Class-Aware Patch Embedding Adaptation for Few-Shot Image Classification

Fusheng Hao[1,2] Fengxiang He[3] Liu Liu[4] Fuxiang Wu[1,2] Dacheng Tao[4] Jun Cheng[1,2*]

[1]Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems,
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China

[2]The Chinese University of Hong Kong, Hong Kong, China

[3]AIAI, School of Informatics, University of Edinburgh, United Kingdom

[4]School of Computer Science, Faculty of Engineering, The University of Sydney, Australia

Abstract

"A picture is worth a thousand words", significantly beyond mere a categorization. Accompanied by that, many patches of the image could have completely irrelevant meanings with the categorization if they were independently observed. This could significantly reduce the efficiency of a large family of few-shot learning algorithms, which have limited data and highly rely on the comparison of image patches. To address this issue, we propose a Class-aware Patch Embedding Adaptation (CPEA) method to learn "class-aware embeddings" of the image patches. The key idea of CPEA is to integrate patch embeddings with class-aware embeddings to make them class-relevant. Furthermore, we define a dense score matrix between class-relevant patch embeddings across images, based on which the degree of similarity between paired images is quantified. Visualization results show that CPEA concentrates patch embeddings by class, thus making them class-relevant. Extensive experiments on four benchmark datasets, miniImageNet, tieredImageNet, CIFAR-FS, and FC-100, indicate that our CPEA significantly outperforms the existing state-of-the-art methods. The source code is available at https://github.com/FushengHao/CPEA.

1. Introduction

Real-world images are usually composed of many different entities, *e.g.*, two oxen grazing surrounded by a barn, a fence and trees as shown in Figure 1. Assigning a single annotation to each image that corresponds to only one type of entity is a common practice to construct computer vision datasets, *e.g.*, CIFAR [33] and ImageNet [54]. Such an annotation can only describe part of an image' contents. This is acceptable in many classification scenarios, because the interference caused by other image contents can be mitigated by the use of a large number of labeled images. Specifically, since each class contains a sufficient number of labeled images that vary greatly within the class and the corresponding entities always appear in these images, deep models trained on such data tend to pay attention to the frequently occurring class-relevant entities (*e.g.*, "ox" in Figure 1) while ignoring other irrelevant ones, especially those that frequently appear across classes [26].

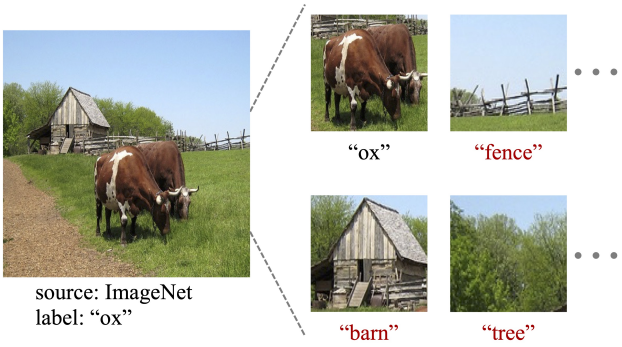


Figure 1. Illustration of multiple entities from different classes simultaneously existing in a real-world image. Despite being annotated as "ox", the image contains entities of other classes, such as "fence", "barn", "tree", *etc*. The core idea of CPEA is to learn "class-aware embeddings" of the image patches.

Big challenges, however, arise in the context of few-shot image classification, in which approaches are expected to correctly identify new classes that are disjoint with the training classes during the test phase, given only a few (*e.g.*, one or five) labeled images for each of these new classes. The challenges are as follows: 1) Due to the scarcity of labeled images of new classes and the extremely limited number of class-relevant entities, it is very difficult for a model to identify which entity determines the class of an image. 2) Entities contained in the training images but not covered by the training classes may happen to be the ones expected to be covered by the new classes at test time, which would introduce ambiguity. 3) Specific patterns learned during the training phase may be overemphasized, but they may not

*Corresponding author (email: jun.cheng@siat.ac.cn).

be relevant to the new classes seen at test time, resulting in supervision collapse [14] and limited generalizability.

A promising solution is to align semantically-relevant regions [24, 27, 70, 14, 26]. SAML [24] proposes to use the activation-based attention to highlight semantically-relevant regions while suppressing others. CAN [27] performs cross-attention between class prototypes and query feature maps to highlight class-relevant regions. DeepEMD [70] looks for the aligned regions by minimizing the earth movers' distance. CTX [14] uses a Transformer-style attention mechanism to perform the spatial and semantic alignment and mitigates supervision collapse by incorporating self-supervised learning in training. FewTURE [26] determines the most informative regions via online optimization and then uses them to reweigh patch correspondences. While these methods have shown great potential in eliminating interference and tackling supervision collapse, there still exist crucial drawbacks. Firstly, aligned semantically-relevant regions are not always beneficial for similarity measure, such as those that are irrelevant to the class of interest. Secondly, the scarcity of labeled images of new classes and the extremely limited number of class-relevant entities makes it difficult to deal with the inaccurate localization and alignment induced by large intra-class variation and background clutter in real-world images.

In this paper, we deal with the above challenges from a new perspective and propose a Class-aware Patch Embedding Adaptation (CPEA) method that can eliminate the interference of single-label annotations without aligning semantically-relevant regions while avoiding supervision collapse. Specifically, we employ self-supervision pretraining instead of the supervised one to avoid supervision collapse, with Masked Image Modelling [76] as a pretext task, which yields semantically meaningful patch embeddings. Since the patch embeddings may be irrelevant to class of interest, this leads to the need for aligning semantically-relevant patches. We avoid the need for localization and alignment mechanisms by making patch embeddings class-relevant. To this end, we introduce a class-agnostic embedding and feed it into the transformer to interact with patch embeddings to make it class-aware. Then, patch embeddings are adapted with the class-aware embeddings to make them class-relevant, which alleviates the scarcity of labeled images by increasing their amount. Furthermore, we define a dense score matrix between class-relevant patch embeddings across images, based on which the degree of similarity between paired images is quantified.

Our main contributions are summarized as follows: 1) We deal with the interference caused by single-label annotations in few-shot settings from a new perspective and demonstrate that the interference can be successfully mitigated without the need for localization and alignment mechanisms. 2) We propose the CPEA, a novel method that makes patch embeddings class-relevant and measures the similarity between class-relevant patch embeddings across images in a dense manner, which improves transferability. 3) Visualizations show that our CPEA makes patch embeddings class-relevant. Extensive experiments are conducted on four popular benchmark datasets and the results indicate that our CPEA achieves superior performance over the state-of-the-art methods.

2. Related work

Few-shot image classification. Few-shot image classification has recently attracted much attention because of its great application prospects in real-world scenarios. Existing methods can be roughly categorized into two groups. The first group is optimization-based methods. They learn a meta-learner, which can optimize a learner in a few steps given the few labeled images. For example, the meta-learner of MAML [18] and Reptile [49] output a set of good model initializations that can be adapted to a specific few-shot classification task in a few gradient steps. To avoid the large computational overhead caused by the need to update all model parameters during inference, CAVIA [78] and LEO [55] propose to perform meta-learning in a low-dimensional representation space. Then, it is found that both the choice of network architectures and the design of meta-learners have a severe impact on the performance and efficiency, which motivates the exploration of various variants of meta-learning methods [13, 50, 29].

The second group is metric-based methods. They focus on learning a feature space suitable for all few-shot classification tasks, in which an appropriate distance function is used for similarity measure. For example, MatchingNet [61] constructs a feature space based on neural networks, where the cosine distance is used for similarity measure. ProtoNet [57] learns a feature space, where the Euclidean distance is used for similarity measure. Then, it is found that the choice of network architectures, the choice of distance functions, the choice of prototypes, and the choice of training strategies all have a severe impact on the performance and efficiency, which motivates the exploration of various variants of metric-based methods [46, 47, 56, 69, 36, 62, 4, 5, 19]. Recently, local feature-based methods have achieved great success in addressing the challenging few-shot image classification problem. One line of such methods directly treat local features as image representations [58, 41, 12, 11] and the other line is to align the semantically relevant local features [24, 27, 70, 14, 26, 23]. Our method makes local features class-relevant, thus avoiding the need for localization and alignment mechanisms and increasing the usability of local features.

Self-supervision in few-shot image classification. Although self-supervision methods have achieved great success on large-scale image datasets, their potential has not

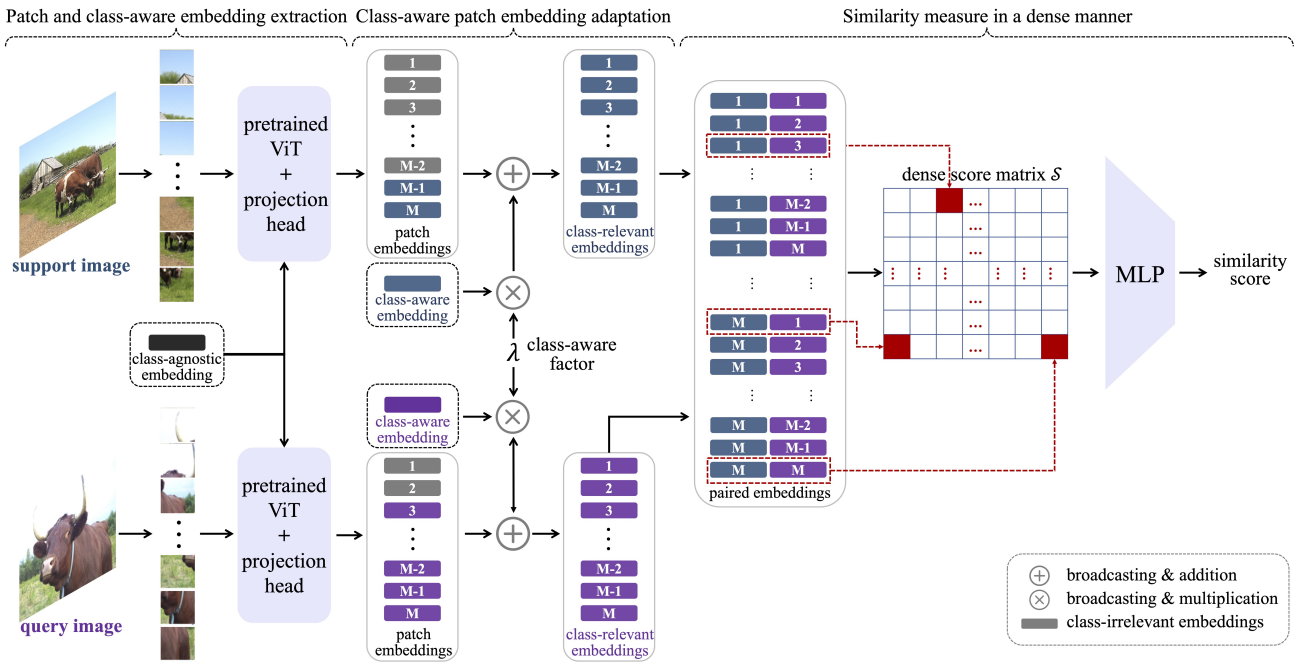


Figure 2. Overview of the proposed class-aware patch embedding adaptation method. The ViT is pretrained with Masked Image Modelling [76] as a pretext task. The class-agnostic embedding is a learnable embedding and made class-aware by constantly interacting with patch embeddings in the ViT. Then, the patch embeddings at the output are adapted with the class-aware embeddings to make them class-relevant. Finally, the similarity score between paired images is obtained by aggregating the dense score matrix. Note that we distinguish patch embeddings by numbers and M denotes the number of patch embeddings.

been fully explored in the few-shot settings. Recent works have shown that they can help improve the generalization ability of learned models [48, 14, 26, 43, 67]. For example, S2M2 [48] integrates two self-supervision methods, *i.e.*, rotation [22] and exemplars [16], with the standard supervised learning to improve the generalization ability of output features of pre-trained models. CTX [14] integrates SimCLR [8] into the episodic training strategy to improve the generalization ability of the learned model. FewTURE [26] uses Masked Image Modelling [3, 38, 25, 76] as a pretext task to pretrain the Vision Transformer (ViT) [15] on small-scale datasets, resulting in features with strong generalization ability.

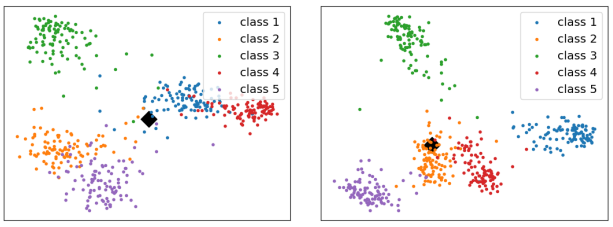
ViT in few-shot image classification. Due to their ability to build long-range dependencies between image patches, ViTs have achieved great success in many application fields of computer vision such as image classification [15, 42] and object detection [42]. However, they rely more heavily on large-scale image datasets than Convolutional Neural Networks (CNNs) due to the lack of the convolutional inductive bias [40]. For example, ViTs have to learn from images the locality and the translation invariance embedded in the CNN design. This data-hungry nature makes it difficult for ViTs to be used as a whole in the few-shot settings, or only a very tiny part of ViTs (*e.g.*, s single Transformer head) can be used in conjunction with CNNs [14, 69, 34]. Recently, FewTURE [26] demonstrates that a fully ViT-based architecture can be successfully generalized on small-scale image datasets.

3. Method

We first formulate the definition of few-shot image classification and then present an overview of the whole pipeline. Next, we detail the class-aware patch embedding adaptation and the dense similarity measure. Finally, we describe the training and inference strategies.

3.1. Problem definition

Few-shot image classification focuses on generalizing the knowledge learned on the training classes $\mathcal{C}_{train}$ to the unseen test classes $\mathcal{C}_{test}$, *i.e.*, $\mathcal{C}_{train} \cap \mathcal{C}_{test} = \emptyset$, given only a few labeled images for each of these test classes. We follow the common practice of previous works [61] to formulate the N-way K-shot classification task in an episodic manner, where N denotes the number of classes and K denotes the number of labeled images contained in each class. An episode is composed of a support set $\mathcal{X}_s = \{(x_i, y_i)\}_{i=1}^{NK}$ and a query set $\mathcal{X}_q = \{(x_i, y_i)\}_{i=1}^{NQ}$, where Q denotes the number of test images contained in each class. The query set is used to evaluate the performance of a model on the few-shot classification task defined by the support set. Our goal is to learn a model on training classes that generalizes well on episodes randomly sampled from the unseen test classes within the *inductive* framework.



(a) (b)

Figure 3. Class-aware embedding visualization. Two different sampling results are given in (a) and (b), respectively, with 100 class-aware embeddings per class. The class-agnostic embedding is denoted by the black "diamond". After interacting with images from different classes, the output states of class-agnostic embedding are class-aware.

3.2. Overview

Figure 2 shows the pipeline of the proposed method. We decompose an image into patches, embed them using linear projection, add position embeddings to the resulting patch embeddings, and prepend a class-agnostic embedding to the sequence of patch embeddings, which are then fed into a standard Transformer encoder. Before performing class-aware patch embedding adaptation, we add a projection head to further transform the embeddings at the output of the Transformer encoder, with aim of increasing their adaptability to the few-shot classification task. Afterwards, the class-aware embeddings are used to adapt patch embeddings to make them class-relevant. To quantify the degree of similarity between paired images, we define a dense score matrix between class-relevant patch embeddings, based on which a MLP is used to aggregate the dense score matrix into a single similarity score.

3.3. Class-aware patch embedding adaptation

Self-supervision pretraining. Single-label annotations cause supervision collapse in few-shot settings, which highlights certain patterns that are useful for distinguishing training classes, rather than those that have good transferability [26]. To address this issue, we employ self-supervision pretraining instead of the supervised one to pretrain the ViT. Specifically, we decompose an image into patches, randomly mask some patches, encode these patches with a ViT and reconstruct the masked patches, where Masked Image Modeling (MIM) [3, 76] is used as a pretext task. The reasons for this are two-fold: 1) Labeled images are scarce in few-shot settings and a considerable amount of patch embeddings can be obtained in a single feedforward. To facilitate making patch embeddings class-relevant, it is necessary to yield semantically meaningful patch embeddings. 2) Different from the self-supervision approaches [7, 9] that focus on self-similarity of global representations between images from different views, MIM aims to reconstruct the masked patches and build an understanding of the structure and content of an image, rather than learning patterns that are mainly useful for training classes, which improves transferability.

Class-aware embedding. After pretraining, a considerable number of patch embeddings can be obtained for an input image in a single feedforward. However, the semantics of these patch embeddings may be irrelevant to the class of interest. Therefore, it is necessary to explore approaches that make the semantics of the patch embeddings relevant to class of interest, in order to treat them as image representations. We propose to adapt patch embeddings with class-aware embeddings to make them class-relevant. It is to be noted that class token is a learnable embedding and plays a key role in ViTs, whose state at the output is served as image representations [15]. Here, two often overlooked facts need to be highlighted: 1) Before being input into a ViT, class token is class-agnostic. 2) By constantly interacting with patch embeddings in the ViT, class token becomes class-aware and its final output state is treated as representations of the corresponding image. Figure 3 shows the class-aware embedding visualization results of images from different classes, which demonstrates that after interacting with images from different classes, the output states of class-agnostic embedding are class-aware. These facts suggest that the class-agnostic embedding may have a strong generalization ability, which motivate us to introduce the class-agnostic embedding and make it class-aware in a similar way[1].

Patch embedding adaptation. Given an image, its patch embeddings and class-aware embedding can be obtained simultaneously. Two facts need to be highlighted: 1) The semantics of these patch embeddings may be irrelevant to the class to which the image belongs and the spatial location of patch embeddings relevant to the class of interest is unknown in advance. This inspires the exploration of aligning semantically-relevant regions. Due to the inaccurate localization and alignment induced by large intra-class variation and background clutter, the results of semantic alignment are far from satisfactory [24]. 2) The number of patch embeddings is usually considerable. Several methods have directly treated patch embeddings as image representations to alleviate the scarcity of labeled images [68, 37]. Although promising performance improvements have been achieved, they suffer from irrelevant patch embeddings. We deal with the above issues from a new perspective of adapting patch embeddings with the class-aware embedding to make them class-relevant, which can be formulated as follows:

$$\bar{z}_i^o = z_i^o + \lambda z_{class}^o \,, \tag{1}$$

where $\bar{z}_i^o$ and z_i^o respectively denote the i-th adapted patch

[1]The connection between class-agnostic embedding and class-aware embedding is that they are class tokens at different stages of the model forward pass.

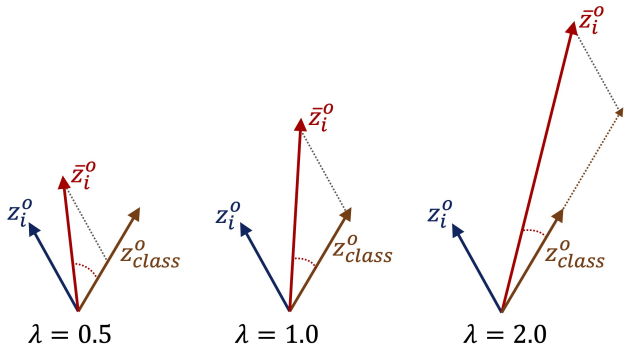


Figure 4. Illustration of how the class-aware factor λ controls the magnitude of class-awareness. Considering that the similarity measure used is insensitive to the norm of embeddings, the larger the value of λ, the smaller the angle between the class-aware embedding and the adapted patch embedding, and the more relevant the adapted patch embedding is to the class of interest.

embedding and the i-th original patch embedding at the output of the projection head, z_{class}^o denotes the class-aware embedding at the output of the projection head, and $\lambda > 0$ denotes the class-aware factor that controls the magnitude of the correlation. The reason behind this design is as follows. Although the labels of the patch embeddings, *i.e.*, $y^{(z_i^o)}$, are unknown, the class-aware embedding's, *i.e.*, $y^{(z_{class}^o)}$, is known. Therefore, the labels of the adapted patch embeddings, *i.e.*, $y^{(\bar{z}_i^o)}$, can be formulated as follows:

$$y^{(\bar{z}_i^o)} = y^{(z_i^o)} + \lambda y^{(z_{class}^o)} . \tag{2}$$

This practice has achieved great success in *mixup* [72]. The difference is that two known classes are mixed in *mixup* while a known class and an unknown class are mixed in our method. Note that both adapted patch embeddings and labels are $l2$ normalized before using, meaning that the larger the value of λ, the more relevant the adapted patch embeddings are to the class of interest, as shown in Figure 4.

3.4. Dense similarity measure

Without loss of generality, we take a support image and a query image as an example to show how to measure the similarity between them. When designing the similarity measure, two points need to be considered: 1) Since the number of class-relevant patch embeddings is considerable, it would be better to use them all at the same time, in order to impose a strong constraint on the whole pipeline. 2) It would be better to design a similarity measure that requires no domain expertise, in order to reduce the difficulty of deployment in practical applications. To this end, we define a dense score matrix $\mathcal{S}$ whose element is a score between adapted patch embeddings across images, which can be formulated as follows:

$$\mathcal{S}_{ij} = d(\bar{z}_i^{o(S)}, \bar{z}_i^{o(Q)})^2 , \tag{3}$$

where $\bar{z}_i^{o(S)}$ and $\bar{z}_i^{o(Q)}$ respectively denote the adapted patch embeddings of the support image and the query image, and $d(\cdot,\cdot)$ denotes the cosine similarity. Then, we flatten the dense score matrix and directly input it into a MLP to output a similarity score. The reasons for these choices are as follows: 1) All class-relevant patch embeddings are used, which helps alleviate the overfitting issue induced by the scarcity of labeled images. 2) The difficulty of choosing the right function from massive suitable functions is bypassed. 3) The cosine function is insensitive to the norm of embeddings, which avoids $l2$ normalization of Eq. (1). 4) The use of the square term can facilitate the independence of the adapted embeddings between different classes.

3.5. Training and inference

Training. There are K labeled images in the n-th support class. After obtaining the similarity scores between the i-th query image and all support images, we can get the similarity score of the i-th query image belonging to the n-th support class as follows:

$$s_{ni} = \sum_{k=1}^{K} s_{nik} , \tag{4}$$

where s_{nik} denotes the similarity score between the i-th query image and the k-th support image in the n-th support class. Then, the probability of the i-th query image belonging to the n-th support class can be calculated as follows:

$$p_{ni} = \frac{\exp(s_{ni})}{\sum_{n=1}^{N} \exp(s_{ni})} . \tag{5}$$

For a given episode, the loss function can be formulated as follows:

$$L = -\frac{1}{NQ} \sum_{i=1}^{NQ} \sum_{n=1}^{N} I(y_i^{(Q)} = n) \log p_{ni} , \tag{6}$$

where $y_i^{(Q)}$ denotes the label of the i-th query image and $I(\cdot)$ is an indicator function that equals one if its arguments are true and zero otherwise. All the learnable weights involved in our method are finetuned by minimizing Eq. (6) using episodes randomly sampled from training classes.

Inference. Given an episode sampled from the unseen test classes, the probability of a query image belonging to each class can be calculated according to Eq. (5). Then, we assign the label of the class with the maximum probability to the corresponding query image. Note that once finetuned on the training classes, our method does not need any adjustments when generalizing to the unseen test classes, in contrast to FewTURE [26] that needs all images of an episode's support set together with their labels to learn the importance for each individual patch token via online optimization at inference time, resulting in that our method is much faster than FewTURE in terms of inference speed.

Model	Backbone	≈ # Params	miniImageNet		tieredImageNet	
			1-shot	5-shot	1-shot	5-shot
SetFeat [2]	SetFeat-12	12.3 M	68.32±0.62	82.71±0.46	73.63±0.88	87.59±0.57
ProtoNet [57]	ResNet-12	12.4 M	62.29±0.33	79.46±0.48	68.25±0.23	84.01±0.56
FEAT [69]	ResNet-12	12.4 M	66.78±0.20	82.05±0.14	70.80±0.23	84.79±0.16
DeepEMD [70]	ResNet-12	12.4 M	65.91±0.82	82.41±0.56	71.16±0.87	86.03±0.58
IEPT [73]	ResNet-12	12.4 M	67.05±0.44	82.90±0.30	72.24±0.50	86.73±0.34
MELR [17]	ResNet-12	12.4 M	67.40±0.43	83.40±0.28	72.14±0.51	87.01±0.35
FRN [63]	ResNet-12	12.4 M	66.45±0.19	82.83±0.13	72.06±0.22	86.89±0.14
CG [75]	ResNet-12	12.4 M	67.02±0.20	82.32±0.14	71.66±0.23	85.50±0.15
DMF [66]	ResNet-12	12.4 M	67.76±0.46	82.71±0.31	71.89±0.52	85.96±0.35
InfoPatch [39]	ResNet-12	12.4 M	67.67±0.45	82.44±0.31	-	-
BML [77]	ResNet-12	12.4 M	67.04±0.63	83.63±0.29	68.99±0.50	85.49±0.34
CNL [75]	ResNet-12	12.4 M	67.96±0.98	83.36±0.51	73.42±0.95	87.72±0.75
Meta-NVG [71]	ResNet-12	12.4 M	67.14±0.80	83.82±0.51	74.58±0.88	86.73±0.61
PAL [45]	ResNet-12	12.4 M	69.37±0.64	84.40±0.44	72.25±0.72	86.95±0.47
COSOC [44]	ResNet-12	12.4 M	69.28±0.49	85.16±0.42	73.57±0.43	87.57±0.10
Meta DeepBDC [65]	ResNet-12	12.4 M	67.34±0.43	84.46±0.28	72.34±0.49	87.31±0.32
LEO [55]	WRN-28-10	36.5 M	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09
CC+rot [21]	WRN-28-10	36.5 M	62.93±0.45	79.87±0.33	70.53±0.51	84.98±0.36
FEAT [69]	WRN-28-10	36.5 M	65.10±0.20	81.11±0.14	70.41±0.23	84.38±0.16
PSST [10]	WRN-28-10	36.5 M	64.16±0.44	80.64±0.32	-	-
MetaQDA [74]	WRN-28-10	36.5 M	67.83±0.64	84.28±0.69	74.33±0.65	89.56±0.79
OM [52]	WRN-28-10	36.5 M	66.78±0.30	85.29±0.41	71.54±0.29	87.79±0.46
FewTURE [26]	ViT-S/16	22 M	68.02±0.88	84.51±0.53	72.96±0.92	86.43±0.67
CPEA (ours)	ViT-S/16	22 M	**71.97**±0.65	**87.06**±0.38	**76.93**±0.70	**90.12**±0.45

Table 1. Few-shot classification accuracies for the 5-way 1-shot and 5-way 5-shot settings on miniImageNet and tieredImageNet. The average accuracies with 95% confidence interval are reported according to the evaluation protocol.

4. Experiments

We first detail the experimental settings and then compare with the counterparts. Finally, we ablate the key components. It is to be noted that more details regarding datasets and ablation study are provided in the supplementary material.

4.1. Experimental settings

Datasets. We evaluate our method on four popular few-shot classification benchmark datasets, *i.e.*, miniImageNet [61], tieredImageNet [53], CIFAR-FS [6], and FC-100 [51]. We follow the common practice of previous methods [69, 26] to split each dataset into training/validation/test datasets. Their label spaces are disjoint, meaning that the classes seen in the training set will not appear in the validation/test set.

Backbone. We use the ViT-S/16 [15] as the backbone. The reason for this choice is that the number of parameters of ViT-S/16 is comparable to that of the backbones commonly used in the few-shot image classification task. The projection head is a MLP and it has two layers with GELU applied to the first fully-connected layer and LayerNorm applied to the second fully-connected layer. The MLP used to aggregate the dense score matrix has two layers with GELU applied to the first fully-connected layer and its output fully-connected layer is 1-d. The ViT-S/16 takes images with a resolution of 224×224 as input and the patch embeddings are 384-d.

Implementation details. Our training procedure consists of two stages. In the first stage, we pretrain the ViT-S/16 by using the strategy proposed in [76] and sticking to the hyperparameter settings reported. Four A100 40G GPUs are used to pretrain the ViT-S/16 and the total number of training epochs is set to be 1,600. To match our computing resources, the batch size is set to be 512. It is to be noted that only the training set of the corresponding dataset is used for pretraining. In the second stage, we finetune the whole pipeline by minimizing Eq. (6). It is to be noted that only episodes sampled from the training classes are used for finetuning. The optimizer used is Adam [32]. The global initial learning rate is set to be 0.001, which is halved every 500 episodes, and the learning rate of the ViT-S/16 is always kept to be one percent of the global learning rate. The weight decay is set to be 0.001 and the total number of episodes is set to be 10,000.

Evaluation protocol. We report the performance on the 5-way 1-shot and 5-way 5-shot image classification tasks, and the average accuracy of 1,000 episodes randomly sampled from the test classes is taken as the final performance with 15 query images per class.

Model	Backbone	≈ # Params	CIFAR-FS		FC100	
			1-shot	5-shot	1-shot	5-shot
ProtoNet [57]	ResNet-12	12.4 M	-	-	41.54±0.76	57.08±0.76
MetaOpt [35]	ResNet-12	12.4 M	72.00±0.70	84.20±0.50	41.10±0.60	55.50±0.60
MABAS [31]	ResNet-12	12.4 M	73.51±0.92	85.65±0.65	42.31±0.75	58.16±0.78
RFS [59]	ResNet-12	12.4 M	73.90±0.80	86.90±0.50	44.60±0.70	60.90±0.60
BML [77]	ResNet-12	12.4 M	73.45±0.47	88.04±0.33	-	-
CG [20]	ResNet-12	12.4 M	73.00±0.70	85.80±0.50	-	-
Meta-NVG [71]	ResNet-12	12.4 M	74.63±0.91	86.45±0.59	46.40±0.81	61.33±0.71
RENet [30]	ResNet-12	12.4 M	74.51±0.46	86.60±0.32	-	-
TPMN [64]	ResNet-12	12.4 M	75.50±0.90	87.20±0.60	46.93±0.71	63.26±0.74
MixFSL [1]	ResNet-12	12.4 M	-	-	44.89±0.63	60.70±0.60
CC+rot [21]	WRN-28-10	36.5 M	73.62±0.31	86.05±0.22	-	-
PSST [10]	WRN-28-10	36.5 M	77.02±0.38	88.45±0.35	-	-
Meta-QDA [74]	WRN-28-10	36.5 M	75.83±0.88	88.79±0.75	-	-
FewTURE [26]	ViT-S/16	22 M	76.10±0.88	86.14±0.64	46.20±0.79	63.14±0.73
CPEA (ours)	ViT-S/16	22 M	**77.82±0.66**	**88.98±0.45**	**47.24±0.58**	**65.02±0.60**

Table 2. Few-shot classification accuracies for the 5-way 1-shot and 5-way 5-shot settings on CIFAR-FS and FC-100. The average accuracies with 95% confidence interval are reported according to the evaluation protocol.

Projection head	1-shot	5-shot
✓	71.99±0.28	87.01±0.17
×	70.74±0.29	86.69±0.18

Table 3. Impact of the projection head on the few-shot classification performance.

Class-aware factor	1-shot	5-shot
$\lambda = 0.0$	70.40±0.66	85.05±0.43
$\lambda = 0.5$	71.27±0.66	86.40±0.40
$\lambda = 1.0$	71.93±0.66	86.91±0.39
$\lambda = 2.0$	71.97±0.65	87.06±0.38
$\lambda = 4.0$	71.80±0.65	87.13±0.38
$\lambda = 8.0$	71.94±0.65	87.22±0.38
$\lambda = 16.0$	71.85±0.65	87.03±0.38

Table 4. Impact of the class-aware factor on the few-shot classification performance.

4.2. Comparison results

Table 1 and Table 2 show the comparison results for the 5-way 1-shot and 5-way 5-shot settings on four benchmark datasets. It is to be noted that 1) CPEA outperforms the counterparts by a large margin. 2) CPEA exceeds the semantic alignment-based methods by a noticeable margin, including FewTURE [26] and DeepEMD [70]. Moreover, CPEA beats the state-of-the-art semantic alignment-based method in terms of inference speed as shown in Table 9. These observations demonstrate the effectiveness of class-aware patch embedding adaptation.

4.3. Ablation study

The key components of the proposed method are ablated and experiments are conducted on miniImageNet [61].

Projection head. Since the projection head increases the adaptability of embeddings to the few-shot image classification task, the performance is improved by 0.78% on average, as shown in Table 3.

Choices in Eq. (3)	1-shot	5-shot
$d(\cdot,\cdot)^2$	71.97±0.65	87.06±0.38
$d(\cdot,\cdot)$	71.09±0.65	86.64±0.38
$\lvert d(\cdot,\cdot)\rvert$	71.48±0.64	86.49±0.41
$4 \times \lvert d(\cdot,\cdot)\rvert$	71.62±0.66	86.34±0.41

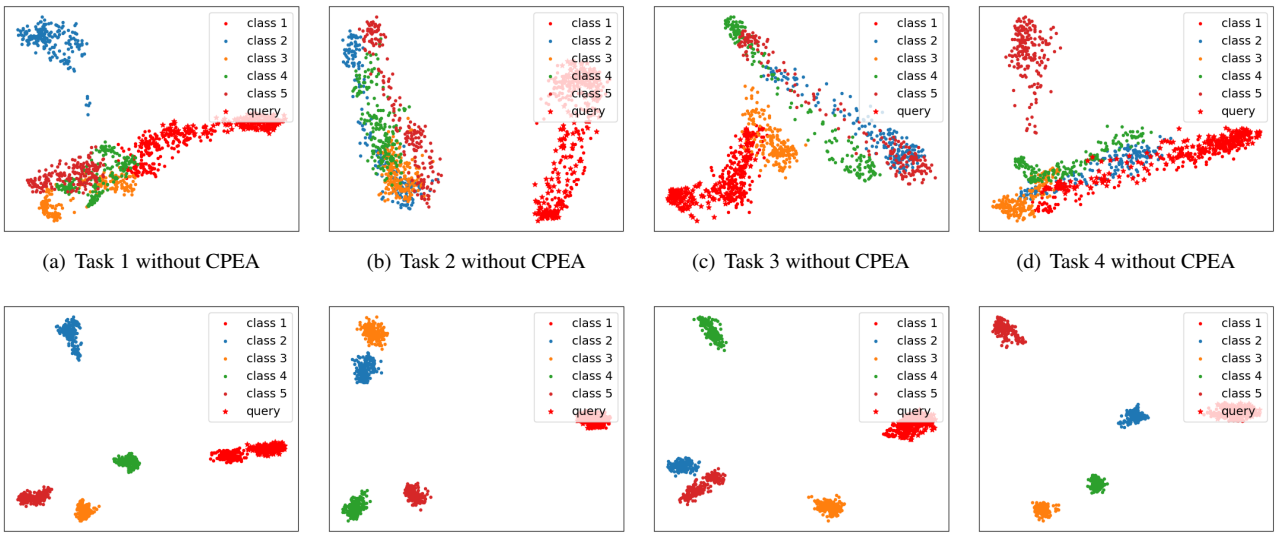
Table 5. Impact of choices in Eq. (3) on the few-shot classification performance.

Pretraining strategy	1-shot	5-shot
DeiT (supervised)	28.58±0.46	36.65±0.48
DINO (self-supervision)	70.65±0.64	85.71±0.40
MIM (self-supervision)	71.97±0.65	87.06±0.38

Table 6. Impact of different pretraining strategies on the few-shot classification performance.

Class-aware factor. $\lambda = 0.0$ means that the patch embeddings are directly treated as image representations. Table 4 shows that adapting patch embeddings to make them class-relevant improves performance by a noticeable margin, demonstrating the effectiveness of class-aware patch embedding adaptation. Both the 1-shot performance and the 5-shot performance are saturated after λ exceeds a certain value, *i.e.*, 1-shot: 2 and 5-shot: 8. Since there are more diverse entities under the 5-shot setting, a larger λ is needed to make all patch embeddings class-relevant. Considering that the 5-shot performance improvement is small when $\lambda > 2.0$, the default class-aware factor is set to be 2.

Choices in Eq. (3). Table 5 shows that both $d(\cdot,\cdot)^2$ and $|d(\cdot,\cdot)|$ can boost performance. Also, scaling $|d(\cdot,\cdot)|$ doesn't bring further performance gains. Since the cosine values within the same class are larger than the others, squaring enlarges the intra-class similarity while reducing the inter-class similarity, thus achieving better performance.



(a) Task 1 without CPEA (b) Task 2 without CPEA (c) Task 3 without CPEA (d) Task 4 without CPEA

(e) Task 1 with CPEA (f) Task 2 with CPEA (g) Task 3 with CPEA (h) Task 4 with CPEA

Figure 5. Patch embedding visualization of four randomly sampled 5-way 1-shot classification tasks with one query image per class. (a), (b), (c), and (d) show the visualization results without CPEA. (e), (f), (g), and (h) show the corresponding visualization results with CPEA. CPEA concentrates patch embeddings by class, thus making them class-relevant.

Classifier	5-shot
Prototype (with Euclidean distance)	82.80±0.59
Prototype (with Cosine. distance)	79.90±0.65
Linear (optimized online)	82.37±0.57
FewTURE (0 steps)	82.68±0.55
FewTURE (20 steps)	84.51±0.53
CPEA (in a dense manner)	87.06±0.38

Table 7. Impact of different classifiers attached to the pretrained backbone ViT-S/16 on the few-shot classification performance.

Number of class-agnostic embeddings	1-shot	5-shot
1	71.97±0.65	87.06±0.38
2	71.43±0.65	86.62±0.41
3	71.34±0.65	86.72±0.40
4	70.57±0.67	86.39±0.36

Table 8. Impact of the number of class-agnostic embeddings on the few-shot classification performance.

Pretraining strategies. Table 6 shows the impact of different pretraining strategies on the few-shot classification performance. Supervision collapse exists in the supervised pretraining, *i.e.*, DeiT [60], thus leading to its poor generalization ability on the unseen test classes. Due to its patch-based nature, MIM [76] performs much better than DINO [7].

Classifier. Table 7 shows the impact of different classifiers attached to the pretrained backbone ViT-S/16 on the few-shot classification performance. The comparison results demonstrate the superiority of CPEA.

Number of class-agnostic embeddings. Table 8 shows the impact of the number of class-agnostic embeddings on the few-shot classification performance. Increasing the number of class-agnostic embeddings does not improve performance. Therefore,we use one class-agnostic embedding by default.

Model	Inference time $[ms]$	5-shot
FewTURE (0 steps)	156.8±2.16	82.68±0.59
FewTURE (10 steps)	162.1±2.11	83.89±0.57
FewTURE (20 steps)	168.6±2.22	84.51±0.53
CPEA	1.352±0.06	87.06±0.38

Table 9. Few-shot classification accuracy and inference time. Inference time are averaged over 1800 query images. An NVIDIA-2080TI is used for evaluation.

Image resolution	1-shot	5-shot
224 × 224	71.97±0.65	87.06±0.38
384 × 384	73.12±0.63	87.60±0.39
448 × 448	73.29±0.63	88.00±0.37

Table 10. Impact of the image resolution on the few-shot classification performance.

Inference time. When generalizing to the unseen test classes, our CPEA does not need any adjustments while FewTURE [26] needs to learn the token importance weight via online optimization. As a result, our CPEA is much faster than FewTURE in terms of inference speed, as shown in Table 9.

Feature visualization. Figure 5 shows the patch embedding visualization results of four randomly sampled 5-way 1-shot classification tasks with/without CPEA. It can be observed that with CPEA, the patch embeddings are clustered by class. This means that the patch embeddings are made class-relevant by CPEA.

Backbone	Model	miniImageNet		CIFAR-FS	
		1-shot	5-shot	1-shot	5-shot
ViT-S/16 [15]	PMF [28]	93.1	98.0	81.1	92.5
	CPEA	**94.3**	**98.3**	**90.4**	**96.3**
ViT-B/16 [15]	PMF [28]	95.3	98.4	84.3	92.2
	CPEA	**95.7**	**98.7**	**92.4**	**96.6**

Table 11. Impact of the external data on the few-shot classification performance.

Metric	1-shot	5-shot
$\|\cdot\|^2$	68.41±0.68	84.45±0.44
$\|\cdot\|^2 / d_{feature}$	71.53±0.67	85.78±0.41
$\|\cdot\|^2 / std$	62.37±0.62	83.11±0.43
cosine	71.97±0.65	87.06±0.38

Table 12. Impact of using the euclidean metrics in Eq. (3) on the few-shot classification performance.

Scalability in image resolution. It seems that our architecture is tied to a certain image resolution. In fact, inserting adaptive average pooling with a fixed output size of 14×14 allows us to increase image resolution without introducing additional parameters in MLP. Also, this strategy brings noticeable performance gains, as shown in Table 10.

External data. PMF [28] uses external data to push the limits of simple pipelines for few-shot image classification. With the same experimental settings as PMF, CPEA significantly outperforms PMF, as shown in Table 11. This observation demonstrates the effectiveness of our method in using external data.

5. Discussion

Despite the effectiveness of euclidean metrics and their generalizations in few-shot image classification [5, 19, 4], two major challenges still exist when using them. One is the scale issue. Table 12 shows that it is nontrivial to find a suitable scaling factor that outperforms the cosine similarity. The other one is the inaccurate mean and covariance estimated with a limited number of labeled data. Our method makes patch embeddings class-relevant, thus increasing the amount of labeled data. The increased labeled data might enable us to accurately estimate the mean and covariance in few-shot settings, which could improve the robustness and usability of the scale-invariant generalizations of euclidean metrics such as Mahalanobis distance. This is a promising direction worth exploring in the future.

6. Conclusion

In this paper, we propose a Class-aware Patch Embedding Adaptation (CPEA) method for few-shot image classification. CPEA can eliminate the interference of single-label annotations and avoid supervision collapse without the need for aligning semantically-relevant regions. The core of CPEA is to integrate patch embeddings with class-aware embeddings to make them class-relevant. To measure the similarity between paired images, we define a dense score matrix between class-relevant patch embeddings, based on which a similarity score is aggregated. Visualization results demonstrate that our CPEA makes patch embeddings class-relevant. Extensive experiments on four benchmark datasets show that our CPEA performs much better than the counterparts while achieving new state-of-the-art results.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (62206268, U21A20487), Shenzhen Technology Project (JCYJ20220818101206014, JCYJ20220818101211025, JCYJ20200109113416531), Guangdong Technology Project (2022B1515120067), and CAS Key Technology Talent Program.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Mixture-based feature space learning for few-shot image classification. In *ICCV*, 2021. 7

[2] Arman Afrasiyabi, Hugo Larochelle, Jean-François Lalonde, and Christian Gagné. Matching feature sets for few-shot image classification. In *CVPR*, 2022. 6

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *ICLR*, 2022. 3, 4

[4] Peyman Bateni, Jarred Barber, Jan-Willem van de Meent, and Frank Wood. Enhancing few-shot image classification with unlabelled examples. In *WACV*, 2022. 2, 9

[5] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *CVPR*, 2020. 2, 9

[6] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019. 6

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4, 8

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 4

[10] Zhengyu Chen, Jixie Ge, Heshen Zhan, Siteng Huang, and Donglin Wang. Pareto self-supervised training for few-shot learning. In *CVPR*, 2021. 6, 7

[11] Jun Cheng, Fusheng Hao, Fengxiang He, Liu Liu, and Qieshi Zhang. Mixer-based semantic spread for few-shot learning. *IEEE TMM*, 2023. 2

[12] Jun Cheng, Fusheng Hao, Liu Liu, and Dacheng Tao. Imposing semantic consistency of local descriptors for few-shot learning. *IEEE TIP*, 2022. 2

[13] Tristan Deleu, David Kanaa, Leo Feng, Giancarlo Kerg, Yoshua Bengio, Guillaume Lajoie, and Pierre-Luc Bacon. Continuous-time meta-learning with forward mode differentiation. In *ICLR*, 2022. 2

[14] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020. 2, 3

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 3, 4, 6, 9

[16] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014. 3

[17] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. Melr: Meta-learning via modeling episode-level relationships for few-shot learning. In *ICLR*, 2020. 6

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[19] Stanislav Fort. Gaussian prototypical networks for few-shot learning on omniglot. *arXiv:1708.02735*, 2017. 2, 9

[20] Zhi Gao, Yuwei Wu, Yunde Jia, and Mehrtash Harandi. Curvature generation in curved spaces for few-shot learning. In *ICCV*, 2021. 7

[21] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *ICCV*, 2019. 6, 7

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3

[23] Fusheng Hao, Fengxiang He, Jun Cheng, and Dacheng Tao. Global-local interplay in semantic alignment for few-shot learning. *IEEE TCSVT*, 2022. 2

[24] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *ICCV*, 2019. 2, 4

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021. 3

[26] Markus Hiller, Rongkai Ma, Mehrtash Harandi, and Tom Drummond. Rethinking generalization in few-shot classification. In *NeurIPS*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[27] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019. 2

[28] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, 2022. 9

[29] Adam Jelley, Amos Storkey, Antreas Antoniou, and Sam Devlin. Contrastive meta-learning for partially observable few-shot learning. In *ICLR*, 2023. 2

[30] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *ICCV*, 2021. 7

[31] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *ECCV*, 2020. 7

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6

[33] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 1

[34] Jinxiang Lai, Siqian Yang, Wenlong Liu, Yi Zeng, Zhongyi Huang, Wenlong Wu, Jun Liu, Bin-Bin Gao, and Chengjie Wang. tsf: Transformer-based semantic filter for few-shot learning. In *ECCV*, 2022. 3

[35] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019. 7

[36] SuBeen Lee, WonJun Moon, and Jae-Pil Heo. Task discrepancy maximization for fine-grained few-shot classification. In *CVPR*, 2022. 2

[37] Wenbin Li, Jinglin Xu, Jing Huo, Lei Wang, Yang Gao, and Jiebo Luo. Distribution consistency based covariance metric networks for few-shot learning. In *AAAI*, 2019. 4

[38] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. In *NeurIPS*, 2021. 3

[39] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. Learning a few-shot embedding model with contrastive learning. In *AAAI*, 2021. 6

[40] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. In *NeurIPS*, 2021. 3

[41] Yang Liu, Weifeng Zhang, Chao Xiang, Tu Zheng, Deng Cai, and Xiaofei He. Learning to affiliate: Mutual centralized learning for few-shot classification. In *CVPR*, 2022. 2

[42] 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 *ICCV*, 2021. 3

[43] Yuning Lu, Liangjian Wen, Jianzhuang Liu, Yajing Liu, and Xinmei Tian. Self-supervision can be a good few-shot learner. In *ECCV*, 2022. 3

[44] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. Rectifying the shortcut learning of background for few-shot learning. In *NeurIPS*, 2021. 6

[45] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. Partner-assisted learning for few-shot image classification. In *ICCV*, 2021. 6

[46] Rongkai Ma, Pengfei Fang, Gil Avraham, Yan Zuo, Tom Drummond, and Mehrtash Harandi. Learning instance and task-aware dynamic kernels for few shot learning. *arXiv:2112.03494*, 2021. 2

[47] Rongkai Ma, Pengfei Fang, Tom Drummond, and Mehrtash Harandi. Adaptive poincaré point to set distance for few-shot classification. In *AAAI*, 2022. 2

[48] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020. 3

[49] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv:1803.02999*, 2018. 2

[50] Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. Boil: Towards representation change for few-shot learning. In *ICLR*, 2021. 2

[51] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 6

[52] Guodong Qi, Huimin Yu, Zhaohui Lu, and Shuzhao Li. Transductive few-shot classification on the oblique manifold. In *ICCV*, 2021. 6

[53] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv:1803.00676*, 2018. 6

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[55] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2018. 2, 6

[56] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *CVPR*, 2020. 2

[57] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2, 6, 7

[58] 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 *CVPR*, 2018. 2

[59] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020. 7

[60] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 8

[61] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 2, 3, 6, 7

[62] Heng Wang, Tan Yue, Xiang Ye, Zihang He, Bohan Li, and Yong Li. Revisit finetuning strategy for few-shot learning to transfer the emdeddings. In *ICLR*, 2023. 2

[63] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *CVPR*, 2021. 6

[64] Jiamin Wu, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Task-aware part mining network for few-shot learning. In *ICCV*, 2021. 7

[65] Jiangtao Xie, Fei Long, Jiaming Lv, Qilong Wang, and Peihua Li. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *CVPR*, 2022. 6

[66] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *CVPR*, 2021. 6

[67] Zhanyuan Yang, Jinghua Wang, and Yingying Zhu. Few-shot classification with contrastive learning. In *ECCV*, 2022. 3

[68] Lifchitz Yann, Avrithis Yannis, and Picard Sylvaine. Local propagation for few-shot learning. In *ICPR*, 2021. 4

[69] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020. 2, 3, 6

[70] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020. 2, 6, 7

[71] Chi Zhang, Henghui Ding, Guosheng Lin, Ruibo Li, Changhu Wang, and Chunhua Shen. Meta navigator: Search for a good adaptation policy for few-shot learning. In *ICCV*, 2021. 6, 7

[72] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2018. 5

[73] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. Iept: Instance-level and episode-level pretext tasks for few-shot learning. In *ICLR*, 2020. 6

[74] Xueting Zhang, Debin Meng, Henry Gouk, and Timothy M Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *ICCV*, 2021. 6, 7

[75] Jiabao Zhao, Yifan Yang, Xin Lin, Jing Yang, and Liang He. Looking wider for better adaptive representation in few-shot learning. In *AAAI*, 2021. 6

[76] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *ICLR*, 2022. 2, 3, 4, 6, 8

[77] Ziqi Zhou, Xi Qiu, Jiangtao Xie, Jianan Wu, and Chi Zhang. Binocular mutual learning for improving few-shot classification. In *ICCV*, 2021. 6, 7

[78] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *ICML*, 2019. 2