Instance-based Max-margin for Practical Few-shot Recognition

Minghao Fu Ke Zhu*
National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
fumh@lamda.nju.edu.cn, zhuk@lamda.nju.edu.cn

Abstract

In order to mimic the human few-shot learning (FSL) ability better and to make FSL closer to real-world applications, this paper proposes a practical FSL (pFSL) setting. pFSL is based on unsupervised pre-trained models (analogous to human prior knowledge) and recognizes many novel classes simultaneously. Compared to traditional FSL, pFSL is simpler in its formulation, easier to evaluate, more challenging and more practical. To cope with the rarity of training examples, this paper proposes IbM2, an instance-based max-margin method not only for the new pFSL setting, but also works well in traditional FSL scenarios. Based on the Gaussian Annulus Theorem, IbM2 converts random noise applied to the instances into a mechanism to achieve maximum margin in the many-way pFSL (or traditional FSL) recognition task. Experiments with various self-supervised pre-training methods and diverse many- or few-way FSL tasks show that IbM2 almost always leads to improvements compared to its respective baseline methods, and in most cases the improvements are significant. With both the new pFSL setting and novel IbM2 method, this paper shows that practical few-shot learning is both viable and promising.

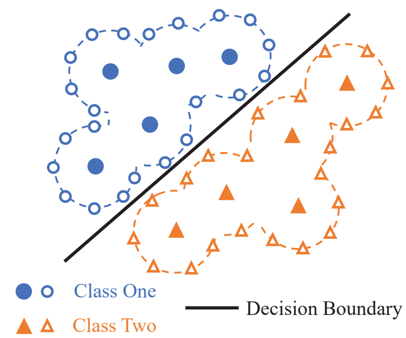

Figure 1. Instance-based max-margin. We create a hypersphere (large dashed circles) centered at every example (solid shapes), sample many virtual examples (hollow shapes) around the hypersphere, and require the decision boundary to classify all virtual samples correctly. Then we in effect achieve instance-based max-margin. Best viewed in color.

1. Introduction

We human have the ability to learn new concepts based on a few exemplars, thanks to both our ability to learn and the previously accumulated knowledge. As a stark contrast, learning machines (especially deep learning models) mostly require plentiful training examples to learn just few (*e.g.*, 5) concepts. Both the vision and learning community are fully aware of this shortcoming, and few-shot learning (FSL) is our community's effort to counter this weakness, which has been studied for a long time [39].

FSL relies on prior knowledge, too. Currently, the setting is to set aside a collection of object categories (known as the *base set*), with many training examples in every base set category. A model is first pre-trained on the base set, which naturally encodes prior knowledge. Then, few *novel* object categories (most commonly 5) are to be recognized based on the pre-trained model given few training examples from these new concepts (*e.g.*, 1 or 5 per category). Note that the base and novel training sets are semantically closely related (*e.g.*, both contain different kinds of birds).

There is a clear mismatch between the human ability and this FSL setting. We human can learn *many* instead of few novel concepts; and, the human prior knowledge include those accumulated as both common-sense and domain knowledge from *many diverse domains* rather than a limited set of concepts closely related to a specific task.

Although this FSL setting has served our community well and has pushed the technical frontier for years, with the immense advances in deep learning and emerging practical needs for real-world FSL applications, *it is time that we need a better, simpler, and more practical FSL setting.*

Hence we advocate a practical few-shot learning (or pFSL) setting: based on an *unsupervised* model pre-trained from a large number of concepts (*e.g.*, ImageNet [30]), simultaneously learn *many* new concepts (*e.g.*, 200) with *few*

*K. Zhu is the corresponding author.

examples (*e.g.*, 1 or 5 per category).

An ImageNet pre-trained model is more analogous to human knowledge than those learned from a base set in traditional FSL, and learning many new concepts makes it more practical in applications. *Without the base set*, the learning phase of pFSL is simple, too. More importantly, evaluation of algorithms is difficult, complicated, and very time consuming in the current FSL setting [16], but as will be discussed later, pFSL does not suffer from this drawback.

Although an ImageNet pre-trained model undoubtedly contains more prior knowledge than a base-set trained one, simultaneously learning many new concepts will inevitably make pFSL much more difficult than traditional FSL. We believe that a more challenging task will help technology advancement, too—the traditional FSL setting is already saturated to certain extent (*cf*. results in [22, 41]).

Hence, we propose a novel approach for pFSL: Instance-based Max-margin (IbM2), which is effective in traditional FSL, too. The max-margin idea has been shown to be effective in improving generalization. Classic methods like SVM [7] build max-margin decision boundaries with *only a very small portion* of the training examples (*i.e.*, support vectors). This idea is problematic when we have both very few training examples and a very high dimensionality. Intuitively, the support vectors chosen by SVM are highly likely wrong or misleading, because a small training set leads to excessively unstable estimation [16].

In the proposed IbM2, we achieve max-margin *at the instance level*, instead of at the class level as in SVM. As illustrated in Fig. 1, we create a hypersphere for *every* training example, with the example being the center of the hypersphere. Then, we can sample as many *virtual samples* as we want around this hypersphere, and assign the label of the center (original training example) to all its associated virtual examples. By finding a properly *maximized radius* for the hypersphere and requiring a classifier to *correctly classify the virtual examples*, we in effect achieve max-margin. Another benefit of IbM2 is that we do not need any special handling for multi-class recognition, while vanilla SVM is for binary classification only. In short, the contributions of this paper are twofold:

- We propose a practical few-shot learning setting (pFSL): many-way (*e.g.*, 200-way) recognition, uses an unsupervised pre-trained model, and has no base set.
- We propose IbM2, an instance-based max-margin algorithm, which suits few-shot, high dimensionality, and multi-class naturally.

As will be shown by extensive experiments, IbM2 consistently improves both traditional FSL and the proposed pFSL. The benefits of pFSL against traditional FSL will be verified by both analyses and experiments, too.

2. Related Work

Few-shot learning aims at recognizing novel categories with the help of prior knowledge. Given the base set as prior knowledge, few-shot learning methods can be roughly divided into two types: those based on meta-learning or transfer-learning.

Meta-learning based FSL. Meta-learning [21] learns a model through a large quantity of episodes consisting of different training and evaluation sets. Given a new task during evaluation, the model performs what it does in training to predict. In FSL, the training episodes are randomly sampled from the base set. One line of work [14, 15, 25, 31] focuses on learning general initial weights for fast adapting the model to unseen categories with a few steps of optimization. For example, [14] explicitly trains the classifier in a model-agnostic way. [25] extends meta-learning with linear predictors as a differentiable convex problem. [15] learns the task distribution from a probabilistic perspective. Another line of work [1, 11, 32, 33, 37, 42, 43] boosts meta-learning by exploring different distance metrics. For example, [32] aggregates training features to generate prototypes for different classes as the classification head. [43] leverages the Earth Mover's Distance to calculate the structural distance for classification. [1] extracts sets of features from each image to build a set-based matching schema.

Transfer-learning based FSL. This line of attack [9, 12, 27, 35, 40] leverages the idea of standard transfer learning, which first pre-trains a model on base classes and then finetunes the model weights with limited training samples from novel classes. [9] normalizes the features and classification weights and calculates their cosine distances as the logits. [27] pre-trains with manifold mixup [36] to improve robustness. [12] introduces a baseline for effectively and transductively finetuning. [40] calibrates the training distribution of a few-shot task using statistics from the base set. [35] explores learning a general representation of base classes in supervised or self-supervised ways with self-distillation.

Self-supervised learning. We stress that in pFSL the pre-trained model is unsupervised, for which the reason is to be explained in Sec. 3. Self-supervised learning (SSL) is the mainstream approach to train a deep net in the unsupervised manner. Popular SSL methods, whether applied to traditional ResNet [18] models (*e.g.*, BYOL [17], MoCo [19] and SwAV [5]) or Vision Transformers (ViT) [13] (*e.g.*, MSN [2], DINO [6] and MoCov3 [10]), all produce pre-trained models based on large-scale datasets. In this paper, we utilize such models as our prior knowledge in FSL.

3. pFSL: Practical Few-Shot Learning

In traditional FSL, the training dataset is split into 1) the *base set* with N_b labeled images, denoted as $D_b = \{(x_i^b, y_i^b)\}_{i=1}^{N_b}$ where $y_i^b \in \mathcal{Y}_b$ is the label of instance x_i^b and

$\mathcal{Y}_b$ is the label space of D_b; 2) the *novel set* with N_n labeled images, denoted as $D_n = \{(x_j^n, y_j^n)\}_{j=1}^{N_n}$ where $y_j^n \in \mathcal{Y}_n$ is the label of x_j^n and $\mathcal{Y}_n$ is the label space of D_n. Note that although $\mathcal{Y}_b \cap \mathcal{Y}_n = \varnothing$, the concepts in $\mathcal{Y}_b$ and $\mathcal{Y}_n$ are semantically similar to each other. Then, the task is to learn a recognizer generalizing well on unknown samples from $\mathcal{Y}_n$. By letting $N_n \ll N_b$, the training samples in D_n are few-shot (*i.e.*, few training examples per class). In traditional FSL, $|\mathcal{Y}_b|$ is relatively large but $|\mathcal{Y}_n|$ is small. For example, in most cases $|\mathcal{Y}_n| = 5$, which is called a 5-way FSL. The evaluation of traditional FSL is a rather complicated task.

3.1. The pFSL Formulation

To make FSL simpler, more practical, and closer to human's few-shot learning capability, we propose a new practical few-shot learning (pFSL) paradigm, which is characterized by the following properties:

- **Removing the base set.** The prior knowledge obtained through pre-training on the base set is limited, as the base set only contains limited number of concepts. In some applications, collecting data sharing similar concepts with the novel set may be as difficult as collecting more samples for the novel set itself.
- **Pre-trained model based on big data.** A model pre-trained on a large training set containing a wide variety of concepts is analogous to the common-sense and domain knowledge our brains encode, which will be useful for few-shot learning in diverse domains.
- **Many-way FSL.** In a N-way k-shot FSL, we seek N to be large and k to be small, *e.g.*, using all the CUB [38] categories ($N = 200$ categories and k is 1 or 5). This makes pFSL much closer to real-world applications than traditional FSL.
- **Unsupervised pre-training.** In pFSL, we require the pre-training to be unsupervised. Since a large scale dataset (*e.g.*, ImageNet) may contain the concept in few-shot learning (e.g., those birds in CUB), avoiding using the labels during pre-training is necessary to make the evaluation of FSL algorithm fairer.

These properties distinguish pFSL from traditional FSL and some recent variants of it [9, 12, 22, 23, 27].

3.2. Simplicity and Better Evaluation

The pFSL framework is clearly *simpler* than traditional FSL, and it leads to not only benefits in the training phase, but more importantly *better evaluation* of FSL algorithms.

Traditional N-way k-shot FSL typical use $N = 5$ and $k = 1, 5$, which inevitably leads to unreliable estimation of the test accuracy. Hence, the common practice is to sample a large number of n episodes ($n \geq 500$), find the accuracy in each episode, and report the average accuracy μ along with its 95% confidence interval $Z_{95\%}$. Note that with n episodes, the standard deviation σ of the accuracy within the n episodes is $\sigma = 0.51\sqrt{n}Z_{95\%}$. For example, a typical result is: average accuracy 64.93%, 95% confidence interval 0.18%, but the standard deviation of accuracy is 9.18% [16] (in which $n = 10000$). In short, one *single* evaluation of traditional FSL *requires training and testing for n times ($n \geq 500$)* and *the evaluation results are still unreliable*. Because of the complicated setup, both learning and evaluation of traditional FSL algorithms are not only costly, but often carried out in slightly different ways, which renders the fair comparison even more difficult.

As a stark contrast, pFSL is much simpler and as the experimental results will show in Table 1, the estimate of average accuracy is reliable. The standard deviation is small (mostly $< 1.0\%$). This reliable estimation comes from the fact that pSFL is many-way (i.e., N is large). Hence, instead of running n episodes, *few episodes* (*e.g.*, 3) is enough to evaluate pFSL, which not only means savings in computation, but also more reliable evaluation of algorithms.

4. IbM2: Instance-based Max-margin

In pFSL, we have a pre-trained model $\mathcal{M}$ (unsupervised trained on a large set) and a novel set. There is no base set D_b any more, hence we will simply denote the novel set as $D = \{x_i, y_i\}_{i=1}^{M}$ (that is, ignoring the subscript $_b$). The label set of D is $\mathcal{Y}$. In an N-way k-shot pFSL task, we have $|\mathcal{Y}| = N$ and there will be k training examples in each of the N novel classes, hence $M = kN$. Note that we expect $|\mathcal{Y}| = N$ to be large (*i.e.*, many-way FSL) and k to be small (*e.g.*, 1 or 5).

4.1. Generating Virtual Samples

Since k (the number of training images in each class) is still small but the number of classes (N) becomes much larger (*e.g.*, from 5 in traditional FSL to 200 in pFSL), we expect that the challenge in learning a pFSL model is greater than that in traditional FSL. This is verified by our experiments (*cf*. the results in Table 1 vs. those in Table 3).

Virtual examples, or examples that are sampled based on the original training examples, have been repeatedly proven useful when the number of training examples is small, *e.g.*, in traditional few-shot learning [40] or long-tailed recognition (where the tail classes have few training images) [20].

The common idea in generating virtual samples is to make the virtual samples follow the underlying distributions of the classes. However, in few-shot learning *this requirement is very difficult to entertain* even with the help of techniques such as distribution calibration (DC) in [40], because there is only 1 or 5 examples per class.

In contrast, we do not try to sample from any underlying distribution, but try to achieve max-margin of the decision boundary, as illustrated in Fig. 1. Specifically, for every training *instance* x_i, we i.i.d. generate R virtual samples $x_{i,r}$ ($1 \leq r \leq R$) which distribute around the shell of the hypersphere centered at x_i and whose radius is controlled by a parameter ϵ. Note that the same ϵ is shared by all training examples x_i ($1 \leq i \leq kN$).

Technically, let z_i be the feature vector produced by the pre-trained model $\mathcal{M}$: $z_i = \mathcal{M}(x_i)$ from a forward calculation. Then, a noise vector $\delta_{i,r}$ is randomly sampled from the standard multi-dimensional normal distribution $N(0, I_d)$, where d is the length of z_i and I_d is a $d \times d$ identity matrix. By adding a scaled version of the noise vector to the i-th training example, we obtain $z_{i,r}$ (the r-th virtual example for x_i):

$$z_{i,r} = z_i + \epsilon\delta_{i,r} \, . \tag{1}$$

Its label $y_{i,r}$ is y_i (the label of x_i) and ϵ is a positive number.

Interesting properties exist in our virtual examples:

- Hypersphere for different training examples *can overlap*, so long as they belong to the same class (*cf*. Fig. 1).
- We require the *virtual* examples to be correctly classified, or, the original training examples are *discarded*.
- The virtual examples lie *around the shell, not the interior* of the hypersphere. According to the Gaussian Annulus Theorem [4], almost all the probability of a high-dimensional spherical Gaussian with unit variance is concentrated in a thin annulus of width $O(1)$ with radius $\sqrt{d}$ when d is large. Since d is indeed large (*e.g.*, $d = 2048$), the virtual examples reside around the shell. Hence, if we maximize the radius parameter ϵ but require that the virtual examples are correctly classified, as Fig. 1 shows, we in essence achieve margin maximization in an *instance-based* manner.

Hence, the proposed method is named Instance-based Max-margin, abbreviated as IbM2.

4.2. Ellipsoidal Noise Generation

One issue with the above isotropic noise sampling is that it ignores the structural property of the training examples. For instance, if the ranges for two feature dimensions are $[-1, 1]$ and $[-100, 100]$, respectively, isotropic sampling is clearly unsuitable.

To overcome this drawback, we calculate a range vector $s = (s_1, s_2, \ldots, s_d)$ using all the original training examples *regardless of the class label*, where s_i is the sample standard deviation for the i-th dimension. Then, the IbM2 virtual examples are generated as (replacing Eq. 1)

$$z_{i,r} = z_i + \epsilon(s \odot \delta_{i,r}) \, , \tag{2}$$

where $\odot$ is element-wise multiplication. The final virtual example generation process is illustrated in Fig. 2. Note that estimating s is dramatically easier than estimating the underlying distribution of every novel class.

It is worth noting that this ellipsoidal noise generation is *unsupervised*, which distinguishes itself from Distribution calibration (DC) [40]. DC samples virtual examples based

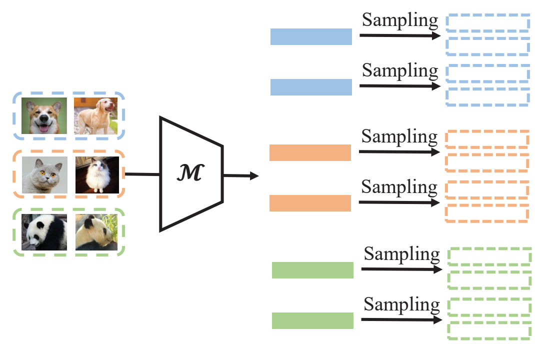

Figure 2. Virtual example generation. Original input images pass through the pre-trained model $\mathcal{M}$ to get feature vectors (solid rectangles). Then, every feature vector i.i.d. samples R virtual examples ($R = 2$ here) using Eq. 2.

on an estimated covariance structure, which is estimated from the *few* novel class training examples in a sample-wise manner and calibrated using the *base set*. As aforementioned, we argue that it makes sense to get rid of the dependency on a base set, then DC is not applicable.

More importantly, the reliability of this estimation of a dense covariance matrix is understandably much lower than our estimation of the range indicator vector s—we estimate d values using $k \times N$ examples, while DC estimates $d \times d$ values using only 1 example and the base set. The base set calibration will help DC, but clearly its estimation is still unreliable, as shown by experiments in [40, 44]

4.3. Binary Search to Find the Largest Possible ϵ

Obviously we want ϵ to be as large as possible to achieve max-margin, under the condition that virtual examples are classified correctly. Hence, we design a simple binary search to find the optimal value for ϵ.

For any given ϵ value, we can generate kNR virtual examples $(x_{i,r}, y_{i,r})$ ($1 \leq i \leq kN$, $1 \leq r \leq R$) using Eq. 2. We denote the virtual training set for this particular value of ϵ as D^ϵ. Then, we train a linear classifier with parameters W to classify D^ϵ and obtain the training accuracy. If the training accuracy is too high (larger than a threshold T), ϵ is enlarged; otherwise ϵ is shrunk. The binary search maintains a search range, when the search range is tight enough (when the range is less than 0.05 in Algorithm 1), the search terminates, and we have found the optimal value $\hat{\epsilon}$.

We have a few notes for finding $\hat{\epsilon}$:

- Unlike DC [40], we do *not* need a validation set.
- The threshold T does not need to be 1 (100%). In SVM learning, the slack variable trick allows some support vectors not to have max-margin. Similarly, we just need T to be close to 1 (0.9 in our experiments).
- Different from classic augmentation techniques [26, 28],

Algorithm 1 Pseudo code for searching ϵ

```
# Inputs:
#     x   : training features of a few-shot task
#     y   : training labels   of a few-shot task
#     R   : sampling times for an instance
#     T   : accuracy threshold for searching
# Outputs:
#     eps   : epsilon for sampling

left = 0.0
right = a large value
eps = right / 2

W = init_classifier
while True:
        acc = train_and_eval(W, x, y, eps, R)
        if acc > T:
                left = eps    # increase epsilon
        else:
                right = eps  # decrease epsilon
        eps = (left + right) / 2.0
        if right - left < 0.05:
                break
```

which introduce minor disturbance into input features, the distribution of $D^{\hat{\epsilon}}$ is drastically different from D because of a large $\hat{\epsilon}$ (empirically 1.24 on average).

Finally, the IbM2 pipeline is simple: First, use Algorithm 1 to find $\hat{\epsilon}$. Second, generate the virtual dataset $D^{\hat{\epsilon}}$. And third, learn a linear classifier W. Note that the pre-trained model $\mathcal{M}$ is frozen and will not be updated.

5. Experimental Results

We evaluate our IbM2 method on both setups: the proposed pFSL in many-way few-shot scenario, and also comparing IbM2 under the traditional FSL few-way setting with state-of-the-art FSL methods to further validate its effectiveness.

5.1. Implementation Details

For both the searching and training stages of IbM2, all features extracted by the backbone $\mathcal{M}$ were $L2$-normalized first. When learning the classification head (both during the binary search and after $\hat{\epsilon}$ was determined), we used Adam as the optimizer and the label smoothed cross-entropy loss [34] as the learning objective. The pre-trained model was trained using various self-supervised learning methods on the ImageNet-1K [30] training set.

During the binary search, for all the experiments we will report, we always set R as 200, T as 0.9 in pFSL and 0.999 in the traditional FSL setting, except when we carried out ablation experiments on these hyperparameters.

Full implementation details are included in the appendix.

5.2. Experiments in the pFSL Setting

Datasets and Evaluation Setup. We explored two common many-way classification datasets, ImageNet-1K [30] and CUB-200-2011 (CUB) [38]. ImageNet-1K contains about 1.28 million training images from 1,000 classes and 50,000 images for evaluation. CUB is a fine-grained recognition dataset composed of 11,788 images belonging to 200 classes of birds, 5,994 for training and 5,794 for evaluation.

For both datasets, we randomly sampled the training sets for pFSL by selecting k images from every class (*i.e.*, k-shot) and using all the classes (1000 and 200, respectively) to form the novel set. k is chosen from $\{1, 2, 3, 4, 5, 8, 16\}$. The evaluation was carried out on the full validation (for ImageNet) or test (for CUB) set.

Utilizing various self-supervised backbone network $\mathcal{M}$ pre-trained on ImageNet, we compare two sets of results: one in which the linear classifier was obtained by using the original training set, the other by using IbM2's virtual examples. We report the top-1 accuracy and its standard deviation computed over 3 randomly sampled novel sets.

Main Results. As the results in Table 1 suggest, we evaluated on two types of backbones: Vision Transformer (ViT) [13] and ResNet50 [18]. The pre-training methods included DINO [6], MoCov3 [10], MSN [2], SimCLR [8] and BYOL [17]. Table 1 shows that *in almost all cases*, IbM2 benefited the few-shot learning process and improved the top-1 accuracy.

When the number of shots (k) is very small (≤ 2), the improvement of IbM2 over the baseline is roughly 1% on average. However, as the number of training shots goes larger, it is clear that the level of accuracy improvement increases gradually from $\sim 1\%$ to $\sim 2\%$, even $> 3\%$ in some cases.

A useful observation from these results is that in pFSL, both the baseline and IbM2 have small standard deviations, that is, more *robust in evaluation*. Hence, we do not need 500 episodes in pFSL any longer—3 is enough.

Semi-supervised learning. In the self-supervised learning literature, it is a common practice to use ImageNet-1K training images to learn a backbone in the unsupervised manner, then use a small portion (*e.g.*, 1%) of the training data now with labels to train a classifier. This semi-supervised learning task can also be viewed as our pFSL setting. Hence, we also report the 1% ImageNet-1K semi-supervised learning (on average 12 labeled training images per class) results in Table 2. IbM2 consistently improved various self-supervised models and backbone architectures *in all cases*, with 0.8% to 2.9% top-1 accuracy increase.

5.3. Experiments in the Traditional FSL Setting

Datasets and Evaluation Setup. We evaluated IbM2 in the traditional FSL setup, too. We conducted experiments on two standard benchmark datasets, *mini*-ImageNet [37] and CIFAR-FS [3]. *mini*-ImageNet consists of 100 categories selected from ImageNet-1K, which are further split into 64 base, 16 val and 20 novel categories according to [29]. CIFAR-FS is created by randomly shuffling the 100 categories of CIFAR-100 [24] into 64 base, 16 val, 20 novel

Dataset	Pre-training Method	Backbone	IbM2	Shot per Class 1	2	3	4	5	8	16
ImageNet-1K	DINO	ViT-S/16		39.2 ± 0.3	49.2 ± 0.2	54.1 ± 0.4	56.7 ± 0.2	58.0 ± 0.1	60.4 ± 0.1	62.7 ± 0.1
			✓	39.2 ± 0.3	49.4 ± 0.3	**54.6 ± 0.4**	**57.6 ± 0.1**	**59.3 ± 0.1**	**62.4 ± 0.2**	**65.8 ± 0.1**
	MoCov3	ViT-S/16		32.7 ± 0.6	42.0 ± 0.2	46.9 ± 0.3	49.6 ± 0.4	51.0 ± 0.1	53.8 ± 0.1	56.6 ± 0.2
			✓	**33.9 ± 0.6**	**43.2 ± 0.2**	**48.4 ± 0.3**	**51.3 ± 0.3**	**52.8 ± 0.2**	**56.1 ± 0.1**	**59.8 ± 0.2**
	MSN	ViT-S/16		47.9 ± 0.1	56.2 ± 0.4	59.8 ± 0.3	61.6 ± 0.1	62.4 ± 0.2	64.4 ± 0.3	66.1 ± 0.1
			✓	47.8 ± 0.2	56.4 ± 0.4	**60.5 ± 0.2**	**62.5 ± 0.2**	**63.6 ± 0.2**	**66.0 ± 0.2**	**68.4 ± 0.0**
		ViT-B/4		53.2 ± 0.2	64.5 ± 0.4	68.9 ± 0.2	70.9 ± 0.2	72.0 ± 0.3	73.8 ± 0.1	75.0 ± 0.2
			✓	**54.0 ± 0.1**	64.9 ± 0.5	**69.4 ± 0.2**	**71.4 ± 0.1**	**72.7 ± 0.4**	**74.7 ± 0.0**	**76.4 ± 0.2**
		ViT-L/7		57.3 ± 0.4	66.5 ± 0.4	69.8 ± 0.5	71.6 ± 0.4	72.2 ± 0.2	73.8 ± 0.1	75.1 ± 0.1
			✓	57.7 ± 0.4	66.6 ± 0.5	70.1 ± 0.6	71.8 ± 0.4	72.6 ± 0.2	**74.3 ± 0.1**	**76.0 ± 0.0**
	SimCLR	ResNet50		21.4 ± 0.4	30.3 ± 0.1	36.1 ± 0.3	39.8 ± 0.2	42.0 ± 0.1	46.8 ± 0.1	51.9 ± 0.0
			✓	**23.6 ± 0.4**	**33.4 ± 0.2**	**39.0 ± 0.4**	**42.0 ± 0.3**	**44.2 ± 0.1**	**48.0 ± 0.0**	**52.7 ± 0.0**
	BYOL	ResNet50		26.5 ± 0.3	35.7 ± 0.2	41.5 ± 0.4	45.1 ± 0.2	47.2 ± 0.1	51.8 ± 0.1	57.1 ± 0.1
			✓	**27.5 ± 0.3**	**37.5 ± 0.1**	**43.3 ± 0.4**	**46.8 ± 0.2**	**49.1 ± 0.1**	**53.2 ± 0.1**	**58.0 ± 0.1**
CUB	DINO	ViT-S/16		35.4 ± 1.2	49.0 ± 0.5	56.8 ± 0.8	60.8 ± 0.7	65.2 ± 0.9	70.6 ± 0.9	75.9 ± 0.3
			✓	**36.2 ± 1.4**	**49.6 ± 0.6**	**57.4 ± 1.0**	**62.0 ± 0.6**	**66.4 ± 0.8**	**72.5 ± 0.8**	**79.0 ± 0.2**
	MSN	ViT-S/16		32.1 ± 1.6	45.0 ± 0.6	53.1 ± 0.6	56.7 ± 0.1	61.4 ± 0.5	67.3 ± 0.0	73.6 ± 0.4
			✓	**33.0 ± 1.4**	**45.8 ± 0.7**	53.2 ± 0.9	57.1 ± 0.4	**62.0 ± 1.0**	**68.4 ± 0.1**	**75.7 ± 0.2**
		ViT-L/7		34.9 ± 1.3	49.4 ± 0.4	58.8 ± 0.8	62.7 ± 0.9	67.2 ± 0.3	73.8 ± 0.8	80.4 ± 0.2
			✓	**37.5 ± 1.2**	**50.1 ± 0.5**	59.0 ± 0.8	62.6 ± 0.5	67.5 ± 0.6	73.9 ± 0.4	**81.0 ± 0.1**

Table 1. Average of top-1 accuracy (%) with standard deviation across 3 random subsets on ImageNet-1K and CUB. A ✓ in the column IbM2 denotes the proposed IbM2 method, or the baseline method if it is blank. Those IbM2 results that are at least 0.5% higher than the baseline are shown in boldface.

Pre-training Method	Backbone	IbM2	Top 1
DINO	ViT-S/16		62.5
		✓	65.0
MoCov3	ViT-S/16		56.0
		✓	58.9
MSN	ViT-S/16		65.9
		✓	67.9
	ViT-B/4		74.9
		✓	76.1
	ViT-L/7		74.8
		✓	75.6
SimCLR	ResNet50		50.5
		✓	51.3
BYOL	ResNet50		55.7
		✓	56.8

Table 2. Top-1 accuracy (%) of 1% ImageNet-1K semi-supervised learning. The training set contains on average 12 labeled training samples per category.

categories. Note that in order to perform traditional few-shot learning, only the novel split of these two datasets is required to sample many 5-way 1/5-shot episodes. We report the metrics mentioned in [16]: the average (ACC_m), worst-case (ACC_1), average of worst 10 (ACC_{10}), average of worst 100 (ACC_{100}) episodes' accuracy, and the standard deviation (σ) over 500 episodes for comprehensive evaluation. To make the results more reliable, we average the value of each metric from 5 runs with different random seeds. We adopted PMF [22], $S2M2_R$ [27] and Meta-Baseline [11] as the pre-training or meta-training approaches to obtain the backbone weights out of the base set.

Main Results. As shown in Table 3, by plugging IbM2 in during classifier learning, the proposed IbM2 improved the average accuracy ACC_m in all cases except only 1 result. And, the standard deviation σ was reduced by IbM2 or remain unchanged in all cases. As [16] advocated, smaller σ means the learning process is more stable.

[16] also advocates the worst-case episode's accuracy is more important than the average accuracy among all episodes. Table 3 shows that IbM2 almost always leads to higher worst-case (ACC_1) or near-worst-case (ACC_{10}, ACC_{100}) accuracy. Furthermore, the average gain of ACC_m is 0.4%, which is far less than that of ACC_1 (1.3%). As for ACC_{10} and ACC_{100}, their gains are very similar, both around 0.5%. These observations demonstrate that IbM2 can boost the recognition accuracy of few-shot learning in almost every scenario, especially the worst case one.

At last, by comparing the numbers in Tables 1 and 3, pFSL is more *challenging* than traditional FSL, which leads to more open questions to solve.

5.4. Ablation Studies

We performed ablation studies for IbM2 in the pFSL setting on ImageNet-1K.

Instance-based vs. class-based max-margin. IbM2 seeks max-margin through an instance-based manner. In this study, we compare this instance-based max-margin with the well-known support vector machine (SVM), which

Dataset	Pre-training & Meta-training	Backbone	IbM2	1-shot					5-shot				
				ACC_m	σ	ACC_1	ACC_{10}	ACC_{100}	ACC_m	σ	ACC_1	ACC_{10}	ACC_{100}
mini-ImageNet	PMF	ViT-S/16		93.4	6.1	61.3	72.4	83.4	98.4	1.8	87.7	91.4	95.6
			✓	94.4	5.6	63.2	74.7	85.1	98.6	1.8	86.1	91.3	95.9
		ViT-B/16		94.9	5.4	66.9	75.5	86.0	98.8	1.6	89.1	92.9	96.3
			✓	95.6	5.1	69.6	77.6	87.1	98.9	1.6	89.3	93.0	96.3
		ResNet50		94.9	5.3	67.2	75.2	86.3	98.5	1.7	89.3	91.9	95.8
			✓	95.3	5.2	66.1	74.6	86.9	98.8	1.6	89.3	92.7	96.3
	$S2M2_R$	WRN-28-10		65.4	10.0	32.0	39.3	50.7	82.3	7.1	52.0	62.5	71.8
			✓	65.8	10.0	32.5	40.5	51.1	82.9	7.0	53.3	63.3	72.3
	Meta-Baseline	ResNet12		62.9	10.4	26.7	34.3	47.4	79.0	7.5	54.7	58.0	67.9
			✓	63.0	10.2	32.3	35.6	47.9	79.5	7.3	54.9	58.5	68.7
CIFAR-FS	PMF	ViT-S/16		87.9	8.3	51.7	61.7	74.8	95.5	4.2	74.7	81.8	88.5
			✓	89.0	8.1	59.2	64.1	75.9	95.7	4.2	73.3	81.7	88.7
		ViT-B/16		89.8	8.1	55.2	64.8	76.8	96.0	4.1	73.1	82.2	89.2
			✓	90.3	8.0	58.9	65.5	77.4	96.0	4.1	74.1	81.8	89.2
		ResNet50		81.7	10.4	35.2	51.6	65.8	91.3	5.5	71.2	75.1	82.7
			✓	82.3	10.3	36.3	51.1	66.5	91.6	5.3	73.3	76.0	83.2
	$S2M2_R$	WRN-28-10		75.2	10.8	44.0	46.7	59.2	87.7	7.1	54.7	67.9	76.8
			✓	75.5	10.8	42.4	45.5	59.2	87.7	7.0	54.9	68.1	77.0
	Meta-Baseline	ResNet12		72.7	11.6	32.0	42.9	56.1	84.8	7.5	54.7	65.1	73.6
			✓	72.3	11.5	32.8	42.3	55.7	85.1	7.5	56.3	65.4	74.1

Table 3. Results of 5-way classification on *mini*-ImageNet and CIFAR-FS in the traditional few-shot learning setting. For the baseline methods, we obtain the results from their respective official implementations.

achieves max-margin in a class-based manner. We experimented with SVM using the linear kernel function in the LIBSVM [7] software package. By iterating the value of SVM's regularization hyperparameter C from the set $\{0.1, 1, 10, 100, 1000\}$, we found the best accuracy among them to compare with our IbM2 method.

As shown in Table 4, the proposed IbM2 method outperforms SVM with the best C by a significant margin in all cases. Specifically, the recognition accuracy improves $\sim$2% on average, and even up to $\sim$4% in the extremely scarce 1-shot case. These results demonstrate that our instance-based margin generated by IbM2 helps more to learn a robust classification boundary, especially when the training distribution is drastically shifted.

Sensitivity of R. As described in Sec. 4, one original training example is turned into R virtual examples to form the training set $D^{\hat{\epsilon}}$. In this part, we study the effect of the hyperparameter R. As shown in Table 5, when the training shots are highly limited, increasing R significantly improves recognition accuracy. As the number of training shots gets larger, the accuracy difference between a large R ($R = 400$) and a small one ($R = 1$) gradually decreases from 1.6% to 0.2%.

When training samples are extremely scarce, it is difficult to model the correct Gaussian distribution with a few samples in a high-dimensional space. Therefore, increasing R is necessary for a good estimation. When more training shots are available, the need for a large value of sampling times is reduced, hence the accuracy difference between different R becomes smaller, too. Based on these results, we let $R = 200$ in all our experiments.

Method	Shot per Class						
	1	2	3	4	5	8	16
SVM	54.8	64.8	68.2	70.0	70.8	72.8	74.7
IbM2	58.2	66.9	70.2	71.9	72.7	74.3	76.1

Table 4. Top-1 accuracy (%) with different margin-based methods on ImageNet-1K classification with ViT-L/7 from MSN as the pre-trained backbone.

R	Shot per Class						
	1	2	3	4	5	8	16
1	32.9	42.7	47.8	50.8	52.5	55.9	60.1
10	33.7	43.6	48.5	51.4	52.9	56.2	60.3
50	33.8	43.7	48.6	51.5	53.0	56.2	60.3
200	34.4	43.8	48.7	51.5	53.0	56.2	60.3
400	34.5	43.8	48.6	51.5	53.0	56.2	60.3

Table 5. Top-1 accuracy (%) of classification with different R on ImageNet-1K with ViT-S/16 from MoCov3 as the pre-trained backbone.

Sampling Schema	Shot per Class						
	1	2	3	4	5	8	16
-	33.1	42.4	47.2	49.8	51.6	55.2	59.2
Spherical	33.0	42.1	47.1	50.2	51.8	55.3	59.8
Ellipsoidal	34.4	43.8	48.7	51.5	53.0	56.2	60.3

Table 6. Top-1 accuracy (%) of classification with different sampling schemas on ImageNet-1K with ViT-S/16 from MoCov3 as the pre-trained backbone. '-' denotes the simple baseline using only the original training examples.

Ellipsoidal vs. isotropic sampling. We have described two noise sampling strategies to generate virtual examples. The ellipsoidal one is preferred over the isotropic one, and is used in IbM2. In this part, we evaluate the effectiveness of this sampling schema by experimentally comparing these two sampling strategies. The isotropic sampling strategy is denoted as *Spherical* in Table 6.

As shown in Table 6, the accuracy in the ellipsoidal sampling schema is consistently better than its spherical counterpart or the baseline. Moreover, in low-shot cases (≤ 3), sampling with spherical Gaussian slightly degraded the recognition accuracy. The reason may be that the standard deviation of different channels calculated from training features varied a lot, thus simply regarding all channels as independent and identically distributed might make the sampled points significantly shifted from the underlying distribution. However, as the number of training samples increases, both isotropic and ellipsoidal sampling outperformed the baseline.

Based on these observations, we chose to adopt the ellipsoidal noise sampling (Eq. 2) in IbM2.

On what classes can IbM2 help? Finally, we study what classes will benefit from IbM2—will most or only a small portion of classes be improved by IbM2? After the training of the baseline and IbM2 finished, we calculated the class-wise validation accuracy for every class on ImageNet-1K, as

$$\mathrm{ACC}^k = \frac{N_{cor}^k}{N_{cls}^k}\,, \tag{3}$$

where ACC^k is the per-class accuracy (i.e., recall) of the k-th class, N_{cls}^k is the number of test samples class k contains ($N_{cls}^k = 50$ in ImageNet-1K) and N_{cor}^k is the number of correctly classified samples of class k. For every class, we obtain the difference of per-class accuracy (that of IbM2 minus that of the baseline). A positive difference is an accuracy gain and a negative gain is in fact an accuracy loss.

We then sort the baseline's per-class accuracy in the ascending order. Following this order, we rearrange the per-class accuracy gains. The 1000 accuracy gains are divided into 10 histogram bins in the rearranged order, and inside each bin the 100 per-class accuracy gains are averaged to obtain the accuracy gain of that bin. Fig. 3 plots the average accuracy gain in each bin.

We find that IbM2 improves average per-class accuracy in almost every histogram bin (*i.e.*, is above the "0.0" horizon in the y-axis). More detailed numbers reveal that the recognition accuracy of more than 65% categories out of 1000 is improved in *all* shots. Furthermore, Fig. 3 shows a trend: the more accurate a class is, the higher gains IbM2 can achieve over the baseline. That is, when the task is too difficult for the baseline, IbM2 can help but its usefulness is restricted. But, when the baseline is already accurate enough (categorical subset 9 and later), the room for IbM2's further improvement is small again. It is the middle range that IbM2 is the most useful.

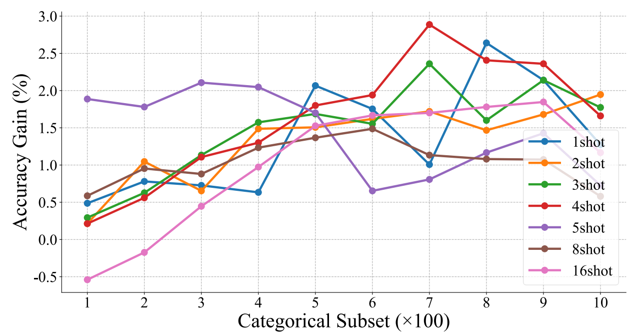

Figure 3. Categorical accuracy gain (%) of IbM2 compared to baseline with ViT-S/16 pre-trained by MoCov3 [10] as the backbone.

6. Conclusions and Limitations

In this paper, we advocated pFSL, a new practical few-shot learning setting. We also proposed IbM2, an instance-based max-margin method to improve few-shot learning. With the technological advancements, it is time to upgrade the traditional FSL settings. We need a simpler, easier-to-evaluate, more challenging and more practical FSL setting, and the proposed pFSL task satisfies these requirements.

To address the challenges associated with the scarcity of training examples, we introduced IbM2. Instead of maximizing the class-level margin, IbM2 is instance-based margin maximization. It achieved significant improvements in both pFSL and the traditional FSL settings consistently. IbM2 is simple and reliable. It introduces only 2 hyper-parameters, and the default values of them worked well in experiments across different architectures and tasks.

As the experiments indicated, IbM2 works the best for the middle range in terms of task difficulty. In a k-shot task when k is extremely small, there is still considerable room for improvement in the pFSL setting. Furthermore, IbM2 freezes the backbone and only learns the linear classifier. In the future, we plan to improve the very small shot cases (*e.g.*, $k = 1$) and better tune the backbone in few-shot learning to push the current limits further.

7. Acknowledgments

This research received substantial support from Jianxin Wu, who wrote IbM2's first working code and whose generous assistance and invaluable guidance were sincerely appreciated throughout our research process. Additionally, we acknowledge the funding provided by the National Natural Science Foundation of China under Grant 62276123 and Grant 61921006.

References

[1] Arman Afrasiyabi, Hugo Larochelle, Jean-François Lalonde, and Christian Gagné. Matching feature sets for few-shot image classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9004–9014, 2022.

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked Siamese Networks for label-efficient learning. In *European Conference on Computer Vision*, page 456–473. Springer, 2022.

[3] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, pages 1–15, 2019.

[4] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, pages 9912–9924, 2020.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised Vision Transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021.

[7] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[9] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, pages 1–16, 2019.

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised Vision Transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9620–9629, 2021.

[11] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-Baseline: Exploring simple meta-learning for few-shot learning. In *IEEE/CVF International Conference on Computer Vision*, pages 9042–9051, 2021.

[12] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, pages 1–20, 2020.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pages 1–21, 2021.

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

[15] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, page 9537–9548, 2018.

[16] Minghao Fu, Yun-Hao Cao, and Jianxin Wu. Worst case matters for few-shot recognition. In *European Conference on Computer Vision*, page 99–115. Springer, 2022.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, 2020.

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

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9726–9735, 2020.

[20] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *IEEE/CVF International Conference on Computer Vision*, pages 235–244, 2021.

[21] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2022.

[22] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9058–9067, 2022.

[23] Seong-Woong Kim and Dong-Wan Choi. Better generalized few-shot learning even without base data. In *AAAI Conference on Artificial Intelligence*, 2023.

[24] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, 2009.

[25] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10649–10657, 2019.

[26] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2967–2976, 2020.

[27] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold Mixup for

few-shot learning. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2207–2216, 2020.

[28] K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

[29] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, pages 1–11, 2017.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

[31] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, pages 1–17, 2019.

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for few-shot learning. In *Advances in Neural Information Processing Systems*, page 4080–4090, 2017.

[33] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation Network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[35] 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 *European Conference on Computer Vision*, page 266–282. Springer, 2020.

[36] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447, 2019.

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for one shot learning. In *Advances in Neural Information Processing Systems*, page 3637–3645, 2016.

[38] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[39] 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*, 53(3):1–34, 2020.

[40] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*, pages 1–13, 2021.

[41] Zhanyuan Yang, Jinghua Wang, and Yingying Zhu. Few-shot classification with contrastive learning. In *European Conference on Computer Vision*, page 293–309. Springer, 2022.

[42] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8805–8814, 2020.

[43] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Few-shot image classification with differentiable Earth Mover's distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12200–12210, 2020.

[44] Zhiyuan Zhao, Qingjie Liu, and Yunhong Wang. Exploring effective knowledge transfer for few-shot object detection. In *ACM International Conference on Multimedia*, page 6831–6839, 2022.