Unsupervised Few-shot Learning via Distribution Shift-based Augmentation

Tiexin Qin, Wenbin Li, Yinghuan Shi, Yang Gao
National Key Laboratory for Novel Software Technology, Nanjing University, China
qtx@smail.nju.edu.cn liwenbin.nju@gmail.com {syh,gaoy}@nju.edu.cn

Abstract

Few-shot learning aims to learn a new concept when only a few training examples are available, which has been extensively explored in recent years. However, most of the current works heavily rely on a large-scale labeled auxiliary set to train their models in an episodic-training paradigm. Such a kind of supervised setting basically limits the widespread use of few-shot learning algorithms, especially in real-world applications. Instead, in this paper, we develop a novel framework called Unsupervised Few-shot Learning via Distribution Shift-based Data Augmentation *(ULDA), which pays attention to the distribution diversity inside each constructed pretext few-shot task when using data augmentation. Importantly, we highlight the value and importance of the distribution diversity in the augmentation-based pretext few-shot tasks. In ULDA, we systemically investigate the effects of different augmentation techniques and propose to strengthen the distribution diversity (or difference) between the query set and support set in each few-shot task, by augmenting these two sets separately (*i.e.*, shifting). In this way, even incorporated with simple augmentation techniques (*e.g.*, random crop, color jittering, or rotation), our ULDA can produce a significant improvement. In the experiments, few-shot models learned by ULDA can achieve superior generalization performance and obtain state-of-the-art results in a variety of established few-shot learning tasks on* mini*ImageNet and* tiered*ImageNet. The source code is available in* https://github.com/WonderSeven/ULDA.

1. Introduction

The ability of learning from limited labeled examples is a hallmark of human intelligence, yet it remains a challenge for modern machine learning systems. This problem recently has attracted significant attention from the machine learning community, which is formalized as few-shot learning (FSL). To solve this problem, a large-scale auxiliary set is generally required to learn transferable knowledge to boost the learning of the target few-shot tasks. Specifically, one kind of FSL methods usually resort to using metric losses to enhance the discriminability of the representation learning, such that a simple nearest neighbor or linear classifier is able to achieve satisfactory classification results [28, 30]. Another kind of FSL methods incorporate the concept of meta-learning and aim to enhance the ability of quickly updating with a few labeled examples [10, 26, 23]. Alternatively, some FSL methods address this problem by generating more examples from the provided ones [11, 5, 6].

Support	Query	(5, 1)	(5, 5)
Tra Aug.	Tra Aug.	32.58	44.40
AutoAugment	AutoAugment	31.53	41.83
Tra Aug.	**AutoAugment**	34.07	47.31
AutoAugment	**Tra Aug.**	**35.37**	**49.16**

Table 1. The comparison with different augmentation methods of N-way K-shot ((N, K) for short) tasks on *mini*ImageNet. Here, Tra Aug. means the way of using traditional augmentation and AutoAugment [8] is a recently developed method. We employ the ProtoNets [28] as the backbone. As observed, the model achieves the better results when using different augmentation techniques to augment the support set and query set, compared with using the same augmentation technique.

Although the aforementioned FSL methods can achieve promising results, most of these methods are fully supervised, which means that they are heavily relying on a large-scale fully labeled auxiliary set (*e.g.*, a subset from ImageNet in previous works [28, 10, 26]). Through this fully labeled auxiliary set, plenty of supervised few-shot tasks (episodes) can be constructed for model training (*i.e.*, episodic-training mechanism [30]). However, in many real-world applications, such a fully supervised condition is relatively severe. It greatly hinder the widespread use of these FSL methods for real applications. Because data labeling for a large-scale dataset is normally time-consuming, laborious, and even very expensive for some domain-professional areas like biomedical data analysis. In contrast, large unlabeled data is easily accessible to many real problems. This gives rise to a more challenging problem, called *unsupervised few-shot learning*, which tries to learn few-shot models by using an unlabeled auxiliary set.

As for unsupervised few-shot learning, only a few works have been proposed. For example, CACTUs [16], as a two-stage method, firstly uses clustering algorithms to obtain pseudo labels, and then trains a model under the common supervised few-shot setting with these pseudo labels. Different from CACTUs, both AAL [2] and UMTRA [18] take each instance as one class and randomly sample multiple examples to construct a support set. Next, they generate a pseudo query set according to the support set by leveraging data augmentation techniques. In this paper, we are more interested in this data augmentation based direction, because it can not only achieve promising results but also can be easily learned in an end-to-end manner. However, we find that the existing data augmentation based methods (*i.e.*, AAL and UMTRA) are sensitive to the selection of augmentation techniques and usually do not contain sufficient regularity for model learning. What's more, they are easily leading to overfitting. We argue that the main bottleneck may raise from the limited distribution diversity between the augmented query set and support set. In a nutshell, the distribution similarity between the query set and support set, caused by a single common data augmentation technique adopted, easily makes the overfitting.

What's the effect if we use different augmentation techniques to augment the support set and query set separately? To figure this out, we perform a simple preliminary experiment (see Table 1). As seen, when using different augmentation techniques, the classification performance can be significantly improved over using the same augmentation technique. Motivated by this observation, we claim to strengthen the distribution difference (or diversity) between the query set and support set, which can alleviate the overfitting of model training and make better generalization performance.

In this paper, we introduce a novel framework named *Unsupervised Few-shot Learning via Distribution Shift-based Data Augmentation* (ULDA). To be specific, our ULDA augments the query set and support set in separate ways, respectively, making a distribution shift between these two sets. The main contributions of our work could be summarized into the following four folds:

- We argue that the distribution diversity between the augmented query set and support set is the key point of data augmentation based methods in unsupervised few-shot learning, for the first time in the literature.
- We propose a *Unsupervised Few-shot Learning via Distribution Shift-based Data Augmentation* (ULDA) framework to strengthen the distribution diversity between the query set and support set by augmenting them separately.
- We develop a new simple augmentation method named *Distribution Shift-based Task Internal Mixing* (DSTIM) to further strengthen the distribution difference between the support and query sets when constructing the pseudo few-shot training tasks.
- Extensive experiments on both *mini*ImageNet and *tiered*ImageNet datasets are conducted to verify the superiority of our proposed framework.

2. Related Work

We briefly review the related work about general few-shot learning, unsupervised learning, and unsupervised few-shot learning.

Few-shot learning (FSL). Few-shot learning aims to learn a new concept on very limited training examples, which has promising practical application value. A vast amount of methods has been proposed in recent years. These methods can be roughly categorized into three classes, *i.e.*, *metric learning-based*, *optimization-based*, and *hallucination-based methods*.

The metric learning-based methods aim to learn discriminative feature representations by using deep metric learning, with the help of intra-class and inter-class constraints [30, 28, 29, 22]. They employ various metric losses (*e.g.*, pairwise loss, triplet loss) to enhance the discriminability of the learned features. The optimization-based methods strive for enhancing the flexibility of the learned model such that it can be readily updated with a few labeled examples [26, 10, 21, 4]. Alternatively, the hallucination-based methods attempt to address the data scarcity problem by directly generating more new examples [33, 1, 6, 5, 7].

Most of these methods train their models under the episodic-training paradigm [30]. They organize a large labeled auxiliary dataset into plenty of mimetic few-shot tasks where each task contains a *support* set and a *query* set. The *support* set is used to acquire task-specific information and the *query* set is used to evaluate the generalization performance of the model. Based on episodic-training, the model expects to learn transferable representations or knowledge, with which, it can generalize to new unseen tasks.

Unsupervised learning. Unsupervised learning methods span a very broad spectrum of approaches. In this section, we only introduce the recent works closely related to our work. A major category of unsupervised learning methods is self-supervised learning (SSL), which aims to learn useful representations with deep neural networks by defining annotation-free pretext tasks, in order to provide a surrogate supervision signal for representation learning. It has been verified that developing pretext tasks like predicting the color of images [3], the relative position of image patches [9, 24], or the random rotation angles of augmented image [12] for representation learning can benefit other downstream tasks. In other words, the deep neural networks are sensitive to these transforms (*i.e.*, color jitter-

ing and rotation, etc.) as they can learn from such kinds of image changes. Moreover, self-supervised learning also shows great regularization effects when integrated with the mainstream method [13].

Unsupervised few-shot learning. Currently, a few works propose *unsupervised few-shot learning* to tackle the huge requirement of a large labeled auxiliary set in supervised few-shot learning. Hsu *et al*. [16] propose CACTUs which uses a clustering algorithm to obtain pseudo labels and then constructs few-shot tasks with these pseudo labels. Differently, Khodadadeh *et al*. [18] and Antoniou *et al*. [2] both propose to randomly sample multiple examples to construct the support set and generate a pseudo query set via data augmentation based on the support set.

Our work belongs to the data augmentation based methods. The main difference is that the existing methods [18, 2] easily suffers from the overfitting problem, while our proposed ULDA can significantly alleviate this problem. This is because there is usually a large distribution similarity between the query set and support set in the existing methods, while our ULDA strengthens a distribution shift between the query set and support set.

3. The Proposed Method

In this section, we first introduce the notations and problem formulation of unsupervised few-shot learning. And next, we discuss the motivation of our work. Finally, we describe the proposed framework, *i.e*., *Unsupervised Few-shot Learning via Distribution Shift-based Data Augmentation* (ULDA), in detail, including each module in ULDA and the extension to optimization-based few-shot learning algorithms.

3.1. Problem Formulation

As aforementioned, the goal of *unsupervised few-shot learning* is to first train a model on a large-scale *unlabeled* auxiliary set D_{train}, and then apply this trained model on a novel labeled test set D_{test}, which is composed of a set of few-shot tasks. Note that, according to the setting of FSL, there are only a few labeled examples (*e.g*., 1 or 5 examples) in each class for each few-shot task in D_{test}. To effectively leverage the unlabeled auxiliary set D_{train} during the training procedure, following the episodic-training mechanism [30], we still try to generate a series of pseudo N-way K-shot tasks (episodes) from D_{train} by using the proposed data augmentation framework. In particular, each pseudo few-shot task is composed of a pseudo support set (for training) and a pseudo query set (for validation). The pseudo support set consists of N classes and K examples per class (*e.g*., $K = 1$ in our paper), termed as $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N\times K}$, whilst the query set $\mathcal{Q} = \{(\hat{x}_1, \hat{y}_2), ..., (\hat{x}_M, \hat{y}_M)\}$ contains M generated examples augmented based on $\mathcal{S}$. At each iteration, the model is trained by one episode (task) to minimize the classification loss on the query set $\mathcal{Q}$ according to the support set $\mathcal{S}$. After tens of thousands of episodes training, the model is expected to reach the convergence and perform well on novel few-shot tasks.

3.2. Motivation from Data Augmentation based Task Construction

Inspired by the literature, we know that the key issue in *unsupervised few-shot learning* is how to construct effect pretext (pseudo) few-shot tasks from an unlabeled auxiliary set D_{train}. If we are able to construct enough pseudo few-shot tasks (which have pseudo labels), and then we can directly learn a few-shot model in a supervised way, by using the episodic-training mechanism [30].

To that end, the latest methods, such as AAL [2] and UMTRA [18], employ the data augmentation techniques to address the above issue. As defined in Section 3.1, a pretext few-shot task usually consists of a pseudo support set $\mathcal{S}$ and a pseudo query set $\mathcal{Q}$. For the construction of the support set $\mathcal{S}$, they randomly sample N unlabeled datapoints as the support examples and randomly assign N labels (classes) for these examples, *i.e*., $y \in \{1, \ldots, N\}$. Next, they augment each image (which has been labeled with pseudo classes) in $\mathcal{S}$ to generate multiple examples within the same class. These augmented examples are taken as the pseudo query set $\mathcal{Q}$, which has the same label space as $\mathcal{S}$. The pretext few-shot tasks constructed in the above way have been verified to be effective and have shown promising results on various datasets [2, 18]. This is because data augmentation can naturally maintain the label of the augmented examples, which can produce reliable pseudo labels for unlabeled examples.

However, the limitation of the above methods is clear. Although the pseudo $\mathcal{Q}$ enjoys the same label space with the pseudo $\mathcal{S}$ by leveraging the nature of data augmentation on label maintenance, $\mathcal{S}$ and $\mathcal{Q}$ have too similar distributions. The reason is that they only adopt one single data augmentation technique for both query and support sets. This limitation will easily leads to a serious overfitting problem in the training stage. As a result, these methods are sensitive to the choice of the data augmentation techniques. This phenomenon can be well explained by our preliminary experiment in Table 1. As seen, when we simply adopt two different data augmentation methods for the query and support sets, the performance can be significantly improved over the single augmentation manner. It means that the distribution diversity between the query set and support set is beneficial to alleviate the overfitting.

This motivates us to study how to increase the distribution difference (diversity) between the pseudo query set and support set, under the principle of maintaining the same label space of these two in this paper. In doing so, the overfitting problem during training can be effectively alleviated

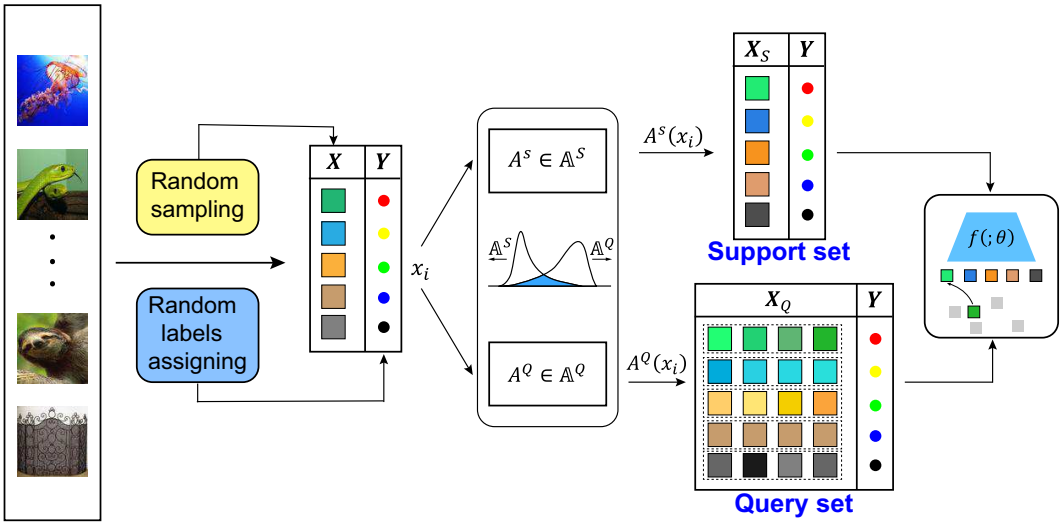

Figure 1. The process of our method ULDA: We start from an unlabeled dataset. After randomly selecting N examples from the dataset, we assign labels to them randomly. We propose to use the distribution shift-based augmentation module to augment these examples separately. Each image in support set is generated by augmentation A^S randomly selected from augmentation set $\mathbb{A}^S$, while each image in query set is generated by augmentation A^Q randomly selected from augmentation set $\mathbb{A}^Q$.

and more robust representations can be learned to tackle the challenging problem of unsupervised few-shot learning.

3.3. The Proposed ULDA Framework

According to the above analysis, we propose a complete framework *Unsupervised Few-shot Learning via Distribution Shift-based Data Augmentation* (ULDA), which intends to learn the representations by maximizing the agreement between support and query sets in the latent space even when there exists a large distribution shift during constructing these two sets. As shown in Figure 1, our framework is composed of the following two major components.

- A *distribution shift-based data augmentation module* that specifically considers the *distribution diversity* in the constructed few-shot tasks during data augmentation. Formally, we form two different sets of augmentation operators for the support and query sets in each constructed few-shot task, which are denoted as $\mathbb{A}^S$ and $\mathbb{A}^Q$, respectively. In general, both the commonly-used augmentation operators (*e.g.*, random crop, color jittering and rotate) and the recently proposed augmentation operators (*e.g.*, AutoAugment [8]) could be the elements of $\mathbb{A}^S$ and $\mathbb{A}^Q$. When we have obtained $\mathbb{A}^S$ and $\mathbb{A}^Q$, the augmentation process is straightforward: 1) randomly sampling multiple data-points as the initial support set, 2) performing augmentation operators in $\mathbb{A}^S$ on these initial support samples to obtain one augmented support set, and 3) similarly performing augmentation operators in $\mathbb{A}^Q$ on these initial support samples to obtain one augmented query set.
- A *metric-based few-shot learning module* that consists of a feature extractor $f(;\theta)$ and a non-parametric classifier. The feature extractor $f(;\theta)$ first learns to map the augmented query and support examples into an appropriate feature space, and then the non-parametric classifier (*e.g.*, kNN) performs classification based on the distances between the query and support examples. Note that our framework allows various alternatives of the metric-based few-shot learning methods. In this paper, we employ ProtoNets [28] to be the backbone as a demonstration of our framework. Moreover, we will discuss an extension to one optimization-based few-shot learning method in Section 3.6.

We randomly sample a mini-batch of N data-points $\{x_1, ..., x_N\}$ from the unlabeled auxiliary set D_{train} as the initial support samples and construct one pseudo few-shot task on augmented examples derived from this initial support set. Specifically, we take each data-point as one class and randomly assign labels for these data-points $X = \{(x_1, 1), ..., (x_N, N)\}$, which is a commonly used strategy in unsupervised learning in the literature [31, 14].

As aforementioned, during the augmentation on the support set, for the i-th initial support image x_i (*i.e.*, the i-th support class), we perform the augmentation operator A_i^S from $\mathbb{A}^S$ ($A_i^S \in \mathbb{A}^S$) on this sample to obtain an augmented support image $A_i^S(x_i)$. Also, for the augmentation on the query set, we randomly select M augmentation operators $A_1^Q, A_2^Q, ..., A_M^Q \in \mathbb{A}^Q$ to augment each initial support image (*i.e.*, each support class) to obtain M augmented query images. So for each few-shot task $\mathcal{T}_z$ consists of an aug-

mented support set $\mathcal{S}$ and an augmented query set $\mathcal{Q}$:

$$\begin{aligned}\mathcal{S} &= \big\{(A_i^S(x_i), i)|i = 1, ..., N\big\},\\ \mathcal{Q} &= \big\{(A_j^Q(x_i), i)|i = 1, ...N\ ,\ j = 1, ..., M\big\},\end{aligned} \tag{1}$$

where $A_i^S(x_i)$ means to perform the sampled operator A_i^S on the i-th initial support image x_i in the initial support set. $A_j^Q(x_i)$ means to perform the sampled operator A_j^Q on the i-th initial support image x_i from the initial support set.

In this work, we emphasize that maintaining a diversity between $\mathbb{A}^S$ and $\mathbb{A}^Q$ (*i.e.*, $\mathbb{A}^S \neq \mathbb{A}^Q$) benefits the performance. This will be thoroughly discussed in the following section. Besides, it has been verified in [18] that the labels in our constructed few-shot tasks maintain class distinctions in most cases which is however important. We summarize the proposed method in Algorithm 1.

3.4. Distribution Shift-based Data Augmentation Module

As analyzed, data augmentation technique plays a key role in aforementioned task construction procedure. However, in common case, the generated tasks do not contain sufficient regularity for model learning as the generated examples are particularly suspect to visual similarity with the original images. To alleviate this, we propose to increase the distribution diversity between the support set and augmented query set with the distribution shift-based data augmentation module which employs separate data augmentation operators to generate the support set and query set.

To systematically study the impact of separate data augmentation, we consider to use both the commonly-used data augmentations and recently proposed augmentations. One type of augmentation involves spatial/geometric transformations, such as random crop and rotation. The other type of augmentation involves appearance transforms, such as color jittering (including brightness, contrast, saturation, hue). Random crop and color jittering are wide used together in few-shot learning, we bind them as traditional augmentation (**Tra Aug.** for short). Typically, for rotation, each image is converted among four directions in $\mathcal{R} = \{0°, 90°, 180°, 270°\}$. The learned AutoAugment method proposed in [8] is also investigated for its promising performance in UMTRA. In addition, we propose a distribution shift-based task internal mixing augmentation strategy which is composed of $\mathrm{TIM}_{\mathrm{sub}}$ and $\mathrm{TIM}_{\mathrm{add}}$ as two augmentation operators. We visualize the augmentations in this work in Figure 2. To understand the efficacy of individual data augmentations $\mathbb{A}^S = \mathbb{A}^Q$ and the importance of augmentation composition $\mathbb{A}^S \neq \mathbb{A}^Q$, we investigate the performance of our framework when applying augmentation in the same or separate manner in Section 4.3.

DSTIM. Inspired by the recent works of generating new examples near the boundary of a classifier in [32, 25], we originally propose a task-level augmentation technique which is termed as *Distribution Shift-based Task Internal Mixing* (DSTIM in shot). DSTIM is a simple yet effective method consisting of two augmentation operators $\mathrm{TIM}_{\mathrm{sub}}$ and $\mathrm{TIM}_{\mathrm{add}}$ which can augment support and query set separately. Besides, these two operators perform convex combination differently between all images in the operated set. To be concrete, for each instance (x_i, y_i) in support (or query) set, we randomly select another instance (x_j, y_j) from the same set and synthesize a new example $(\tilde{x}, \tilde{y})$ as follows:

$$\tilde{x} = \lambda \cdot x_i + (1 - \lambda) \cdot x_j, \quad \tilde{y} = y_i, \tag{2}$$

where for $\mathrm{TIM}_{\mathrm{sub}}$, $\lambda = 0.5 + \max(\lambda, 1 - \lambda), \lambda \sim \mathrm{Beta}(\alpha, \alpha)$, so $\lambda \in [0.5, 1.5]$, while for $\mathrm{TIM}_{\mathrm{add}}$, $\lambda = \max(\lambda, 1 - \lambda), \lambda \sim \mathrm{Beta}(\alpha, \alpha)$, so $\lambda \in [0.5, 1.0]$. When $\lambda > 1.0$, $\mathrm{TIM}_{\mathrm{sub}}$ can generate a new instance by performing subtraction on two images. Note that, x_i and x_j are both input images rather than features. We handle each instance a few times with Eq. (2) to form a new task with more distribution shift between the support set and query set. In this work, we use $\mathrm{TIM}_{\mathrm{sub}}$ to augment images in the support set and $\mathrm{TIM}_{\mathrm{add}}$ for the query set. Basically, DSTIM extends the distribution of raw task by incorporating the prior that if two examples are similar to each other in the original pixel space, then it is possible that they are closer in the feature space. The $\mathrm{TIM}_{\mathrm{add}}$ operator extends the distribution to the margin of two examples whilst the $\mathrm{TIM}_{\mathrm{sub}}$ operator extends to get away from other examples. Besides, as we keep the value of λ more than 0.5, this leads to the synthetic label y_i rather than y_j, so it is an identity-preserved augmentation.

Algorithm 1 The main sampling strategy in ULDA

require: N: class-count, M: meta-test size, Z: episodic number
require: $\mathcal{U}$: unlabeled auxiliary set
require: $\mathbb{A}^S, \mathbb{A}^Q$: two sets of different augmentation operators

for z in 1...Z **do**
  Sample N data-points $x_1...x_N$ from $\mathcal{U}$.
  Randomly assign labels to sampled data-points: $X = \{(x_1, 1)...(x_N, N)\}$.
  Generate support set $\mathcal{S}$ by using operator sampled from $\mathbb{A}^S$ to augment each sample in X.
  Generate query set $\mathcal{Q}$ by using M operator sampled from $\mathbb{A}^Q$ to augment each sample in X.
  $\mathcal{T}_z \leftarrow \{\mathcal{S}, \mathcal{Q}\}$
return $\{\mathcal{T}_z\}$

3.5. Metric-based Few-shot Learning Module

One of the major category methods to deal with the few-shot problem is metric-based few-shot learning methods, which aim to enhance the discriminability of feature representations of images via deep metric learning. The main

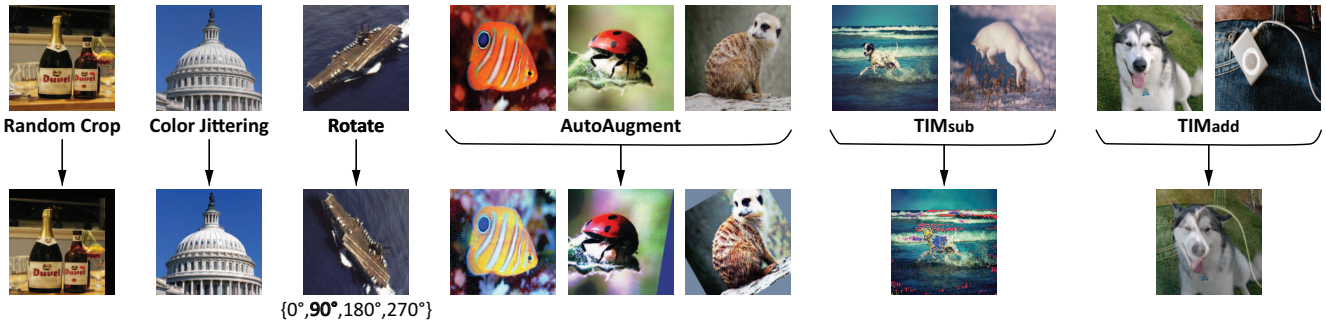

Figure 2. Illustrators of the employed augmentation techniques in this work. Top: Original images, Bottom: augmented images, transformed by an augmentation operator.

component of these algorithms is a feature extractor $f(\cdot;\theta)$, which is a convolution neural network with parameters θ. Given an episode (few-shot task) $\mathcal{T}_z$, the feature extractor will map each image x_i in $\mathcal{T}_z$ into a d-dimensional feature $f(x_i;\theta)$ (or metric space). In a learned metric space, the images in query set are close to a labeled image in support set when they share similar semantic information [29, 22]. Normally, Euclidean distance or cosine distance is employed to measure the similarity between two examples. As the feature extractor plays a key role in the final results, the diversity of the augmented examples is crucial to exhibit the feature extractor to extract discriminative features. Our proposed ULDA framework indeed increases the diversity via modifying the distribution of the images in support set and query set to guarantee the diversity. To formalize this effect, we incorporate this module with a representative few-shot learning algorithm, ProtoNets [28], which is also the backbone of our framework.

Given a N-way K-shot episode $\mathcal{T}_z$, ProtoNets computes the "prototype" via averaging features for each class in support set with the feature extractor $f(;\theta)$:

$$\mathbf{p}_i = \frac{1}{K}\sum_{x\in\mathcal{S}^i} f\Big(A^S(x);\theta\Big)\,, \tag{3}$$

where $\mathcal{S}^i = \{x|(x,y)\in\mathcal{S}, y=i\}$ and $A^S \in \mathbb{A}^S$. These "prototypes" are used to build a simple similarity-based classifier. Then, given a new image x_q from query set, the classifier outputs a normalized classification score computed with Euclidean distance for each class i:

$$C^i\big(f(x_q;\theta)\big) = \frac{\Big(f\big(A^Q(x_q);\theta\big)-\mathbf{p}_i\Big)^2}{\sum_{j=0}^{N}\Big(f\big(A^Q(x_q);\theta\big)-\mathbf{p}_j\Big)^2}\,, \tag{4}$$

where $A^Q \in \mathbb{A}^Q$. So, the image x_q will be classified to its closest prototype. The loss function for updating the parameter θ is formalized as:

$$\mathcal{L} = \sum_{\mathcal{T}_z\sim\mathcal{T}}\sum_{(x_q,y_q\in\mathcal{Q})} -\log C^{y_q}\big(f(x_q;\theta)\big). \tag{5}$$

Note that, the distance between $f(A^Q(x_q);\theta)$ and its corresponding prototype will not change if we keep $\mathbb{A}^S = \mathbb{A}^Q$. And this makes no sense to secure the discriminability of the feature extractor. Besides, as we use rotation as an augmentation technique, we also incorporate with a self-supervised loss to predict the rotation angle where the detail could be referred to literature [13].

3.6. The Extension to Optimization-based Algorithms

Optimization-based algorithms belong to a more common category for few-shot learning, striving for enhancing the flexibility of the model such that it can be readily updated using a few labeled examples. These methods either aim to optimize the meta-learned classifier or adaptive neural network generation using the support set. See Section 2 for more details. Even sink to the support set, query set is also employed to judge the update of model parameters [10]. So it is hopeful that our framework will also work when incorporated with optimization-based algorithms. We use a recently proposed method, MetaOptNet [21] to validate the scalability of our framework.

4. Experiments

In this section, we detail the experimental setting and compare our ULDA with state-of-the-art approaches on two challenging datasets, *i.e.*, *mini*ImageNet [30] and *tiered*ImageNet [27], which are widely used in the literature. We did not include Omniglot [20] in our evaluation since the performance of Omiglot is usually regard to be saturated.

4.1. Experimental Setting

Datasets. *mini*ImageNet [30] and *tiered*ImageNet [27] are introduced as benchmark datasets.

- The ***mini*ImageNet** is the most popular benchmark in the field of few-shot learning, which was introduced in [30]. The dataset is composed of 100 classes selected from ImageNet [19], and each class contains

Algorithm (N, K)	Clustering	(5, 1)	(5, 5)	(5, 20)	(5, 50)
Training from scratch	N/A	27.59±0.59	38.48±0.66	51.53±0.72	59.63±0.74
k_{nn}-nearest neighbors	DeepCluster	28.90±1.25	42.25±0.67	56.44±0.43	63.90±0.38
linear classifier	DeepCluster	29.44±1.22	39.79±0.64	56.19±0.43	65.28±0.34
MLP with dropout	DeepCluster	29.03±0.61	39.67±0.69	52.71±0.62	60.95±0.63
cluster matching	DeepCluster	22.20±0.50	23.50±0.52	24.97±0.54	26.87±0.55
AAL-ProtoNes [2]	N/A	37.67±0.39	40.29±0.68	-	-
AAL-MAML++ [2]	N/A	34.57±0.74	49.18±0.47	-	-
CACTUs-ProtoNets [16]	BiGAN	36.62±0.70	50.16±0.73	59.56±0.68	63.27±0.67
CACTUs-MAML [16]	BiGAN	36.24±0.75	51.28±0.68	61.33±0.67	66.91±0.68
CACTUs-ProtoNets [16]	DeepCluster	39.18±0.71	53.36±0.70	61.54±0.68	63.55±0.64
CACTUs-MAML [16]	DeepCluster	39.90±0.74	53.97±0.70	**63.84±0.70**	**69.64±0.63**
UMTRA [18]	N/A	39.93±–	50.73±–	61.11±–	67.15±–
UFLST [17]	DBSCAN	33.77±0.70	45.03±0.73	53.35±0.59	56.72±0.67
ULDA-ProtoNets(ours)	N/A	**40.63±0.61**	**55.41±0.57**	63.16±0.51	65.20±0.50
ULDA-MetaOptNet(ours)	N/A	**40.71±0.62**	**54.49±0.58**	**63.58±0.51**	**67.65±0.48**
Supervised (Upper Bound)					
ProtoNets	N/A	46.56±0.76	62.29±0.71	70.05±0.65	72.04±0.60
MAML	N/A	46.81±0.77	62.13±0.72	71.03±0.69	75.54±0.62

Table 2. Unsupervised few-shot classification results in % of N-way K-shot (N,K) learning methods on the *mini*ImageNet. All results are averaged over 1000 tasks which are randomly constructed from test set. "-" means the results are not reported in their source papers.

600 images with the size of 84×84. We follow the data split proposed by [26], which splits the total 100 classes into 64 classes for training, 16 classes for validation and 20 classes for test. The validation set is only used for picking the best model during training.

- The ***tiered*ImageNet** consists of 608 classes (779,165 images) selected from ImageNet [19]. This dataset is grouped into 351 training classes, 97 validation classes and 160 novel test classes. Each image is resized to 84×84.

Backbone network. We employ a four-layers convolutional neural network, which is widely adopted in the few-shot literature as the feature extractor backbone. Each layer comprises a 64 filters (3×3 kernel) convolutional layer, a batch normalization layer, a ReLU layer and a 2×2 max-pooling layer. All input images are resized to $84 \times 84 \times 3$, and the output features are flattened into 1600-dimensional vectors as the same setting as these previous works [28, 10].

Training strategy. We conduct 5-way 1-shot classification tasks during meta-training on the aforementioned datasets. We randomly sample and construct 10,000 tasks in each epoch and train our networks for a total of 60 epochs. All backbone networks are optimized by SGD with Nesterov momentum of 0.9 and weight decay of 0.0005. The initial learning rate is set as 0.001 and multiplied by 0.06, 0.012, 0.0024 after 20, 40, and 50 epochs, respectively. We conduct all the experiments on GTX 2080Ti. Note that, for a fair comparison, the hyper parameters in all of these methods are kept to be the same.

Parameter setup. In Eq. (2), we empirically set $\alpha = 0.8$ for $\mathrm{TIM}_{\mathrm{sub}}$ and $\alpha = 0.6$ for $\mathrm{TIM}_{\mathrm{add}}$. Our model is robust with different values of α. Thus, we set it in a slightly different manner following our distribution-diversity argument.

4.2. Unsupervised Few-shot Learning Results

To verify the effectiveness of our approach for unsupervised few-shot learning, we compare the proposed ULDA framework with two baselines (ProtoNet [28] and MetaOptNet [21]) and other state-of-the-art methods in various settings. Moreover, to make our results more convincing, we randomly sample 1,000 episodes from the test set for evaluation. Also, we take the top-1 mean accuracy as the evaluation criterion and repeat this process five times. Besides, the 95% confidence intervals are also reported.

Results on *mini*ImageNet. The experimental results on *mini*ImageNet are summarized in Table 3.6. Our ULDA achieves state-of-the-art results on both 5-way 1-shot and 5-way 5-shot settings and competitive results on 5-way 20-shot and 5-way 50shot settings. Besides, our ULDA performs much better than the baseline method, *i.e.*, *training from scratch*. Importantly, the results of our ULDA are very close to the results of supervised meta-training approaches with a labeled auxiliary set, *i.e.*, ProtoNets and MAML. Note that, when using the same few-shot learning algorithm (*i.e.*, ProtoNets), our ULDA framework outperforms all other methods across different classification tasks. Compared with CACTUs, our ULDA gains 1.45%, 2.05%, 1.62%, 1.93% performance boost under 5-way 1-shot, 5-shot, 20-shot and 50-shot settings, respectively. As

Algorithm (N, K)	Clustering	(5, 1)	(5, 5)	(5, 20)	(5, 50)
Training from scratch	N/A	26.27±1.02	34.91±0.63	38.14±0.58	38.67±0.44
ULDA-ProtoNets(ours)	N/A	**41.60**±0.64	**56.28**±0.62	**64.07**±0.55	**66.00**±0.54
ULDA-MetaOptNet(ours)	N/A	**41.77**±0.65	**56.78**±0.63	**67.21**±0.56	**71.39**±0.53
Supervised (Upper Bound)					
ProtoNets	N/A	46.66±0.63	66.01±0.60	77.62±0.46	81.70±0.44
MetaOptNet	N/A	47.32±0.64	66.16±0.58	77.68±0.47	80.61±0.48

Table 3. Unsupervised few-shot classification results in % of N-way K-shot (N,K) learning methods on the *tiered*ImageNet. All results are averaged over 1000 tasks which are randomly constructed from test set.

$\mathbb{A}^S$	$\mathbb{A}^Q$	KL	FID	(5, 1)	(5, 5)
Tra Aug.	Tra Aug.	-0.73	16.07	32.58±0.49	44.40±0.49
AutoAugment	AutoAugment	-0.61	19.52	31.53±0.49	41.83±0.53
Tra Aug.	AutoAugment	-	-	34.07±0.51	47.31±0.52
AutoAugment	Tra Aug.	-0.71	133.97	35.37±0.53	49.16±0.52
AutoAugment	Rotation	-0.46	183.06	39.18±0.58	53.30±0.58
AutoAugment	Rotation+Tra Aug.	**-0.38**	172.22	39.28±0.59	53.55±0.58
AutoAugment	Rotation+TIM_{add}	-0.57	181.14	39.42±0.57	53.87±0.58
AutoAugment+TIM_{sub}	Rotation+TIM_{add}	-0.55	185.27	39.52±0.58	54.26±0.57
AutoAugment+TIM_{sub}	Rotation+Tra Aug.+TIM_{add}	-4.03	**202.42**	**39.64**±0.60	**54.37**±0.58

Table 4. The comparison with different augmentation methods. The results in % of N-way K-shot (N, K) are reported.

for CACTUS, a two-stage model, it uses clustering algorithms to assign pseudo labels before constructing tasks, the quality of these pseudo labels will limit the final results, while our ULDA does not have this limitation. Besides, when compared with AAL, which is the closest work to our ULDA, our ULDA achieves 2.96% and 15.12% performance boost for 5-way 1-shot and 5-way 5-shot, respectively.

Results on *tiered*ImageNet. We turn to *tiered*ImageNet, a more challenging dataset, that contains more complex classes and examples than *mini*ImageNet. Since the recent unsupervised few-shot leaning methods (*i.e.*, CACTUs, UMTRA) did not report experimental results on this dataset, we only compare our methods with the baseline method *training from scratch*. The results are illustrated in Table 4.2. As seen, our ULDA performs much better than learning from scratch and slightly weaker than the supervised methods.

4.3. Ablation Study

Effectiveness of different blocks. Our method achieves the new state-of-the-art result in unsupervised few-shot learning literature. To analyze how much each block (Separate Aug., Rotate, Self-supervision loss and DSTIM) contributes to the ultimate result, we conduct a serial of ablation studies. All results are shown in Table 4.3. Note that we use the best combination of different modules as our final model. The *Separate Aug.* means performing AutoAugment on support images and performing traditional augmentation on query images. As seen, the *self-supervision loss* (SSL loss for short) contributes a little to the final result, but the augmentation method during developing a pretext task (*i.e.*, Rotation here) leaves a great performance boost. As the convolutional neural networks are sensitive to image rotation, we use rotation to augment query set can strengthen the distribution difference between query and support set.

Separate Aug.	Rotate	SSL loss	DSTIM	(5, 1)	(5, 5)
				32.58±0.49	44.40±0.49
✓				35.37±0.53	49.16±0.52
	✓			34.85±0.52	46.88±0.57
✓	✓			39.28±0.59	53.55±0.58
✓	✓	✓		40.32±0.61	54.91±0.59
✓	✓	✓	✓	**40.63**±0.61	**55.41**±0.57

Table 5. **Ablation Study.** The various blocks we employ to improve the test accuracy (%) on 5-way *mini*ImageNet benchmark.

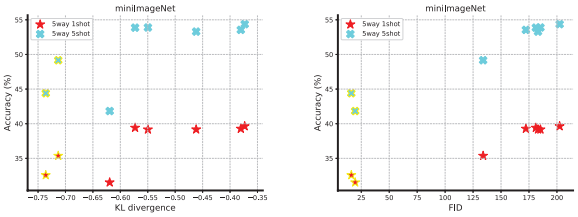

Figure 3. The result with different distribution margin between support and query set. The geometries with yellow outline are augmented with the same method.

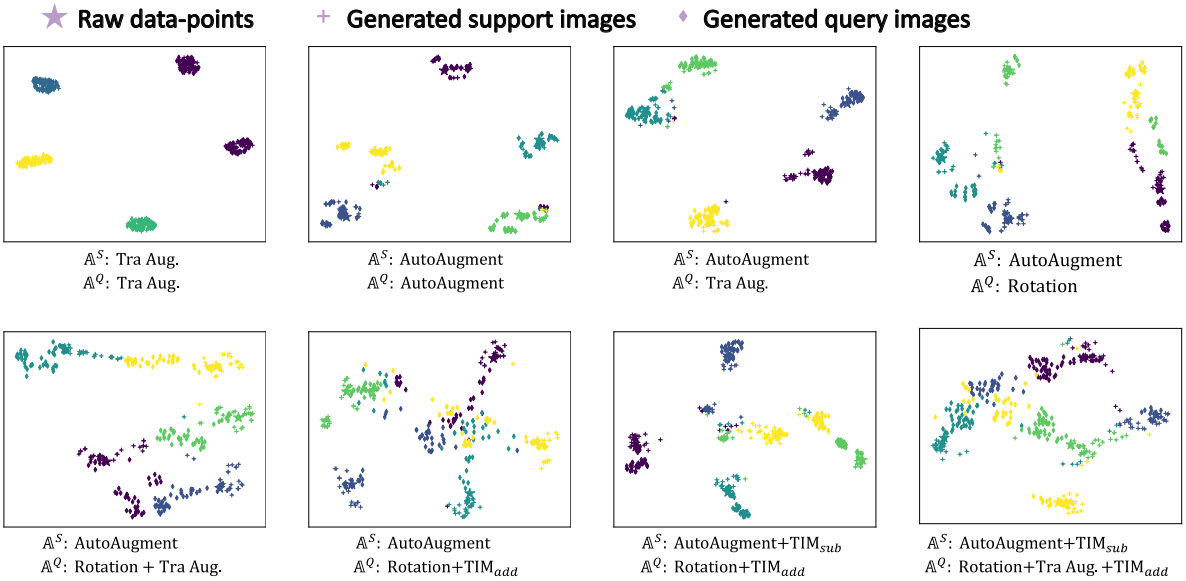

Figure 4. Visualization of feature transformations in generated support images and query images. Same color means generated from the same data-point. The generated images own more diversity and there exist little overlap between generated support images and query images via our approach. Zoom in for best visual effect.

Effectiveness of distribution shift-based augmentation module. Despite the promising results achieved by our entire framework, we also expect to know how it works, especially the relationship between the distribution shift in generated two sets and the final results. With this purpose, we employ the aforementioned augmentation techniques (*i.e.*, random crop, color jittering, rotation, AutoAugment and our proposed DSTIM) and combine them in various ways to produce these two sets with different distribution shift. Besides, we use Kullback-Leibler divergence (KL divergence) and Fréchet Inception Distance (FID) [15] to evaluate the distribution difference. The results are illustrated in Figure 4.3, detailed data can be referred in Table 4.2. We can draw the conclusion from these results that:

1. The models tend to perform much better when meta-trained on the tasks in which large distribution difference exists in the generated query set and support set.

2. It can be observed that augmenting the query set and support set separately usually works well.

3. Combining different augmentation methods can generate more diverse examples to strengthen the distribution difference.

In order to intuitively felt the effect of our framework, we also visualize the augmentation effect in feature space in Figure 4.3. We find that, when augmenting support set and query set with the same augmentation techniques, the generated query set gathers tightly around support set, and these tend to exist heavy overlap in these augmented data-points. However, with our approach, the generated examples share more diversity and more distribution difference between the support set and query set.

5. Conclusions

In this paper, we present an unsupervised few-shot learning framework that aims to increase the diversity of generated few-shot tasks based on data augmentation. We argue that when strengthening the distribution shift between the support set and query set in each few-shot task with different augmentation techniques can increase the tasks' ability for model training. A serial of experiments have been conducted to demonstrate the correctness of our finding. We also incorporate our framework with two representative few-shot learning algorithms, *i.e.*, ProtoNets and MetaOptNet, and achieve the state-of-the-art results across a variety of few-shot learning tasks established on *mini*ImageNet and *tiered*ImageNet.

Future works include: (1) the extension to common unsupervised learning, (2) the incorporation with GAN-based data augmentation techniques to directly increase the distribution shift by training a generator with this aim, (3) the implementation to the related vision applications.

References

[1] Amit Alfassy, Leonid Karlinsky, Amit Aides, Joseph Shtok, Sivan Harary, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. Laso: Label-set operations networks for multi-label few-shot learning. In *CVPR*, 2019.

[2] Antreas Antoniou and Amos Storkey. Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. In *ICML*, 2019.

[3] Nikola Banić, Karlo Koščević, and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017.

[4] Weiyu Chen, Yencheng Liu, Zsolt Kira, Yuchiang Frank Wang, and Jiabin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[5] Zitian Chen, Yanwei Fu, Kaiyu Chen, and Yu-Gang Jiang. Image block augmentation for one-shot learning. In *AAAI*, 2019.

[6] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019.

[7] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. In *IEEE Transactions on Image Processing*, 2019.

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pages 113–123, 2019.

[9] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015.

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

[11] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *NIPS*, pages 975–985. 2018.

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

[13] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *ICCV*, pages 8059–8068, 2019.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, pages 6626–6637, 2017.

[16] Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. *ICLR*, 2019.

[17] Zilong Ji, Xiaolong Zou, Tiejun Huang, and Si Wu. Unsupervised few-shot learning via self-supervised training. *arXiv preprint arXiv:1912.12178*, 2019.

[18] Siavash Khodadadeh, Ladislau B?l?ni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification. In *NIPS*, 2019.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[20] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

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

[22] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Gao Yang, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *CVPR*, 2019.

[23] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017.

[24] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *ECCV*, 2016.

[25] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *ICCV*, 2019.

[26] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[27] 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 preprint arXiv:1803.00676*, 2018.

[28] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *NIPS*, 2017.

[29] Flood Sung, Yongxin Yang, Li Zhang amd Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[30] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *NIPS*, 2016.

[31] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *CVPR*, 2018.

[32] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minization. In *ICLR*, 2018.

[33] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *CVPR*, 2019.