighting strategy of ML can improve the model performance in most cases. In addition, according to ②③ and ④⑤, the calculation time of the mining strategy in ML has greatly reduced. To sum up, the mining strategy of ML can not only enhance the model performance, but also improve the computational efficiency.

5.3 Visualization analysis

The heat maps of different methods are visualized by Grad-CAM [56], they are shown in Fig. 4, the feature encoder pays more attention to the warm tone region and ignores the cold tone region. The different methods described in Fig. 4 are shown in Table 9.

In Fig. 4, for "None" and "Self", when there are many objects in the image, the attention of the feature encoder is easily influenced by the interfering objects; when there are few objects in the image and the target is large, the feature encoder can quickly notice the target, but the attention area is small, the model cannot obtain the complete semantic information. Compared with the feature encoders of "Self" and "None", SPRM can pay more attention to the whole target and capture more complete semantic information.

The feature vectors of several categories in mini-ImageNet and tiered-ImageNet dataset are shown in Figs. 5 and 6 by UMAP [57], respectively. In Figs. 5 and 6, the feature vectors of the same category are more compact in "Self" than that in "None", which proves that the rotation self-supervised loss can improve the feature representation

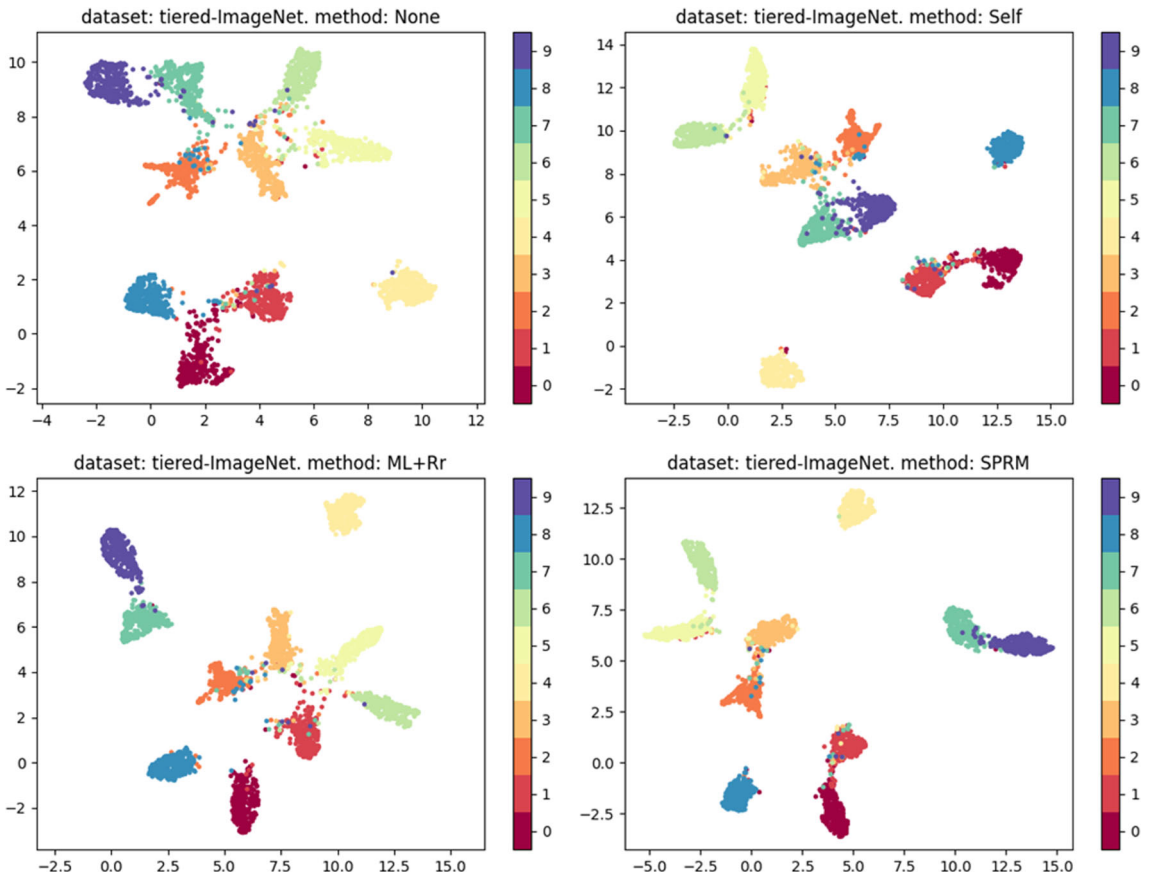

Fig. 6 The feature vector visualization of several categories in tired-ImageNet by UMAP(2-dim)

ability of the classification model. The same phenomenon occurs in "ML+Rr", which further confirms that the ML can reduce the intra-class difference and expand the inter-class difference; the SPRM method combines the advantages of the rotation self-supervised loss, ML and resistance regularization to optimize the decision boundary and improve the model performance.

6 Conclusion

In this paper we propose a new few-shot classification model named self-supervised pairwise-sample resistance model (SPRM). It contains a classification model and a meta-learning model. The rotation self-supervised loss is utilized as an auxiliary loss in the classification model to obtain the feature extractor with strong representational ability, which is used as an initialize feature extractor in the meta-learning model; and the meta loss (ML) and resistance regularization are proposed and applied in the meta-learning model to improve the model performance. SPRM is evaluated on the 5-way 1-shot and 5-way 5-shot tasks of mini-ImageNet and tired-ImageNet. The experimental results indicate that our method is superior to the other advanced methods in few-shot classification tasks.

Acknowledgements This work was funded by the National Natural Science Foundation of China under Grant 51774219, Key R& D Projects in Hubei Province under grant 2020BAB098.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al (2016) Matching networks for one shot learning. Adv Neural Inf Process Syst 29
2. Chen W-Y, Liu Y-C, Kira Z, Wang Y-CF, Huang J-B (2019) A closer look at few-shot classification. arXiv:1904.04232
3. Guo Y, Codella NC, Karlinsky L, Codella JV, Smith JR, Saenko K, Rosing T, Feris R (2020) A broader study of cross-domain few-shot learning. In: European conference on computer vision, Springer pp 124–141
4. Tseng H-Y, Lee H-Y, Huang J-B, Yang M-H (2020) Cross-domain few-shot classification via learned feature-wise transformation. arXiv:2001.08735
5. Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F (2020) A survey on contrastive self-supervised learning. Technologies 9(1):2
6. Sun Q, Liu Y, Chua T-S, Schiele B (2019) Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 403–412
7. Chen Y, Wang X, Liu Z, Xu H, Darrell T (2020) A new meta-baseline for few-shot learning
8. Lake B, Salakhutdinov R, Gross J, Tenenbaum J (2011) One shot learning of simple visual concepts. In: Proceedings of the annual meeting of the cognitive science society, vol 33
9. Wang X, Yu F, Wang R, Darrell T, Gonzalez JE (2019) Tafe-net : Task-aware feature embeddings for low shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1831–1840
10. Rusu AA, Rao D, Sygnowski J, Vinyals O, Pascanu R, Osindero S, Hadsell R (2018) Meta-learning with latent embedding optimization. arXiv preprint arXiv:1807.05960
11. Lee K, Maji S, Ravichandran A, Soatto S (2019) Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10657–10665
12. Ge W (2018) Deep metric learning with hierarchical triplet loss. In: Proceedings of the European conference on computer vision (ECCV), pp 269–285
13. He X, Zhou Y, Zhou Z, Bai S, Bai X (2018) Triplet-center loss for multi-view 3d object retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1945–1954
14. Sohn K (2016) Improved deep metric learning with multi-class n-pair loss objective. Adv Neural Inf Process Syst 29
15. Wang J, Zhou F, Wen S, Liu X, Lin Y (2017) Deep metric learning with angular loss. In: Proceedings of the IEEE international conference on computer vision, pp 2593–2601
16. Wu C-Y, Manmatha R, Smola AJ, Krahenbuhl P (2017) Sampling matters in deep embedding learning. In: Proceedings of the IEEE international conference on computer vision, pp 2840–2848
17. Doersch C, Gupta A, Efros AA (2015) Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision, pp 1422–1430
18. Noroozi M, Favaro P (2016) Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision, Springer, pp 69–84
19. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA (2016) Context encoders : feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2536–2544
20. Zhang R, Isola P, Efros AA (2016) Colorful image colorization. In: European conference on computer vision, Springer, pp 649–666
21. Zhang R, Isola P, Efros AA (2017) Split-brain autoencoders : unsupervised learning by cross-channel prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1058–1067
22. Gidaris S, Singh P, Komodakis N (2018) Unsupervised representation learning by predicting image rotations. arXiv:1803.07728
23. Hjelm RD, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, bengio Y (2018)
24. Tian Y, Krishnan D, Isola P (2020) Contrastive multiview coding. In: European conference on computer vision, Springer, pp 776–794
25. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: International conference on machine learning, PMLR, pp 1597–1607
26. Gidaris S, Bursuc A, Komodakis N, Pérez P, Cord M (2019) Boosting few-shot visual learning with self-supervision. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 8059–8068

27. Lee H, Hwang SJ, Shin J (2019) Rethinking data augmentation. Self-supervision and self-distillation
28. Wang X, Han X, Huang W, Dong D, Scott MR (2019) Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5022–5030
29. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10(2)
30. Ren M, Triantafillou E, Ravi S, Snell J, Swersky K, Tenenbaum JB, Larochelle H, Zemel RS (2018) Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676
31. Li K, Zhang Y, Li K, Fu Y (2020) Adversarial feature hallucination networks for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13470–13479
32. Yu Z, Chen L, Cheng Z, Luo J (2020) Transmatch : a transfer-learning scheme for semi-supervised few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12856–12864
33. Liu Y, Schiele B, Sun Q (2020) An ensemble of epoch-wise empirical bayes for few-shot learning. In: European conference on computer vision, Springer, pp 404–421
34. Liu B, Cao Y, Lin Y, Li Q, Zhang Z, Long M, Hu H (2020) Negative margin matters : understanding margin in few-shot classification. In: European conference on computer vision, Springer, pp 438–455
35. Simon C, Koniusz P, Nock R, Harandi M (2020) Adaptive subspaces for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4136–4145
36. Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P (2020) Rethinking few-shot image classification : a good embedding is all you need? In: European conference on computer vision, Springer, pp 266–282
37. Kim J, Kim H, Kim G (2020) Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In: European conference on computer vision, Springer, pp 599–617
38. Ye H-J, Hu H, Zhan D-C, Sha F (2020) Few-shot learning via embedding adaptation with set-to-set functions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8808–8817
39. Dhillon GS, Chaudhari P, Ravichandran A, Soatto S (2019) A baseline for few-shot image classification. arXiv:1909.02729
40. Zhang C, Cai Y, Lin G, Shen C (2020) Deepemd : few-shot image classification with differentiable earth mover's distance and structured classifiers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12203–12213
41. Afrasiyabi A, Lalonde J-F, Gagné C (2020) Associative alignment for few-shot image classification. In: European conference on computer vision, Springer, pp 18–35
42. Laenen S, Bertinetto L (2021) On episodes, prototypical networks, and few-shot learning. Adv Neural Inf Process Syst 34:24581–24592
43. Afrasiyabi A, Lalonde J-F, Gagné C (2021) Mixture-based feature space learning for few-shot image classification. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9041–9051
44. Chen Z, Ge J, Zhan H, Huang S, Wang D (2021) Pareto self-supervised training for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13663–13672
45. Shen Z, Liu Z, Qin J, Savvides M, Cheng K-T (2021) Partial is better than all : revisiting fine-tuning strategy for few-shot learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 35. pp 9594–9602
46. Xu W, Xu Y, Wang H, Tu Z (2021) Attentional constellation nets for few-shot learning. In: International conference on learning representations
47. Zhang H, Koniusz P, Jian S, Li H, Torr PH (2021) Rethinking class relations : absolute-relative supervised and unsupervised few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9432–9441
48. Hu Y, Gripon V, Pateux S (2021) Leveraging the feature distribution in transfer-based few-shot learning. In: International conference on artificial neural networks, Springer, pp 487–499
49. Qiao S, Liu C, Shen W, Yuille AL (2018) Few-shot image recognition by predicting parameters from activations. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7229–7238
50. Ravichandran A, Bhotika R, Soatto S (2019) Few-shot learning with embedded class models and shot-free meta training. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 331–339
51. Gidaris S, Komodakis N (2019) Generating classification weights with gnn denoising autoencoders for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 21–30
52. Li H, Eigen D, Dodge S, Zeiler M, Wang X (2019) Finding task-relevant features for few-shot learning by category traversal. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1–10
53. Xing C, Rostamzadeh N, Oreshkin BO, Pinheiro PO (2019) Adaptive cross-modal few-shot learning. Adv Neural Inf Process Syst 32
54. Wu F, Smith JS, Lu W, Pang C, Zhang B (2020) Attentive prototype few-shot learning with capsule network-based embedding. In: European conference on computer vision, Springer, pp 237–253
55. Zhou Z, Qiu X, Xie J, Wu J, Zhang C (2021) Binocular mutual learning for improving few-shot classification. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 8402–8411
56. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam : visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
57. McInnes L, Healy J, Melville J (2018) Umap : uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426

Weigang Li received the Ph.D degree in Control theory and control engineering from Northeastern University, Liaoning, China, in 2013. He is currently a professor with School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, China and PhD supervisor. His research interests include artificial intelligence and machine learning algorithms.

Lu Xie received the B.S. degree in Electronic information from Xinyang Normal University, Henan, China, in 2018. She is currently a doctoral candidate with the School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, China. Her research interests include information processing and deep-learning algorithm.

Ping Gan received the MA. Eng. Degree in Electronics and Communication Engineering, Wuhan University of Science and Technology, Hubei, China, 2022. He currently works as a Qunar algorithm engineer. His research interests include few-shot learning and multimodal machine learning. Corresponding author of this paper.

Yuntao Zhao received the Ph.D degree in Control theory and control engineering from University of Science and Technology Beijing, Beijing, China, in 2010. He is currently an associate professor with School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, China. His research interests include intelligent optimization algorithm and Robot Path Planning.