Decision Boundary Optimization for Few-shot Class-Incremental Learning

Chenxu Guo[1] , Qi Zhao[1] , Shuchang Lyu[1] , Binghao Liu[1] , Chunlei Wang[1] , Lijiang Chen[1] and Guangliang Cheng[*2]
[1]Department of Electronic Information Engineering, Beihang University
[2]Department of Computer Science, University of Liverpool

Abstract

Few-shot class-incremental learning (FSCIL) is gaining prominence in real-world machine learning applications, including image classification and face recognition. Existing methods often employ parameter freezing for the backbone and classify based on metric learning. However, these methods suffer from two significant problems. Firstly, training the backbone solely on base classes limits its performance on novel classes due to information loss. Secondly, conventional metric-based strategies for prototype generation tend to introduce confusion in decision boundaries during few-shot tasks. To address these challenges, we propose a novel approach called Decision Boundary Optimization Network (DBONet) for few-shot class-incremental learning. To tackle the first issue, DBONet incorporates an augmentation feature extractor along with a corresponding loss function. This augmentation feature extractor combines samples from different categories to capture richer features. For the second issue, we leverage limited sample representativeness information by introducing the Prototype Generation Module (PGM) into DBONet, enabling the generation of more representative prototypes. The prototypes produced by PGM significantly contribute to the accurate delineation of decision boundaries. Furthermore, we exploit intra-class information to enhance classification precision. Extensive experiments on CIFAR100, mini*ImageNet, and CUB200 datasets demonstrate that our proposed approach achieves new state-of-the-art results.*

1. Introduction

Image recognition has garnered significant attention in recent years [23, 25, 27, 38, 41]. The ability of models to acquire new visual knowledge from limited samples has become a central focus for numerous researchers [10, 11, 42]. In practical applications, visual recognition systems often encounter challenges related to data privacy or device memory limitations, which pose difficulties in retraining the model using previously learned data [46]. To address this issue, class-incremental learning (CIL) has emerged as a dedicated approach, where the learning of each class is treated independently. The primary objective of CIL is to enable a learning system to continuously acquire new knowledge from novel classes while retaining a substantial portion of the previously learned knowledge [36].

*Corresponding author: Guangliang.Cheng@liverpool.ac.uk

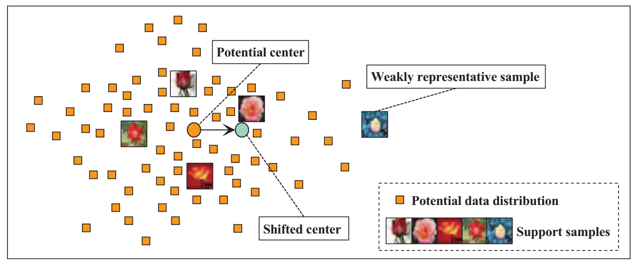


Figure 1: The weakly representative samples result that class prototype shifted with potential center. The green circle point denotes the prototype produced with limited samples, and the orange circle point denotes the potential class center.

Conventional class-incremental learning (CIL) tasks typically operate under the assumption that an ample amount of data for novel categories is available for model learning. However, in practical scenarios, it is often the case that there is an insufficiency of data pertaining to novel categories, thereby necessitating the learning system to effectively acquire knowledge using limited samples [45, 60]. These specific learning scenarios are referred to as **F**ew-**S**hot **C**lass-**I**ncremental **L**earning (FSCIL) [46].

FSCIL exhibits remarkable parallels with the learning process observed in human beings. Humans continuously receive and assimilate new knowledge from their surroundings, and this learning occurs gradually over time [19]. Importantly, humans tend to retain the vast majority of the knowledge they have acquired, making it challenging to forget previously learned information. Furthermore, humans possess the remarkable ability to leverage their existing knowledge to comprehend novel content. This ability to build upon prior knowledge plays a vital role in the human learning process.

Deep learning models currently fall short of achieving human-level performance, particularly when trained on limited samples, which often leads to underfitting issues. Additionally, disregarding previous data while learning new classes can result in catastrophic forgetting of the model [36]. This problem is particularly pronounced when dealing with limited samples of novel categories compared to conventional continual incremental learning (CIL) tasks.

In recent times, several studies have employed metric learning strategies to preserve the model's recognition ability on base classes. Existing FSCIL frameworks [5, 6, 9, 13, 37, 46, 55, 63] have made significant advancements. However, these methods primarily rely on optimizing the cross-entropy loss for a single category and cannot be effectively applied when dealing with dual category virtual samples [53, 15].

The semantic features across different categories typically exhibit a lack of generality, giving rise to three primary challenges when dealing with novel class samples in comparison to previous class samples. Firstly, *the features of novel classes are often not fully expressed*, resulting in the feature vectors' norm being statistically smaller than that of the base class samples. Consequently, the model tends to confuse categories between novel and base classes, for instance, mistaking dolphins from the base class for sharks in the novel class. As a result, severe category confusion problems often occur. Secondly, *the support set's samples are often non-representative*, a common issue in real-world applications like face recognition. Obtaining ideal photos with strong feature representation in few-shot conditions, such as frontal shots under good lighting, is challenging. To overcome this, we propose a prototype generation module (PGM) that produces superior class prototypes in latent space, mitigating the impact of potential outliers on class representation learning. Additionally, *our analysis reveals varying intra-class variances for each class in the latent space*. This difference in the spatial distribution of feature vectors between classes poses challenges when using the equal category decision method. To address this, we propose a novel classifier based on intra-class variance for a more accurate decision boundary in latent space.

To address the aforementioned challenges, we present a novel architecture DBONet, which comprises three key components: the *augmentation feature extractor*, the *prototype generation module*, and the *intra-class variance classifier*. Specifically, the augmentation feature extractor aims to obtain robust global features from images, thus achieving fully expressed features. The prototype generation module (PGM) aims to create more suitable prototypes that adapt well to conditions with limited samples. Finally, the intra-class variance classifier is introduced to tackle the decision boundary shift problem. By effectively eliminating above issues, our approach further enhances the overall performance and accuracy of the model.

The contributions of this paper can be summarized as follows:

- A novel end-to-end FSCIL learning framework, Decision Boundary Optimization Network (DBONet), is proposed to acquire highly representative features, thus achieving much better performance.
- A Prototype Generation Module (PGM) is proposed, whereby the utilization of representative samples enables the derivation of significantly improved class prototypes in the latent space.
- An intra-class variance classifer is employed to adaptively adjust the class decision boundary to reduce confusion between classes.
- Extensive experiments on three benchmark datasets demonstrate the state-of-the-art (SOTA) performance of the proposed method in FSCIL tasks.

2. Related Work

Class-incremental learning (CIL). The main content of CIL is making learning system can continually learn knowledge without forgetting [26, 28, 62]. The current methods can be generally divided into three types. Regularization Approaches [1, 20, 35, 58] mainly adopt regularization terms and classification loss to alleviate catastrophic forgetting. Some methods which using rehearsal and replay mechanism [16, 33, 36, 40] to prevent the forgetting of previous tasks. Another group of study [2, 14, 52] aim to mute bias from most recently learned task to tackle CIL tasks. In CIL tasks, novel classes usually have adequate samples. So many methods proposed for CIL task may suffer from reduced efficacy with the condition of limited novel samples.

Few-shot learning (FSL). The purpose of few-shot learning is to be able to learn valid information from limited samples, which usually requires pre-training or meta-learning to obtain a model that can quickly adapt to few-shot scene [34, 43, 50]. Related few-shot learning study primarily includes model-based, metric-base, optimization-based methods. Model-based methods [29, 30, 31] involve model architectures specifically tailored for fast learning. Metric-based methods [17, 22, 32, 42, 45, 49, 56] focus on how to pull support samples and query samples in latent space, while scaling up the distance between different classes. Optimization-based methods [11, 39, 44] learn an optimizer through meta learning which can quickly adapt to new categories with limited samples. Most of FSL study don't consider differentiating between base classes and novel classes together.

Few-shot class-incremental learning (FSCIL). There are some finetune-based methods to tackle FSCIL tasks. Finetune-based methods optimize parameters with novel data using various balance mechanism to solve catastrophic forgetting and overfitting problems [5, 9, 19, 37, 46, 63]. Recently, methods which freeze backbone during novel classes learning perform significant results. Zhang *et al.* [59] use a pseudo incremental learning method to train a attention-base module to enhance the model performance. Zhou *et al.* [61] propose forward compatible training by assigning virtual prototypes to compress the embedding of base classes and reserve space for novel classes. The above methods do not effectively solve the problem of confusion and scope imbalance in the latent space. We reckon that the decision boundary of the latent space can be further optimized by adjusting the prototype.

Prototypical learning (PL). The basic assumption of prototypical learning is that each category has a potential center, and samples between different categories are approximately separable in latent space [24, 42]. Yue *et al.* [57] leverage prototype which can preserve semantic structure for Unsupervised Domain Adaptation. Chen *et al.* [4] perform fine-grained image classification with comparing part of images and prototypes. Li *et al.* [21] use limited support samples as prototype to guide few-shot image segmentation. There are also some study which aim to exploit potential of the prototype sufficiently. Yang *et al.* [54] propose a new approach to handle outlier data by utilizing trained prototypes and an assumed Gaussian distribution. Deng *et al.* [8] argue that the prototype should be treated as a distribution instead of a point in the latent space. Inspired by above mentioned prototypical learning researches, we propose to sufficiently explore the role of prototypes to tackle the FSCIL tasks.

3. Method

3.1. Problem Formulation

In an N-way K-shot FSCIL task, let training set streams as $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_n$, totally $(n+1)$ sessions, and the corresponding samples label sets are $Y_0, Y_1, Y_2, \cdots, Y_n$. The test set streams is $\mathcal{D}^t_0, \mathcal{D}^t_1, \mathcal{D}^t_2, \cdots, \mathcal{D}^t_n$, and the corresponding samples label sets are $Y^t_0, Y^t_1, Y^t_2, \cdots, Y^t_n$. $\mathcal{D}_0$ is the base session training sets with sufficient samples. The training label sets from different sessions are disjoint, which can be formulated as $Y_i \cap Y_j = \emptyset$ for $i \neq j$. In other words, the model can only access a set of specific categories of training data at different sessions. At the end of each session, the model needs to evaluate on the previous data and current data together. So the test set label space in i-th session can be summarized as $Y^t_i = Y_0 \cup Y_1 \cup Y_2 \cup \cdots \cup Y_{i-1} \cup Y_i$. The novel sessions in training set usually have only a limited amount of samples. Without loss of generality, we use N classes, and each class has K samples. Taking *mini*ImageNet dataset as an example, in the session 0, there are 60 base classes, and each class has 500 training samples. Other sessions contain $5\ ways \times 5\ shot$, totally 25 training samples for each session.

3.2. Augmented Feature Training

Our training pipeline mainly comprises two stages. In the first stage, we train the encoder using feature augmentation to obtain a global feature representation of each image. In the second stage, we categorize samples into two types based on their representativeness and then train the prototype generation module to obtain a prototype for each class.

In the previous work CEC [59], encoders and attention mechanisms were trained using a feature enhancement method involving image rotation. Building upon this, we introduce an innovative approach to enrich the feature space further, leveraging manifold mixup augmentation [48]. The architecture of our proposed method is illustrated in Fig. 2. The $h(\cdot)$ indicates the pre-encoder which consists of the first three layers of ResNet [12] backbone, and the $g_s(\cdot)$ and $g_a(\cdot)$ indicate the *stable* feature extractor and the *augmentation* feature extractor. This can be shown as following:

$$E(x) = Concat(g_s(h(x)), g_a(h(x))) \tag{1}$$

where $E(x)$ is the output global feature. In the following, we will use the $E(\cdot)$ to represent the inference function of the DBONet.

As depicted in Fig. 2, the model processes an input image and generates two distinct feature vectors using the stable feature extractor and the augmentation feature extractor. We refer to these feature vectors as the "stable feature" and the "augmented feature" of the image, respectively. Subsequently, these two feature vectors are concatenated to create a comprehensive global feature representation for the image.

Our study aims to optimize high-level semantic features by fusing diverse categories of image features. To achieve this, we follow a three-step process: 1) We input pairs of images from different classes into the pre-encoder, resulting in two dense feature maps. 2) The two feature maps obtained from different categories are fused using a randomly selected weight coefficient λ from the interval $[0.45, 0.55]$. 3) These fused feature maps are then passed through two feature extraction modules g_s and g_a, dedicated to extracting stable and augmented features, respectively.

$$F' = \lambda * h(x_i) + (1 - \lambda) * h(x_j) \tag{2}$$

The process involves taking images x_i and x_j from different base categories and passing them through the third layer of ResNet to obtain dense features $h(x_i)$ and $h(x_j)$. These dense features are fused to create the combined feature F'. Subsequently, the fused dense feature is fed into an

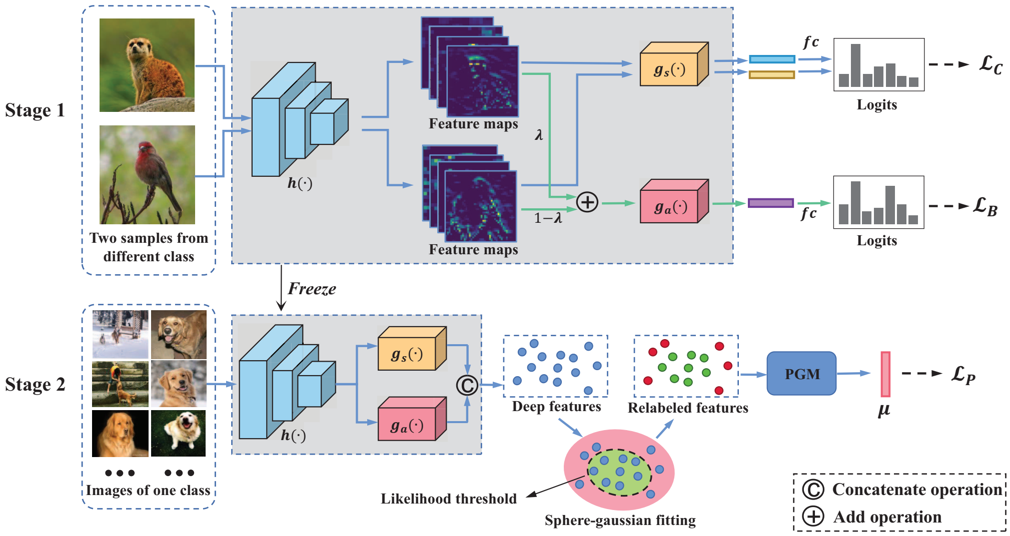


Figure 2: Structure of DBONet. The fc in stage 1 denotes the output feature vector multiple with classifier weight. The © in stage 2 denotes concatenate operation.

augmentation feature extractor to generate a global feature representation. Finally, the global feature representation is multiplied with matrix W as described in Eq. 3 to obtain logits output, and the loss is calculated using Eq. 4.

$$l = fc(g_a(F')) = W^\top g_a(F') \tag{3}$$

where l is the logit output and W is the classifier weight for each class.

$$\begin{aligned}\mathcal{L}_B &= BICELoss(l, \lambda_{y_1}, \lambda_{y_2}, y_1, y_2)\\ &= -[\sum_{k=1,2} \lambda_{y_k} l_{y_k} - log(\sum_{j=1,2} \lambda_{y_j} e^{l_{y_j}} + \sum_{i\neq y_1, i\neq y_2} e^{l_i})]\end{aligned} \tag{4}$$

where y_1 and y_2 are the labels of the samples which are fused, and λ_{y_1} and λ_{y_2} ($\lambda_{y_1} + \lambda_{y_2} = 1$) are the corresponding fusion weight for the sample feature vectors. l_{y_1} and l_{y_2} are the logit output value for the class y_1 and y_2. The goal of $BICELoss$ is to enhance the recognition of virtual categories by effectively separating them from other class prototype vectors within the latent space.

In addition, we also optimize the pre-encoder $h(\cdot)$ and the stable feature extractor $g_s(\cdot)$ with using traditional cross entropy loss. So our training pipeline in stage 1 can be formulated as Algorithm. 1.

Algorithm 1: Stage 1 training pipeline.

Input: h, g_s and g_a with randomly initial parameters, $\mathcal{D}_0$
Output: trained models h, g_s, g_a
for *epoch in max_epochs* **do**
  for $images, y$ *in* $\mathcal{D}_0$ **do**
    $F \leftarrow h(images)$;
    randomly choose λ from $[0.45, 0.55]$;
    $F_m, y_m \leftarrow$ randomly permute dense feature F, y in dimension of batchsize ;
    $F' \leftarrow \lambda * F + (1-\lambda) * F_m$;
    $\mathcal{L}_C \leftarrow CrossEntropy(fc(g_s(F)), y)$;
    $\mathcal{L}_B \leftarrow BICELoss(fc(g_a(F')), \lambda, 1-\lambda, y, y_m)$;
    optimize model h, $g_s$with $\mathcal{L}_C$;
    optimize model h, $g_a$with $\mathcal{L}_B$;
  end
end

3.3. Cosine Variance Classifier

Consider a simple toy example, illustrated in Fig. 3, where two distinct categories exhibit Gaussian distributions with varying variances in the latent space. The ideal classification boundary, represented by the green dashed line in Fig. 3, should accurately separate the two categories. However, when utilizing an unweighted Euclidean distance measurement for classification, the decision boundary of the two distributions is depicted as the red line in Fig. 3, which lies at the midpoint between the two distributions. Unfortunately, this red line classification boundary leads to an increased classification error. To address this issue, we propose the need for further optimization of the decision boundary within the latent space. Our objective is to minimize the classification error as much as possible, ensuring an accurate separation of categories and improving the overall performance of the classification model.

In the latent space, the feature vectors of each class of image occupy a certain range. In the experiment, we find that the range of different categories is not always allocated according to the average distance, and the range occupied by different categories in latent space is also different. Con-

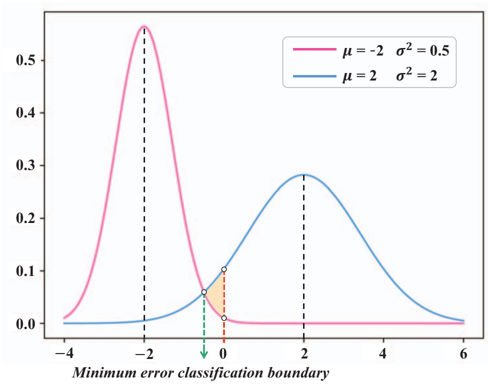


Figure 3: Different variance Gaussian distribution.

ventional average prototype cosine distance measurement cannot take this difference into account, so we propose a classifier that includes the cosine variance of the category.

To enhance the precision of class probability calculation, we assume that the data feature vector follows a spherical Gaussian distribution, which can be formulated as follows:

$$S(\nu;\mu,\sigma,a) = ae^{\sigma(\langle\mu,\nu\rangle-1)} \tag{5}$$

where ν is the input vector, a is a scalar coefficient, μ is center vector of spherical Gaussian distribution, σ is a scalar which can reflect the variance of the distribution, and $\langle\mu,\nu\rangle = \frac{\mu^\top\nu}{||\mu||_2*||\nu||_2}$ is the cosine function. We define the training set which belongs to the i-th class as $\mathcal{S}_i$, and the number of samples in $\mathcal{S}_i$ as $|\mathcal{S}_i|$. We use the Eq. 6 and Eq. 7 to calculate the mean μ_i and cosine variance σ_i of $\mathcal{S}_i$ in latent space.

$$\mu_i = \frac{1}{|\mathcal{S}_i|}\sum_{x\in\mathcal{S}_i} E(x) \tag{6}$$

Cosine variance is defined as follow:

$$\sigma_i = (\frac{1}{|\mathcal{S}_i|}\sum_{x\in\mathcal{S}_i}\langle\mu_i, E(x)\rangle)^\gamma \tag{7}$$

where γ is a factor to scale σ_i, and then we can calculate the probability $p(y=i|x)$ that the sample x belongs to i-th class using the Eq. 8.

$$p(y=i|x) = \frac{S(E(x);\mu_i,\sigma_i,1)}{\sum_{j=1}^{C} S(E(x);\mu_j,\sigma_j,1)} \tag{8}$$

where C is the number of total classes.

3.4. Prototype Generation Module

We propose a prototype generation module to generate prototype vectors closer to the potentially general center for each category, and calculate the average cosine variance for each category to measure the coverage of the category on the latent hypersphere space. To split representative and weakly representative samples from training set, we need to use the Eq. 8 to calculate the probability that one sample belongs to its label category. And by comparing the probability and threshold ϵ, we can select representative samples. And then, we label the samples of the base class 1 if it is representative, else 0. After that, we use the encoder $E(\cdot)$ to extract the feature vectors of all samples of the training set. And we train a prototype generation module with this feature set, the function of the module is to generate a more precise prototype vector based on the representative samples of the support set for a class. Here we use the transformer architecture [47] for the prototype generation module, the training process is shown in Algorithm. 2.

Algorithm 2: Stage 2 training pipeline.

Input: frozen Encoder $E(\cdot)$, $\mathcal{D}_0$, PGM with randomly initial parameters
Output: trained PGM

for $\mathcal{S}_i$ *in* $\mathcal{D}_0$ **do**
  $\mu_i \leftarrow \frac{1}{|\mathcal{S}_i|}\sum_{x\in\mathcal{S}_i} E(x)$;
  $\sigma_i \leftarrow (\frac{1}{|\mathcal{S}_i|}\sum_{x\in\mathcal{S}_i}\langle\mu_i, E(x)\rangle)^\gamma$;
end
for $\mathcal{S}_i$ *in* $\mathcal{D}_0$ **do**
  for x *in* $\mathcal{S}_i$ **do**
    assign x with label 1 if $p(y=i|x) \geq \epsilon$ else 0;
  end
end
for *epoch in* $\{1, 2, \cdots, total\_training\_epochs\}$ **do**
  $\mathcal{C} \leftarrow$ randomly sample N_c classes from base classes ;
  $\mathcal{L}_P \leftarrow 0$;
  for c *in* $\mathcal{C}$ **do**
    $x \leftarrow$ representation-balanced randomly sample K samples from $\mathcal{S}_c$;
    $\mu \leftarrow$ PGM(x) ;
    $\mathcal{L}_P \leftarrow \mathcal{L}_P + 1 - \langle\mu_c, \mu\rangle$;
  end
  optimize PGM with $\mathcal{L}_{cos}$;
end

During the episodic training, at each task, we select K samples of a class to generate a prototype by using the PGM. This training method can simulate the condition of learning at few-shot incremental session. To improve the robustness of PGM for weakly representative samples, we use a representation-balanced select strategy to select samples from base classes. In our implementation for Algorithm. 2, we set the K is 5 and the episodic classes N_c is 5. At training phase for PGM, if the current epoch is less than half of the total training epochs, we select 4 representative samples and 1 weakly representative sample as the samples set of a class. And if the current epoch is more than half of the total training epochs, we select 3 representative samples

Table 1: Comparison with the state-of-the-art on CIFAR100 dataset. The * denotes result report in corresponding paper.

Method	Average class-wise accuracy (%) ↑									PD ↓
	0	1	2	3	4	5	6	7	8	
Ft-CNN*	64.10	39.61	15.37	9.80	6.67	3.80	3.70	3.14	2.65	61.45
iCaRL*[36]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	50.37
EEIL*[3]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	48.25
Rebalancing*[14]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	50.56
TOPIC*[46]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	34.73
Decoupled-Cosine*[49]	74.55	67.43	63.63	59.55	56.11	53.80	51.68	49.67	47.68	26.87
CEC*[59]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	23.93
Fact*[61]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	22.50
DBONet	**77.81**	**73.62**	**71.04**	**66.29**	**63.52**	**61.01**	**58.37**	**56.89**	**55.78**	**22.03**

Table 2: Comparison with the state-of-the-art on *mini*ImageNet dataset.

Method	Average class-wise accuracy (%) ↑									PD ↓
	0	1	2	3	4	5	6	7	8	
Ft-CNN*	61.31	27.22	16.37	6.08	2.54	1.56	1.93	2.60	1.40	59.91
iCaRL*[36]	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	44.10
EEIL*[3]	61.31	46.58	44.00	37.29	33.14	27.12	24.10	21.57	19.58	41.73
Rebalancing*[14]	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17	47.14
TOPIC*[46]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	36.89
Decoupled-Cosine*[49]	70.37	65.45	61.41	58.00	54.81	51.89	49.10	47.27	45.63	24.74
CEC*[59]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	24.37
Fact*[61]	72.56	69.63	66.38	62.77	60.6	57.33	54.34	52.16	50.49	22.07
DBONet	**74.53**	**71.55**	**68.57**	**65.72**	**63.08**	**60.64**	**57.83**	**55.21**	**53.82**	**20.71**

Table 3: Comparison with the state-of-the-art on CUB200 dataset.

Method	Average class-wise accuracy (%) ↑											PD ↓
	0	1	2	3	4	5	6	7	8	9	10	
Ft-CNN*	68.68	43.70	25.05	17.72	18.08	16.95	15.10	10.06	8.93	8.93	8.47	60.21
iCaRL*[36]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	47.52
EEIL*[3]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	46.57
Rebalancing*[14]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	48.81
TOPIC*[46]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.26	42.40
Decoupled-Cosine*[49]	75.52	70.95	66.46	61.20	60.86	56.88	55.40	53.49	51.94	50.93	49.31	26.21
CEC*[59]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	23.57
Fact*[61]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	**18.96**
DBONet	**78.66**	**75.53**	**72.72**	**69.45**	**67.21**	**65.15**	**63.03**	**61.77**	**59.77**	**59.01**	**57.42**	21.24

and 2 weakly representative samples to further exploit the representation capture ability of the PGM.

And then we use the $\mathcal{L}_P$ mentioned in Algorithm. 2 to constrain the prototype to groundtruth, and the groundtruth is set as the average feature vector of all samples of this class. Through above training process, we can train a PGM to obtain a accurate prototype of one class as much as possible with the condition of limited samples. We empirically set the hyperparameter γ as 0.25 and the representativeness threshold ϵ as 0.03.

4. Experiments

4.1. Datasets and implementation details

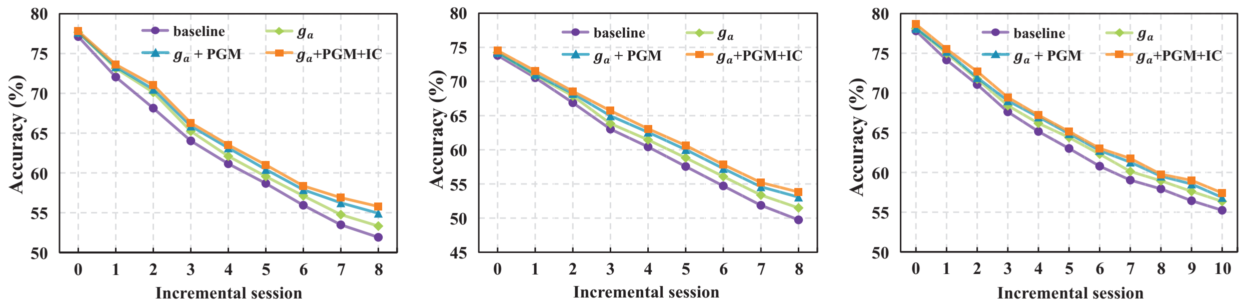
We follow TOPIC [46] and use datasets CIFAR100 [18], *mini*ImageNet [7] and Caltech-UCSD Birds-200-2011(CUB200) [51] to evaluate our methods. CIFAR100 has 100 categories of images. Each category has 500 training images and 100 test images. Size of each image is 32 $\times$32 pixels. *mini*ImageNet also has 60,000 images with 100 classes. And per class has 500 training images and 100 testing images. Size of each image is 84 $\times$84 pixels. CUB200 is a fine-grained image classification benchmark, which consists of 200 different species of birds. It contains 5994 images for training and 5794 images for testing. Each image has a size of 224 $\times$ 224 pixels. In FSCIL task, we follow the setting as the same as previous

Table 4: Average class-wise accuracy of base and novel categories.

Method	Average class-wise accuracy (%) ↑					
	CIFAR100		*mini*ImageNet		CUB200	
	base	novel	base	novel	base	novel
Decoupled-Cosine	72.2	10.9	69.36	10.035	76.1	9.125
CEC	71.7	15.3	69.82	14.345	76.34	16.19
Fact	75.43	17.43	71.08	16.655	78.24	23.265
DBONet	**77.11**	**23.785**	**73.74**	**22.59**	**79.23**	**25.705**

Table 5: Ablation study on three benchmark datasets.

Dataset	Method	Average class-wise accuracy (%) ↑											PD↓
		0	1	2	3	4	5	6	7	8	9	10	
CIFAR100	baseline	77.13	72.05	68.14	64.03	61.15	58.68	55.94	53.48	51.91	×	×	25.22
	g_a	77.67	73.14	70.12	65.24	62.1	59.56	57.12	54.76	53.32	×	×	24.35
	g_a+PGM	77.67	73.33	70.5	65.89	63.14	60.43	57.89	56.24	54.93	×	×	22.74
	g_a+PGM+IC	77.81	73.62	71.04	66.29	63.52	61.01	58.37	56.89	55.78	×	×	22.03
*mini*ImageNet	baseline	73.82	70.56	66.89	63.03	60.43	57.56	54.71	51.87	49.74	×	×	24.08
	g_a	74.24	70.92	67.88	63.84	61.46	58.83	56.09	53.37	51.49	×	×	22.75
	g_a+PGM	74.24	71.04	68.24	64.97	62.57	60.03	57.26	54.58	53.07	×	×	21.17
	g_a+PGM+IC	74.53	71.55	68.57	65.72	63.08	60.64	57.83	55.21	53.82	×	×	20.71
CUB200	baseline	77.82	74.16	71.07	67.63	65.16	63.02	60.79	59.04	56.93	55.44	54.24	23.58
	g_a	78.13	74.98	71.76	68.42	66.19	64.4	62.31	60.13	58.5	57.62	55.96	21.17
	g_a+PGM	78.13	75.23	71.93	69.02	66.98	64.88	62.76	61.29	59.53	58.54	56.85	21.28
	g_a+PGM+IC	78.66	75.53	72.72	69.45	67.21	65.15	63.03	61.77	59.77	59.01	57.42	21.24



(a) CIFAR100 (b) *mini*ImageNet (c) CUB200

Figure 4: Ablation study on three benchmarks. The results show that our proposed module has incremental improvements for FSCIL task. Compared to baseline model, we improve by 3.87%, 4.08%, 3.18% on the last session class-wise average accuracy for CIFAR100, *mini*ImageNet and CUB200, respectively. And we achieve 3.19%, 3.37%, 2.34% decrease on performance drop for CIFAR100, *mini*ImageNet and CUB200.

works [46, 59, 61]. For CIFAR100, there are 60 categories as the base data and the remaining 40 categories as the novel data. The model learns novel data in the manner of $5way \times 5shot$, with a total of 8 sessions. And for *mini*ImageNet, we divide it into 60 and 40 categories, with the same setting of CIFAR100. For CUB200, we divide it into 100 and 100 categories, respectively. The model learns novel data in the manner of $10way \times 5shot$, with a total of 10 sessions. We use ResNet20 as the backbone for the CIFAR100 dataset, and the corresponding pre-encoder $h(\cdot)$ structure is consistent with the first two residual layers of ResNet20, $g_s(\cdot)$ and $g_a(\cdot)$ structure are consistent with the last layer of ResNet. We train 100 epochs with an initial learning rate of 0.01. We adopt ResNet18 as the backbone architecture for the *mini*ImageNet dataset. The pre-encoder function $h(\cdot)$ aligns structurally with the first three layers of ResNet18, while the structures of $g_s(\cdot)$ and $g_a(\cdot)$ are analogous to the last layer of ResNet18. The training process consists of 100 epochs, initialized with a learning rate of 0.01. For the CUB200 dataset, we maintain network consistency with the architecture used for *mini*ImageNet. The training process spans 100 epochs, commencing with an initial learning rate of 0.001. Our implementation is based on the PyTorch library, and we employ SGD with momentum as the optimization algorithm, alongside milestones as the scheduler.

4.2. Comparison with the State-of-the-art Methods

We report the performance over benchmark datasets CIFAR100, *mini*ImageNet and CUB200 in Tab. 1, Tab. 2 and Tab. 3. Compared with current SOTA results, we obtained 3.68%, 3.33% and 0.86% improvements at last session class-wise average accuracy on CIFAR100, *mini*ImageNet and CUB200 datasets, respectively. In order to maintain consistency in model capacity, we halve the convolution kernel width of the last layer of ResNet, so as not to bring about excess parameters and computations. The last layer width of ResNet20 for CIFAR100 is 64, which we reduced to 32, *mini*ImageNet and CUB200 use ResNet18, and the width of the last layer is 512, which we changed it to 256. The performance drop is the result that the session 0 accuracy subtract the last session accuracy. Our performance improvement is also reflected in the term of performance drop , which means that our classification accuracy improvement is not only dependent on the base classes.

We also calculate the average accuracy of the base class and the novel class separately, and we can see that our model mainly improves classification performance on the novel classes compared with base classes. Compared to the Fact [61], we have improved the classification accuracy of the base classes by 1-2%, and the accuracy of the novel classes can be improved by 2-6%, which is primarily attributed to our innovative approach involving feature augmentation training and prototype vector optimization. Compared to previous freezing backbone methods, our proposed method achieves better performance balance on base classes and novel classes.

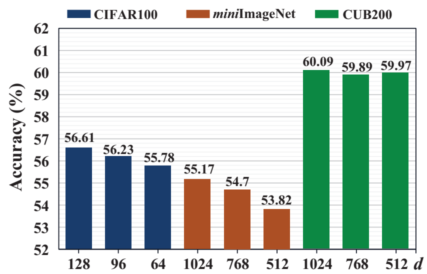


Figure 5: Channel width of the last layer (g_a, g_s) study on three benchmark datasets.

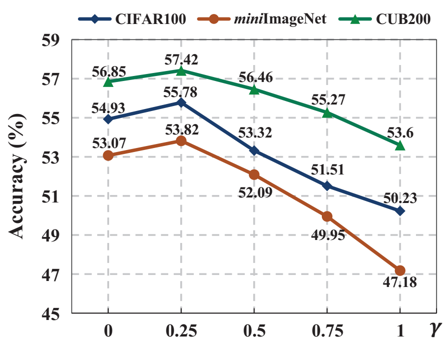
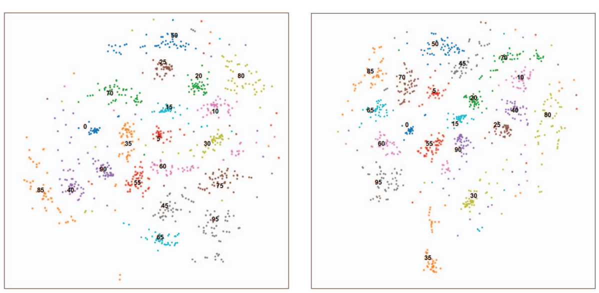


Figure 6: Performance with different hyperparameters γ on three benchmark datasets.



(a) Features distribution output by conventional network structure. (b) Features distribution output by augmentation feature extractor.

Figure 7: t-SNE visualization on *mini*ImageNet dataset.

4.3. Ablation study

On the basis of building the baseline method, we gradually add the required modules. The results are shown in Tab. 5 and Fig. 4. g_a in Tab. 5 represents the use of augmented features and $BICELoss$ in Eq. 4 to train the g_a extractor module, while baseline uses conventional samples and cross entropy loss for training. PGM in Tab. 5 denotes the use of prototype generation module to generate prototype vectors. Finally, the IC in Tab. 5 indicates the classifier based on cosine variance is added. To assess the effectiveness of our method, we conduct ablation experiments on all three datasets. To ensure a fair comparison, we train g_a without utilizing manifold mixup features by substituting it with a convolutional layer of the same structure. We train the alternative model using cross-entropy loss and then concatenate the feature vectors obtained from the last two layer modules. This demonstrates the contribution of manifold mixup features in our overall approach.

4.4. Analysis

To investigate the impact of prototype dimension, we conduct an experiment with different output dimensions of two feature extractors across three datasets. The results are presented in Fig. 5. Notably, as the number of dimension increased, our model exhibited additional performance improvements. In Fig. 5, the variable d represents the sum of the last layer width of the feature extractors $g_a(\cdot)$ and $g_s(\cdot)$.

We conduct a comparative experiment (Fig. 6) to investigate the impact of parameter γ. Setting γ to 1 result in a significant decrease in accuracy, indicating that excessive introduction of intra-class variance information confuses the embedded spatial classification boundary. Conversely, with γ set to 0.25, we achieve the highest classification accuracy in the final parameter session, reaching a better equilibrium state.

As shown in the Fig. 7, we visualize the feature vectors of test set samples of categories 0, 5, 10, $\cdots$, 95 (we treat the first category as 0) by using the t-SNE algorithm. In order to facilitate the presentation of the results, we only select 50 samples from each category. Fig. 7 (a) shows the output result of the conventional network structure introduced in Sec. 4.3, and Fig. 7 (b) presents the output result of the augmentation feature extractor. From the results we can observe that the distribution of samples of novel class are more dispersed compared with most of the base classes. The features by the augmentation feature extractor are often more compact, especially on categories 65, 70, 75 and 95.

5. Conclusion

In this paper, we propose a framework based on decision boundary optimization to apply in few-shot class-incremental learning, and use the manifold mixup for feature augmentation to further improve the feature extraction ability. The PGM produces much better class prototype to reduce the classification error caused by the decision boundary shift. The classifier considering the intra-class variance further obtains a more accurate decision boundary in the latent space. Our extensive experiments show that proposed method achieves SOTA performance.

Acknowledgements. This work was supported by National Natural Science Foundation of China under Grants 62072021.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision*, pages 139–154, 2018.

[2] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 583–592, 2019.

[3] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision*, pages 233–248, 2018.

[4] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2534–2543, 2021.

[6] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafscil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14166–14175, 2022.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[8] Jiankang Deng, Jia Guo, Jing Yang, Alexandros Lattas, and Stefanos Zafeiriou. Variational prototype learning for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11906–11915, 2021.

[9] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1255–1263, 2021.

[10] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. DMT: dynamic mutual training for semi-supervised learning. *Pattern Recognit.*, 130:108777, 2022.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9057–9067, 2022.

[14] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[15] Zhi Hou, Baosheng Yu, and Dacheng Tao. Batchformer: Learning to explore sample relationships for robust representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7256–7266, 2022.

[16] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[17] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *Proceedings of the 34th International Conference on Machine Learning deep learning workshop (ICMLW)*, volume 2. Lille, 2015.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Anna Kukleva, Hilde Kuehne, and Bernt Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9020–9029, 2021.

[20] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in Neural Information Processing Systems*, 30, 2017.

[21] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021.

[22] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2019.

[23] Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *CoRR*, abs/2304.09854, 2023.

[24] Cheng-Lin Liu and Masaki Nakagawa. Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition. *Pattern Recognition*, 34(3):601–615, 2001.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[26] Viktor Losing, Barbara Hammer, and Heiko Wersing. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274, 2018.

[27] Shuchang Lyu, Ting-Bing Xu, and Guangliang Cheng. Embedded knowledge distillation in depth-level dynamic neural network. *CoRR*, abs/2103.00793, 2021.

[28] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

[29] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[30] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[31] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR, 2018.

[32] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[33] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019.

[34] Archit Parnami and Minwoo Lee. Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint arXiv:2203.04291*, 2022.

[35] Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1320–1328, 2017.

[36] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[37] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard Zemel. Incremental few-shot learning with attention attractor networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[39] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[40] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30, 2017.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[42] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[43] Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 2023.

[44] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[45] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[46] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[48] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. pages 6438–6447. PMLR, 2019.

[49] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[50] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[51] Mita T. Wah C. Schroff F. Belongie S. Perona P Welinder P., Branson S. "caltech-ucsd birds 200". Technical report, California Institute of Technology. CNS-TR-2010-001., 2010.

[52] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[53] Jingyi Xu and Hieu Le. Generating representative samples for few-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9003–9013, 2022.

[54] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3474–3482, 2018.

[55] Sung Whan Yoon, Do-Yeon Kim, Jun Seo, and Jaekyun Moon. Xtarnet: Learning to extract task-adaptive representa-

tion for incremental few-shot learning. In *International Conference on Machine Learning*, pages 10852–10860. PMLR, 2020.

[56] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pages 7115–7123. PMLR, 2019.

[57] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13834–13844, 2021.

[58] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[59] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12455–12464, 2021.

[60] Qi Zhao, Shuchang Lyu, Lijiang Chen, Binghao Liu, Ting-Bing Xu, Guangliang Cheng, and Wenquan Feng. Learn by oneself: Exploiting weight-sharing potential in knowledge distillation guided ensemble network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[61] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9046–9056, 2022.

[62] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023.

[63] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6801–6810, 2021.