This is a repository copy of *Balancing feature alignment and uniformity for few-shot classification*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/205361/

Version: Accepted Version

Article:

Yu, Y., Zhang, D., Ji, Z. et al. (3 more authors) (2023) Balancing feature alignment and uniformity for few-shot classification. IEEE Transactions on Image Processing, 34. pp. 4456-4469. ISSN: 1057-7149

https://doi.org/10.1109/TIP.2023.3328475

© 2023 The Authors. Except as otherwise noted, this author-accepted version of a journal article published in IEEE Transactions on Image Processing is made available via the University of Sheffield Research Publications and Copyright Policy under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Balancing Feature Alignment and Uniformity for Few-Shot Classification

Yunlong Yu, *Student Member, IEEE,* Dingyi Zhang, Zhong Ji, *Senior Member, IEEE,* Xi Li, *Senior Member, IEEE,* Jungong Han, *Senior Member, IEEE,* Zhongfei Zhang, *Fellow, IEEE*

***Abstract*—Given a few samples for each novel class, Few-Shot Learning (FSL) aims to correctly recognize new samples from the novel classes, by learning a model from the base classes. The existing methods focus on learning transferable knowledge from the base classes by maximizing the information between the feature representations and their labels, which may suffer from the *supervision collapse* issue due to the bias toward the base classes. In this paper, we address this issue by preserving the intrinsic structure of the data to learn a generalized model for the novel classes. Following the InfoMax principle, we maximize both the mutual information (MI) between the samples and their feature representations and the MI between the feature representations and their class labels, leading to a balance between discrimination and generalization for the feature representations. Specifically, we maximize the MI of samples and their representations with two low-bias estimators to perform feature representation learning, an estimator between a pair of intra-class samples, and an estimator between a sample and its augmented views. We formulate the whole idea into a united framework that perturbs the feather embedding space by both distilling knowledge between class-wise pairs and enlarging the feature representation diversity. Through extensive experiments on a variety of popular FSL benchmarks, the proposed approach achieves comparable performances with state-of-the-art competitors, including 69.53% accuracy on the miniImageNet dataset and 77.06% accuracy on the CIFAR-FS dataset under the 5-way 1-shot task.**

***Index Terms*—Few-Shot Learning, Mutual Information, Feature Representation, Knowledge Distillation, and Self-Supervised Augmentation.**

I. Introduction

THe availability of large amounts of annotated data has promoted deep learning techniques to advance significantly in the computer vision areas over the last decade. Despite the dramatic advances, in actual computer vision applications, a large amount of annotated data cannot be easily obtained. It is thus imperative to develop methods for learning from a few training samples. Research related to this subject is usually termed as Few-Shot Learning (FSL) [1], [2], [3], [4] that aims at accurately recognizing samples from the target classes with a few samples per class.

To address this task, the meta-learning-based approaches [1], [5], [6], [7], [8] have been extensively studied and have dominated FSL areas in recent years. These approaches aim at training meta-models that are agnostic to different FSL tasks from a collection of mimic FSL tasks formed with the base data. Once trained, the meta-models are applied to solving new FSL tasks sampled from the disjoint target classes. Recently, some approaches [9], [10], [11], [12], [13] have demonstrated that training a basic classification model with the cross-entropy loss on the base classes performs very competitively for the downstream FSL tasks. These approaches attempt to train an effective feature extractor via either fine-tuning [10], [13] with an episodic sampling way or retraining another network [11] based on a pre-trained classification network. However, most existing works train the feature extractor by overemphasizing the correct predictions of base samples, which will be biased toward the base classes and suffer from *supervision collapse* [14], [15], [16] where the trained model drops any information that is not necessary for predicting the training classes, including the information that may be necessary for transferring to novel classes. From an information-theoretic perspective, the existing methods maximize the mutual information (MI) between the feature representations and their associated class labels but neglect to maximize MI between the feature representations and the raw input samples.

As the InfoMax principle [17] indicates, maximizing MI between the feature representations and the input samples would preserve more information about the raw input samples, which is beneficial for learning more generalized information. However, maximizing MI between the feature representations and the input samples is intractable. Thus, some works attempt to find MI estimators to address the supervision collapse issue. Knowledge Distillation (KD) [18], [19] and Label Smoothing (LS) [20], [21] are two popular strategies to remedy the model from the collapse issue by maximizing the mutual information between the feature representations and the supervised representations. As shown in Fig. 1(a)(b), KD obtains the supervised representations from a pre-trained teacher model while LS artificially designs a soft representation. [11] applies the knowledge distillation framework for FSL and demonstrates that softening the hard one-hot label would benefit learning more transferable feature embeddings for novel classes. However, traditional KD requires training a teacher network in advance, which is limited due to the computation costs. Though the soft labels provided with LS

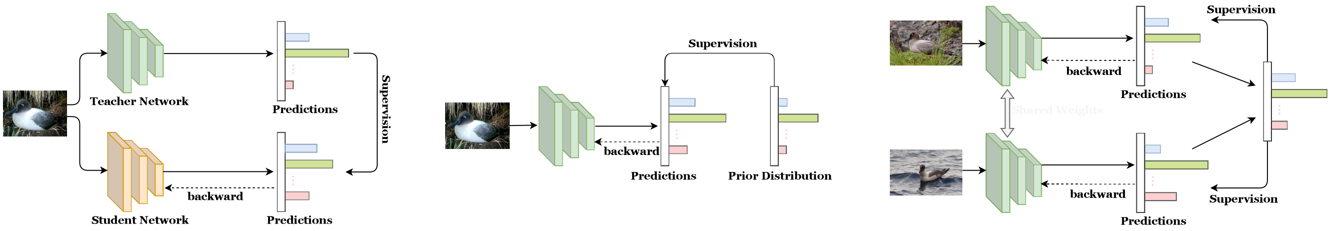
Y. Yu and D. Zhang are with the College of Information Science and Electronic Engineering, Zhejiang University, and also with Zhejiang Provincial Key Laboratory of Information Processing, Communication, and Networking (IPCAN), Hangzhou 310007, China (e-mail: {yuyunlong, dyzhang}@zju.edu.cn)

Z. Ji is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: jizhong@tju.edu.cn)

Xi Li is with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: xilizju@zju.edu.cn)

Jungong Han is with the Department of Computer Science, Aberystwyth University, U.K. (e-mail: jungonghan77@gmail.com)

Z. Zhang is with the Computer Science Department, Watson School, State University of New York Binghamton University, Binghamton, NY, 13902 USA (e-mail: zhongfei@cs.binghamton.edu).



(a) Knowledge Distillation

(b) Label Smoothing

(c) Teacher-free Knowledge Distillation

Fig. 1: The basic illustration of traditional two-stage knowledge distillation, label smoothing, and the proposed computation-free knowledge distillation. Three strategies differ in the way of providing soft labels. The knowledge distillation provides the soft labels with a pre-trained teacher, the label smoothing technique gives the soft labels from a prior distribution, and the proposed strategy produces the soft labels with the other sample from the same class.

methods could relieve the overfitting issue, they hardly contain the semantic relationships among classes, thus bringing in little benefits for or even hurting the feature generalization on the novel classes.

In this paper, we address the *supervision collapse* issue by maximizing two low-bias estimators of the MI between the raw samples and their feature representations, an estimator between a pair of intra-class samples, and an estimator between a sample and its augmented views. For the estimator between intra-class pairs, we propose to mutually distill knowledge between a pair of intra-class samples, as shown in Fig. 1(c). Compared with the traditional two-stage KD methods, our one-stage framework is more efficient as it requires no teacher network in advance but only with a computation-free pairwise sampling strategy, and has the additional capability of relieving the overconfidence risk as it dynamically changes the soften label for each sample during training. We theoretically show that mutually distilling knowledge between intra-class pairs would encourage the model to pay more attention to the hard samples, which would benefit the model generalization as training the hard samplers requires capturing more valid patterns. For the estimator between a sample and its augmented views, we perturb the feature embedding space to prevent the intra-class feature representations from collapsing into a feature embedding by diversifying the feature representations based on the augmented way. We formulate the whole idea into a united framework to boost the model generalization ability while keeping its discriminative ability by balancing the alignment and uniformity in the feature embedding space.

We evaluate our approach on four FSL benchmarks. Though simple, our approach obtains 69.53% accuracy on miniImageNet [22] and 77.06% accuracy on CIFAR-FS [23] under the 5-way 1-shot task, which is comparable or even better over the state-of-the-art competitors.

To summarize, the highlights of our work are:

1) We propose an effective FSL approach that learns generalized feature representations by balancing the alignment and uniformity in the feature embedding space. Our method maximizes both the MI between the feature representations and their class labels and two low-bias estimators of the MI between the feature representations and their raw samples.
2) We formulate the whole idea into an easy-to-implement framework that smooths the feature distributions by mutually distilling knowledge in a pair of samples from the same class and perturbs the feature embedding space by diversifying the feature representations based on the augmentation way.
3) We theoretically reveal that mutually distilling knowledge between intra-class pairs would encourage the model to pay more attention to the hard instances, leading to a better generalization since learning the hard instances could learn more valid patterns.

The remaining sections are organized as follows. Sec. II describes the related work. Sec. III presents the proposed method, including the theoretical analysis, and the framework for training the model. Sec. IV provides extensive experiments and evaluations, followed by the conclusion in Sec. V.

II. Related Work

A. Few-Shot Learning

The existing FSL approaches are roughly divided into three categories, *i.e.*, gradient-based approaches, data-augmentation approaches, and metric-based approaches.

Most of the gradient-based approaches are categorized into the meta-learning paradigm as they aim at learning the task-agnostic knowledge via training a collection of few-shot tasks sampled from the base classes. On one hand, some studies [1], [24], [25] attempt to learn a suitable initialization of the model parameters, aiming at quickly adapting to new few-shot tasks within a few iterations. On the other hand, some approaches try to learn a new optimizer for replacing the traditional stochastic gradient descent optimizer with an LSTM-based meta-learner [26] or an external memory [27].

The data-augmentation approaches attempt to address the data scarcity issue by generating or hallucinating additional data for target classes. These approaches try to hallucinate data from a few samples of the target classes either with the variational models learned from the base classes [28], [29] or directly with a generator trained in an adversarial way [8], [30]. Besides, some approaches try to augment the target data via performing quality-controlled image distortions [31] or weaving a self-supervision strategy into the training objective [32], [33].

A majority of metric-based approaches lead to the state-of-the-art for the FSL tasks. These approaches aim at learning a general distance metric [5], [6] that can be used to compare the similarities between different samples. Most existing methods build upon this idea to learn an effective feature extraction model via directly constructing constraints on the feature embeddings with a sophisticated meta-learning strategy [22], [34]. However, recent works [10], [9], [11], [12] have revealed that either training a simple classification network on base classes or applying the fine-tuned pipeline with the pre-trained model could beat the most existing complicated meta-learning designed approaches. Although not performing a direct constraint on feature embeddings, they could capture more generalized features among all the base classes instead of a subset of classes. Our approaches also fall into this category. The most related works to ours are [10], [11]. Both these studies contain two stages that first train a classification model with the base data and then fine-tune the model with meta-learning [10] or re-train another model distilled with the pre-trained model [11]. In contrast, our approach consists of a one-stage training process to regularize the intra-class feature embedding distributions, without a pre-trained powerful teacher network.

B. Knowledge Distillation

Knowledge Distillation (KD), as an important model compression technique traced back decades ago, has been proven to be effective for transfer learning tasks. It is re-popularized by Hinton *et al.* [18], which has shown that the knowledge can be distilled and transferred to a student network from a large ensemble of teacher models. Since then, KD is widely explored and applied in many machine learning tasks. The existing techniques on KD are roughly categorized into two groups.

The first group mainly concentrates on the way to mine dark knowledge. These approaches attempt to transfer knowledge from the teacher network to the student network via either similarity constraint [35], relation alignment [36], or attention maps [37], [38]. The other group seeks to find a good teacher network to distill the student network, including the ensemble of teacher models [39], intermediate-performance teacher models [40], and "tolerant" teacher [41] that selects less peaked predictions for distillation. Without a pre-trained teacher network, deep mutual learning [42] introduces two parallel networks to mutually distill each other at the same time. Neither requiring a pre-trained model nor a parallel network, the proposed approach attempts to achieve knowledge distillation with the cooperation between pair-wise samples in a class-wise manner, thus being more efficient and easy to implement. A similar idea has been explored in [43] that uses the class prediction of the intra-class sample to supervise the sample training. Differently, our strategy mutually distills each other in an input intra-class pair instead of fixing the supervision.

There are some attempts at integrating knowledge distillation into the FSL framework. For example, RFS-distill [11] first introduces the knowledge distillation into FSL that follows the traditional two-stage training pipeline. Following [11], [44] combine self-supervision into the learning process while [45] performs a self-knowledge distillation where both the teacher network and student network share the same architecture. These methods perform knowledge distillation with a two-stage pipeline, which requires training a teacher network in advance. In contrast, our method is a teacher-free knowledge distillation strategy, which is more efficient.

C. Self-Supervised Learning for FSL

Recently there have been some attempts [33], [32], [46], [47] at integrating self-supervised learning into the framework of FSL. These approaches benefit from sharing inductive bias between the main task (FSL) and auxiliary self-supervised tasks, thus boosting the FSL performance. However, these works struggle in providing an in-depth understanding of why self-supervised learning has positive effects on FSL. Our work takes a step forward, revealing the effects of self-supervised augmentation on perturbing the feature embedding space for improving the model's generalization from the perspective of mutual information.

III. Methodology

A. Problem Definition

Given a set of base classes $\mathcal{C}_{base}$ with a large number of labeled samples for each class, FSL is to classify the test samples into the candidate target classes $\mathcal{C}_{target}$, with only a few support samples being provided for each target class. Note that the base classes and the target classes are disjoint in the label space, *i.e.*, $\mathcal{C}_{base} \cap \mathcal{C}_{target} = \varnothing$. FSL is to train a model with the base classes and generalize on the few-shot tasks. Specifically, an N-way K-shot task consists of a support set and a query set, of which the support set contains N classes with K samples for each class while the query set contains the same N classes with Q samples for each class.

B. Proxy-based Baseline for FSL

Supervised training aims to reduce the classification loss on the base classes, producing a feature extractor for the novel classes to perform FSL tasks with a predefined distance metric. For a sample x from the base set $\mathcal{C}_{base}$, its predictive probability for the i-th class is computed as:

$$P_i(x) = \frac{\exp(z_i/\tau)}{\sum_{m=1}^{M} \exp(z_m/\tau)}, \tag{1}$$

where $\mathbf{z} = [z_1, ..., z_i, ..., z_M] = h_{cls} \circ g(x)$ denotes the logit vector of sample x, *i.e.*, the class predictions before the softmax function, g and h_{cls} denote the feature extractor and the classification head, respectively. $\tau > 0$ is the temperature scaling parameter, which controls the smoothness of the distribution. M denotes the base class number.

To ensure the consistency between the training and test processes, we take the parameters of the classification head as class proxies, each of which denotes the visual prototype of the corresponding base class. The class prediction for sample x is obtained with:

$$\mathbf{z} = d\left(g(x);\ \mathbf{A}\right), \tag{2}$$

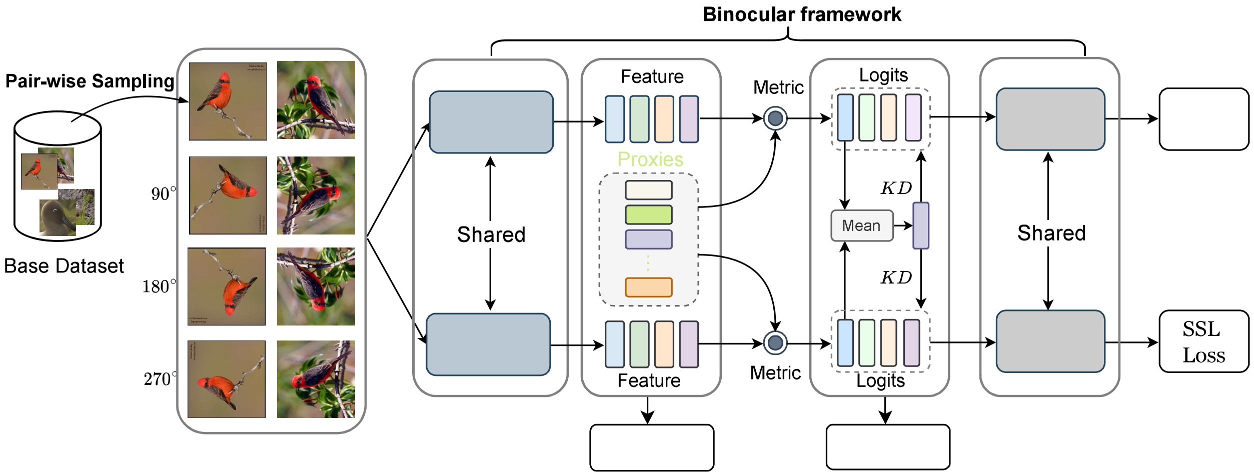


Fig. 2: Illustration of the proposed BFAU framework. BFAU takes a pair of samples from the same class and their augmented views as input to a feature encoder to obtain their feature representations. Then, four different objectives are performed on the feature representations to regularize the model in extracting generalized features for novel classes, including a cross-entropy loss, a feature alignment loss in the feature embedding space, a mutual knowledge distillation loss, and a self-supervised loss.

where d denotes the similarity metric, $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_i, ..., \mathbf{a}_M]$ denotes the learnable proxy matrix, where $\mathbf{a}_i$ is the prototype for class i. Note that when the similarity metric is the inner product, this strategy is the same as the conventional classification pipeline. The proxy-based training strategy is widely applied for the metric learning area [48], [49] and is also exploited in [12], [50] for FSL. For sample x, its classification loss is defined as,

$$L_{ce}(x) = H\left(y, P(x)\right), \tag{3}$$

where H denotes the cross-entropy loss, $P(x)$ is the predictive probability of visual sample x and y denotes its ground-truth label.

As indicated in [51], minimizing the cross-entropy loss is equivalent to maximizing the mutual information $I(\mathbf{Z}; Y)$ between the feature representations $\mathbf{Z}$ and labels Y. Maximizing $I(\mathbf{Z}; Y)$ is beneficial for the recognition of the base classes. However, such a training mechanism concentrates on the class-specific features of the base classes, which will cause *neural collapse* issue [52] that the representations of intra-class samples collapse to their class prototype. Thus, the model trained with the cross-entropy loss may neglect the general features widely shared among different classes, which will compromise the performance when the model is applied for the downstream tasks, especially for the novel classes that are disjoint from the training classes.

Following the InfoMax principle [17] that preserving the raw data information would benefit in learning more generalization representations among the classes, we relieve the objective loss to not only maximize the mutual information $I(\mathbf{Z}, Y)$ between the feature representations and their class labels but also maximize the mutual information $I(\mathbf{Z}, X)$ between the raw input samples and their feature representations. Maximizing $I(\mathbf{Z}, Y)$ encourages the model to learn feature representations associated with their labels while maximizing $I(\mathbf{Z}, X)$ encourages the model to preserve the raw information of the input samples with less bias to the class labels.

In practice, the maximization of $I(\mathbf{Z}, X)$ is intractable. Thus, we resort to an alternative objective $I(\mathbf{Z}; \mathbf{Z}')$, a lower bound of $I(\mathbf{Z}, X)$, where $\mathbf{Z}'$ denotes the feature embeddings of the other view of X. In this paper, we explore two kinds of views, an augmented view from the sample itself and a view from the associated class of the sample.

C. Mutual Information between Intra-class Samples

We first explore the mutual information between the intra-class samples. Specifically, we sample a pair of samples from the same class and maximize the mutual information objective $I(\mathbf{Z}_i, \mathbf{Z}_j) = I(f(X_i), f(X_j))$, X_i can be seen an intra-class view of X_j and vice versa. Given a pair of intra-class samples x_i and x_j, their class predictive probabilities $P(x_i)$ and $P(x_j)$ could be obtained with Eq. (1), and their mutual information objectives are:

$$I(x_i, x_j) \propto I(P(x_i), P(x_j)), \quad I(x_j, x_i) \propto I(P(x_j), P(x_i)). \tag{4}$$

In practice, maximizing Eq. (4) is hard to optimize. Thus, we introduce an alternative objective:

$$I(x_i, x_j) \propto I(P(x_i), P(x_{ij})) + I(P(x_j), P(x_{ij})), \tag{5}$$

where $P(x_{ij})$ denotes the mean value of $P(x_i)$ and $P(x_j)$. We add the mutual information between the intra-class samples to the objective to train the model. The objective for sample x_i is formulated as:

$$L(x_i) = \alpha H(y, P(x_i)) + \beta H(P(x_{ij}), P(x_i)), \tag{6}$$

where α and β are two hyper-parameters to balance the two items and $\alpha + \beta = 1$. Minimizing the cross-entropy loss $H(y, P(x_i))$ is equivalent to maximizing the mutual information $I(P(x_i); y)$ and minimizing $H(P(x_{ij}), P(x_i))$ is equivalent to maximizing the mutual information $I(x_i, x_j)$.

Interestingly, minimizing $H(P(x_{ij}), P(x_i))$ is also equivalent to distill $P(x_i)$ with $P(x_{ij})$. Thus, the objective loss for sample x_i is:

$$L(x_i) = H(\alpha y + \beta P(x_{ij}), P(x_i)). \tag{7}$$

Eq. (7) builds a balance between the mutual information $I(\mathbf{Z}, Y)$ and $I(\mathbf{Z}_i, \mathbf{Z}_j)$. This formulation is very similar to the knowledge distillation objective function where $P(x_{ij})$ is replaced with the prediction of a teacher model. In contrast, our strategy distills the samples without a teacher model but with their intra-class partners, which is easy to implement.

In the implementation, we develop a binocular framework for the above formulation to perform the mutual knowledge distillation for each input pair. As shown in Fig. 2, a pair of samples from the same class are sent to the framework that consists of twin networks and they are mutually supervised with each other. Therefore, for each input sample, it is supervised with both its ground-truth label and a soft label provided by the mean value of itself and its peer's predicted probability. Note that the mean value is detached and no gradient is propagated through it during the training. Since $P(x_{ij})$ denotes the relationships between the sample and the base classes, minimizing Eq. (7) could maximize the mutual information between the input pairs on the relationships with the base classes.

As discussed above, Eq. (7) is similar to the knowledge distillation methods. Different from the existing knowledge distillation approaches, the input samples of our method are not supervised by themselves from either a pre-trained teacher model or the previous iterations but are supervised by their intra-class samples. The only requirement of the proposed method is to sample in class-wise pairs. Such a sampling strategy is easy to implement and may bring in a basket of benefits. First, the knowledge may be distilled in a single network without a pre-trained teacher model, which relieves the computation burden. Second, the distilled knowledge from the teacher is not static but dynamically evolved as the sampling and the training proceed. Third, the mean predicted probability of the input pair serving as the teacher forces the model to produce more consistent and generalized feature representations. Finally, it is empirically observed that our method performs better than the two-stage KD competitors for FSL tasks.

D. Mutual Information between Different Sample Views

We then explore the mutual information between the sample and the augmented views from the sample itself. Due to its simplicity and effectiveness, the image rotation is used to augment the samples with three possible 2D rotations in $\mathcal{R} = \{90^\circ, 180^\circ, 270^\circ\}$. Note that the sample itself can be seen as being rotated with 0° and thus the combination rotation set is denoted as $\hat{\mathcal{R}} = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. For an input sample x_i, we first create its three rotated copies $\{x_i^r | r \in \mathcal{R}\}$, where x_i^r denotes the sample x_i rotated by r degrees, and then extract their visual embeddings with the feature extractor g and obtain their class prediction with the classification head h_{cls}. The mutual information between the sample and its augmented views could be estimated in both feature embedding space and class prediction logits space.

In the feature embedding space, maximizing the mutual information $I(x_i, x_j)$ is equivalent to minimizing the distances of their feature embeddings, which is formulated with:

$$L_{fa}(x_i, x_j) = \sum_{\forall r \in \hat{\mathcal{R}}} \|g(x_i^r) - g(x_j^r)\|_2. \tag{8}$$

where x_i and x_j are from the same class. Minimizing Eq. (8) encourages the feature alignment of the input pairs in different rotation views. In this way, the manifold information of different rotations in the feature embedding space could be preserved.

In the logits space, all the samples and their augmented views are projected into the label space, of which the dimensionality is equal to the class number. Thus, the logits of each sample contain the structure information of different classes. As the classification loss only performs on the original samples, their augmented views are harder to classify. To this end, we distill the knowledge from the mean probability value of the original input sample pair to the augmented ones. Accordingly, maximizing the mutual information $I(x_i, x_j)$ in the logits space is estimated with:

$$L_{kd}(x_i, x_j) = \sum_{\forall r \in \hat{\mathcal{R}}} H(P(x_{ij}), P(x_i^r)) + H(P(x_{ij}), P(x_j^r)). \tag{9}$$

where $P(x_{ij})$ is the mean value of the original input pair predicted probabilities $P(x_i)$ and $P(x_j)$. As Sec. A pointed out, the knowledge distillation encourages the model to focus on the hard samples. In this way, the model pays more attention to the hard samples and the rotated samples to extract more general representation patterns, which is beneficial to the downstream FSL tasks.

E. Balancing Feature Alignment and Uniformity Framework

We formulate the whole idea into a united framework. As illustrated in Fig. 2, the framework consists of a feature encoder and a classification head. Since we augment the samples with different rotations, a self-supervised head to predict the rotations of the input samples is also added after the class-predicted logits.

Based on the rotation predictions from both the input samples and their rotated views, the self-supervised head performs as a rotation classifier to predict their rotation labels. Thus, the self-supervised loss is formulated as:

$$L_{ssl}(x_i, x_j) = \sum_{\forall r \in \hat{\mathcal{R}}} H(r, Q(x_i^r)) + H(r, Q(x_j^r)), \tag{10}$$

where $Q(x)$ is the rotation predicted probability of x. Minimizing Eq. (10) is equivalent to maximizing the mutual information between the feature representations and the rotation labels, which learns less bias toward the class labels.

To this end, the overall objective of the framework is:

$$\begin{aligned} L_{all} = \mathbb{E}_{(x_i, x_j) \sim \mathcal{C}_{base}} & \overbrace{L_{ce}(x_i, x_j) \ + \alpha L_{fa}(x_i, x_j)}^{Alignment} \\ & \underbrace{\beta L_{kd}(x_i, x_j) + \gamma L_{ssl}(x_i, x_j)}_{Uniformity}, \end{aligned} \tag{11}$$

where $L_{ce}(x_i, x_j) = L_{ce}(x_i) + L_{ce}(x_j)$ denotes the cross-entropy loss of both x_i and x_j, α, β, and γ are hyper-parameters. The *alignment* denotes that the difference between the representations of different samples from the same classes should be minimized while the *uniformity* denotes that the difference between the representations of different views of the same samples should be maximized. To this end, our method strikes a balance between alignment and uniformity for feature representation learning and is abbreviated as **BFAU**. Minimizing the first two terms of Eq. (11) is beneficial for recognition of the base classes while minimizing the last two terms would reduce the bias toward the base classes, which encourages the model to learn comprehensive information to accommodate the novel classes. Thus, the objective function builds a balance between discrimination and generalization.

F. Applying the Model for FSL

Once the model is trained on the base classes, we remove both the classification head and self-supervised head and obtain the feature encoder g, which maps the input instances into the feature embedding space where the similarities are obtained. Given an N-way K-shot classification task with the support set S, we follow [5] and calculate the visual prototype of each class. Specifically, for class c, its prototype $\mathbf{p}_c$ is:

$$\mathbf{p}_c = \frac{1}{|S_c|} \sum_{x \in S_c} g(x), \tag{12}$$

where S_c and $|S_c|$ denote the data set and sample number for class c, respectively.

For each test sample x_t in the query set, its probability belonging to the class c is:

$$p(y = c | x_t) = \frac{\exp(d(g(x_t), \mathbf{p}_c))}{\sum_{n=1}^{N} \exp(d(g(x_t), \mathbf{p}_n))}, \tag{13}$$

where d is a similarity metric and the cosine similarity is applied in this work. To this end, the test samples can be predicted based on their probabilities belonging to all the candidate N classes.

IV. EXPERIMENTS

In this section, we first document the experimental settings and implementation details and then comprehensively compare the proposed approach with some competitors. Finally, both the ablation study and analysis are provided.

A. Datasets and Settings

Datasets. We evaluate our models on four datasets, miniImageNet [22], tieredImageNet [7], CIFAR-FS [23], and Caltech-UCSD Birds-200-2011 (CUB) [53]. Both miniImageNet and tieredImageNet datasets are the subsets of the ILSVRC-12 ImageNet dataset. Specifically, the miniImageNet dataset contains 100 classes and 600 downsampled images of size 84×84 per class. Following the split introduced in [26], 64 classes are used for training; the remaining 16 and 20 classes are used for model validation and testing, respectively. In contrast to the miniImageNet, the tieredImageNet dataset has a hierarchical structure of broader categories corresponding to high-level nodes. The top hierarchy has 34 categories, which are divided into 20 training, 6 validation, and 8 test categories, respectively, corresponding to 351 base, 97 validation, and 160 test classes. This high-level split strategy ensures that the training classes are semantically distinct from the test classes. The average number of samples in each class is 1,281. Similarly, all images are resized to 84×84 pixels. The CIFAR-FS dataset is the derivative of the CIFAR-100 dataset that contains 100 object classes, each of which has 600 samples of 32×32 pixels. Specifically, the CIFAR-FS dataset randomly splits the original 100 CIFAR classes into 64, 16, and 20 classes for training, validation, and testing, respectively. CUB is a fine-grained dataset that consists of 200 bird classes. We follow [12] and split the 200 classes into 100, 50, and 50 for training, validation, and testing, respectively.

In the experiments, we train a model with the training set and select the model that performs the best on the evaluation set for the test. We evaluate our models with five runs and report their average value as the final performance. For each trial, the instance number of each query class is set to 15 and the classification performance is averaged over 600 randomly sampled FSL tasks from the test set. For the hyper-parameters, we select them from 0 to 1 with an interval of 0.2 and report the performances of the model that performs the best on the validation set.

Evaluation metric. For FSL, we follow the existing FSL competitors and take both the classification accuracy and the 95% confidence interval as the evaluation metric. For the ablation study, only the classification accuracy is reported.

Implementation details. We use the ResNet12 [73] backbone as the feature extractor for a fair comparison with the existing approaches. We also conduct experiments with the ResNet18 [73] backbone on miniImageNet, CIFAR-FS, and CUB datasets. *If not specified, the results are obtained with the ResNet12 backbone.* For miniImageNet and CIFAR-FS datasets, we train 60 epochs with batch sizes of 32 and 56, respectively. The model is trained with the SGD optimizer with a momentum of 0.9; the learning rate is initialized as 0.05 and decayed with a factor of 0.1 at epochs 40 and 50, respectively. For both tieredImageNet and CUB datasets, we train 100 epochs with the SGD optimizer. The learning rate is also initialized as 0.05 and decayed at epochs 60 and 80 with a factor of 0.1, respectively. The batch size is set to 32. In the training stage, the standard data augmentation strategies are applied, including random resized crop and horizontal flip. The model is trained with PyTorch on the platform with two 1080 GPUs.

Competitors. To show the effectiveness of our model, we only select the competitors published in the recent three years for comparison. These competitors are in the same setting as ours and their results are directly reported from the published literature.

B. Comparison with State-of-The-Art

Results on both miniImageNet and tieredImageNet. TAB. I shows the results of our BFAU and the competitors

TABLE I: FSL accuracy (%) and 95% confidence interval on both miniImageNet and tieredImageNet datasets. The best results are in **bold**. Underline denotes the second-best.

Backbone	Method	1-Shot	5-Shot
miniImageNet			
ResNet18	Su *et.al* [33]	-	76.0 ± n/a
	SimpleShot [54]	62.85 ± 0.20	80.02 ± 0.14
	AFHN [30]	62.38 ± 0.72	78.16 ± 0.56
	Arcmax [55]	59.88 ± 0.67	80.35 ± 0.73
	MixtFSL [56]	60.11 ± 0.73	77.76 ± 0.58
	BFAU (Ours)	**67.21 ± 0.38**	**82.86 ± 0.30**
SetFeat12	SetFeat [57]	68.32 ± 0.62	82.71 ± 0.46
PyramidFCN	MCL [58]	67.45 ± n/a	84.36 ± n/a
ResNet12	Shot-Free [59]	59.04 ± n/a	77.64 ± n/a
	MetaOptNet [60]	64.09 ± 0.62	80.00 ± 0.45
	MABAS [61]	65.08 ± 0.86	82.70 ± 0.54
	RFS-distill [11]	64.82 ± 0.60	82.14 ± 0.43
	FEAT [62]	66.78 ± 0.20	82.05 ± 0.14
	DSN-MR [63]	64.60 ± 0.72	79.51 ± 0.50
	DeepEMD [64]	65.91 ± 0.82	82.41 ± 0.56
	DeepDBC [65]	67.34 ± 0.43	82.38 ± 0.32
	FRN [66]	66.45 ± 0.19	82.83 ± 0.13
	DAN [67]	67.76 ± 0.46	82.71 ± 0.31
	MixtFSL [56]	63.98 ± 0.79	82.04 ± 0.49
	SKD-GEN1 [44]	67.04 ± 0.85	83.54 ± 0.54
	SnaTCHer-L [68]	67.60 ± 0.83	82.36 ± n/a
	RENet [69]	67.60 ± 0.44	82.58 ± 0.30
	BML [70]	67.04 ± 0.63	83.63 ± 0.29
	COSOC [71]	69.28 ± 0.49	**85.16 ± 0.42**
	Li *et.al* [72]	68.94 ± 0.28	85.07 ± 0.50
	BFAU (Ours)	**69.53 ± 0.32**	84.81 ± 0.31
tieredImageNet			
PyramidFCN	MCL [58]	72.01 ± n/a	86.31 ± n/a
SetFeat12	SetFeat [57]	73.63 ± 0.88	87.59 ± 0.57
ResNet12	MetaOptNet [60]	65.99 ± 0.72	81.56 ± 0.53
	MABAS [61]	65.08 ± 0.86	82.70 ± 0.54
	RFS-distill [11]	71.52 ± 0.69	86.03 ± 0.49
	FRN [66]	71.16 ± 0.22	86.01 ± 0.15
	DeepEMD [64]	71.16 ± 0.87	86.03 ± 0.58
	DeepDBC [65]	72.34 ± 0.49	87.31 ± 0.32
	DSN-MR [63]	67.39 ± 0.82	82.85 ± 0.56
	DAN [67]	71.89 ± 0.52	85.96 ± 0.35
	SnaTCHer-L [68]	70.85 ± n/a	85.23 ± n/a
	MixtFSL [56]	70.97 ± 1.03	86.16 ± 0.67
	SKD-GEN1 [44]	72.03 ± 0.91	86.50 ± 0.58
	RENet [69]	71.61 ± 0.51	85.28 ± 0.35
	COSOC [71]	73.57 ± 0.43	87.57 ± 0.10
	Li *et.al* [72]	**73.76 ± 0.32**	**87.83 ± 0.59**
	BML [70]	68.99 ± 0.50	85.49 ± 0.34
	BFAU (Ours)	72.89 ± 0.42	86.68 ± 0.38

TABLE II: FSL accuracy (%) and 95% confidence interval on both CIFAR-FS and CUB datasets. The best results are highlighted in bold. Underline denotes the second-best.

Backbone	Method	1-Shot	5-Shot
CIFAR-FS			
ResNet18	AFHN [30]	68.32 ± 0.93	81.45 ± 0.87
	BFAU (Ours)	**77.82 ± 0.43**	**89.04 ± 0.25**
ResNet12	Shot-Free [59]	69.20 ± 0.40	84.70 ± 0.40
	MetaOptNet [60]	72.80 ± 0.70	85.00 ± 0.50
	RFS-distill [11]	73.89 ± 0.80	86.93 ± 0.50
	DeepEMD [64]	75.65 ± 0.83	88.69 ± 0.50
	RENet [69]	74.51 ± 0.46	86.60 ± 0.32
	BML [70]	73.45 ± 0.47	88.04 ± 0.33
	SKD-GEN1 [44]	76.90 ± 0.90	88.90 ± 0.60
	BFAU (Ours)	**77.06 ± 0.41**	**89.86 ± 0.32**
CUB			
ResNet18	Baseline++ [12]	67.02 ± 0.90	83.58 ± 0.54
	AFHN [30]	70.53 ± 1.01	83.95 ± 0.63
	MixtFSL [56]	73.94 ± 1.10	86.01 ± 0.50
	Arcmax [55]	74.22 ± 1.09	88.65 ± 0.55
	BFAU (Ours)	**78.83 ± 0.34**	**89.72 ± 0.23**
SetFeat12	SetFeat [57]	79.60 ± 0.80	90.48 ± 0.44
ResNet12	Deep DTN [74]	72.00 ± n/a	85.10 ± n/a
	FEAT [62]	68.87 ± 0.22	82.90 ± 0.15
	RFS-distill [11]	77.12 ± 0.55	88.89 ± 0.31
	FRN [66]	83.16 ± n/a	**92.59 ± n/a**
	DeepEMD [64]	**83.35 ± n/a**	91.60 ± n/a
	RENet [69]	79.49 ± 0.44	91.11 ± 0.24
	BML [70]	76.21 ± 0.63	90.45 ± 0.36
	BFAU (Ours)	80.48 ± 0.40	92.08 ± 0.28

on both miniImageNet and tieredImageNet datasets. On one hand, Our BFAU performs very competitively on miniImageNet when compared with the models trained with the same backbone. For ResNet18, our model obtains 4.36% and 2.84% improvements over [54] for 5-way 1-shot and 5-way 5-shot classification tasks, respectively. Besides, the 5-way 5-shot classification accuracy of our model has 6.86% improvement over that of [33], a method also enjoying the benefits of self-supervision learning. For ResNet12, our model has a 0.25% improvement on the 5-way 1-shot classification task over the second-best method and is 0.35% inferior to the best method COSOC [71]. However, COSOC requires manually cropping each image in a base set according to the largest rectangular bounding box containing the foreground object, which is hard to implement. Moreover, on both 5-way 1-shot and 5-shot tasks, our model performs much better than [11] that distills knowledge from a pre-trained teacher model, which indicates the effectiveness of the proposed mutual distillation framework.

On the other hand, BFAU also performs very competitively on the tieredImageNet dataset. Specifically, our BFAU is only 0.87% and 1.15% inferior to [72] on 1-shot and 5-shot classification tasks, respectively. However, [72] and [57] exploit support-query relationships in a many-to-many correspondence way on the dense feature maps and require more operations, while our method exploits support-query relationships in a one-to-one correspondence way on the global image representation vector and is more efficient. Besides, we observe that the performances on the tieredImageNet dataset are higher than those on the miniImageNet, even though the tieredImageNet dataset is more challenging. The reason may be that the tieredImageNet consists of more instances with more base categories; hence the models could learn a good feature representation for the downstream FSL tasks.

Results on CIFAR-FS and CUB. In TAB. II, we demonstrate the comparison results on both CIFAR-FS and CUB datasets. For the CIFAR-FS dataset, we observe that our model performs the best for both 5-way 1-shot and 5-shot classification tasks with different backbones. When compared with the models trained using ResNet12, our model has 0.16% and 0.96% improvements over SKD-GEN1 [44] that performs the second-best on the 5-way 1-shot and 5-shot classification tasks, respectively. Besides, we observe our method performs

TABLE III: FSL accuracy (%) of the model with different losses on miniImageNet, CIFAR-FS, and CUB datasets.

Objective	miniImageNet		CIFAR-FS		CUB	
	1-shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
CE	62.02	79.64	71.50	86.00	73.12	87.49
Eq. (3)	64.96	80.21	72.93	86.72	75.34	88.16
Eq. (7)	66.72	82.30	75.50	87.92	76.05	88.38
Eq. (11)	69.48	84.72	76.98	89.42	80.31	90.06

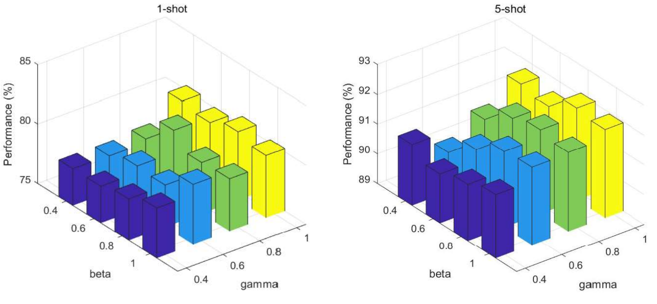
much better than BML [70], a competitor also applies a mutual framework. Differently, BML takes a single sample as input to two separate networks, one is for whole feature representation, and the other is for local feature maps, while our method takes a pair of samples as input to a single network to mutually distill each other. When compared with the models using ResNet18, our model has 9.50% and 7.59% improvements over AFHN [30] for 5-way 1-shot and 5-shot classification tasks, respectively. We speculate that AFHN may suffer from the overfitting issue with the ResNet18 which has a larger capacity than ResNet12. In contrast, our method may mitigate this issue by softening the labels and augmenting the training data. For the CUB dataset, we observe that the proposed BFAU performs very competitively on both tasks with both backbones. When comparing the results with ResNet12, BFAU performs slightly worse than [64], [66]. We argue that both competitors exploit the support-query correspondences on either the feature map or the image grids and learn finer feature matching, thus is beneficial for fine-grained datasets. In contrast, our method performs the support-query relationships in a simple prototype-to-vector way. When comparing the results with ResNet18, BFAU obtains the best on the two tasks and achieves 4.61% improvement over the second-best method on the 5-way 1-shot task, which indicates that the proposed BFAU is very competitive on the fine-grained dataset.

C. *Further Analysis*

Impacts of Objective Functions. In this section, we conduct experiments to evaluate the impacts of objective functions. TAB. III shows the results of different objective functions on three datasets. 'CE' denotes the cross-entropy loss. 'Eq. (3)' denotes the proxy-based cross-entropy loss. 'Eq. (7)' denotes the combination of both cross-entropy loss and knowledge distillation loss. 'Eq. (11)' further combines the self-supervised loss and feature alignment loss on the basis of Eq. (7). Note that the results in TAB. III are obtained when all the hyper-parameters equal 1. From the results, we have the following observations. First, the model trained with proxy-based cross-entropy loss boosts the performance on three datasets, especially on the 1-shot task. Second, the mutual knowledge distillation could improve the performances significantly over the baseline, indicating the effectiveness of the distillation approach. Third, exploiting the augmented samples further boosts the FSL performances, indicating that the data augmentation brings benefits in generalizing the downstream tasks. Besides, it is interesting to observe that the performance gains on the 5-way 1-shot classification task are larger than those on the 5-way 5-shot classification task. We speculate that the reason is that the classifiers are more robust to the outliers when more support data are provided.

TABLE IV: FSL accuracy (%) of the combinations of different loss items on miniImageNet, CIFAR-FS, and CUB datasets.

L_{fa}	L_{kd}	L_{ssl}	miniImageNet		CIFAR-FS		CUB	
			1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
✗	✗	✗	64.96	80.21	72.93	86.72	75.34	88.16
✓	✗	✗	65.02	80.68	72.28	86.22	75.88	87.93
✗	✓	✗	66.56	82.46	74.55	88.33	76.55	90.66
✗	✗	✓	62.93	80.19	70.46	86.88	73.22	88.04
✓	✓	✗	66.87	83.12	76.09	88.55	78.69	90.96
✓	✗	✓	65.12	80.89	73.69	87.81	77.31	89.26
✗	✓	✓	68.94	84.01	75.60	88.53	79.16	91.28
✓	✓	✓	69.48	84.72	76.98	89.42	80.31	90.06



(a) 1-Shot (b) 5-Shot

Fig. 3: Few-shot results of different hyper-parameters on the CUB dataset.

To further validate the effects of the losses in Eq. (11), we conduct ablation studies to evaluate the impacts of different loss terms on miniImageNet, CIFAR-FS, and CUB datasets. From the results reported in TAB. IV, we observe that the feature alignment loss L_{fa} improves a little or even hurts the performance while adding L_{ssl} individually into the objective would hurt the performance. However, their combination would improve the performance of the three datasets. In contrast, adding L_{kd} individually would boost the performance with large margins. Note that the third line in TAB. IV is different from the third line in TAB. III as the former also consists of the knowledge distillation on the augmented views. Though L_{fa} and L_{ssl} bring little performance improvement, combining them with L_{kd} would significantly boost the performance.

Impacts of Hyper-parameters. In this experiment, we evaluate the impacts of hyper-parameters on the few-shot classification performances. We observe that the few-shot performances are not sensitive to α and set it to 1, thus we vary both β and γ from 0.4 to 1 with an interval of 0.2 to evaluate their impacts. As illustrated in Fig. 3, we observe that the performances differ significantly with different β and γ, and the superior performances are obtained when γ equals 1.

Evaluation of Different Ways of Producing Soft Labels. TAB. V illustrates the few-shot classification results of Label Smoothing [20], TF-KD [21], RFS-distill [11], [45], and our method trained with Eq. (7). These methods differ in the way

TABLE V: Comparison results (%) of Label Smoothing, TF-KD [21], RFS-distill [11], [45] and ours on three datasets. * denotes the results implemented by ourselves with the released codes using the cosine similarity metric, same with ours.

Method	miniImageNet		CIFAR-FS		CUB	
	1-shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
LS [20]	63.38	75.90	71.17	83.30	72.21	84.47
Tf-KD [21]	64.21	77.95	72.43	84.78	73.46	85.93
RFS-distill* [11]	66.03	80.86	74.37	86.18	77.12	88.89
Li *et.al* [45]	65.39	81.51	74.64	87.63	-	-
Ours	66.72	82.30	75.50	87.92	76.05	88.38

TABLE VI: Comparison results (%) of different self-supervised augmentation ways.

Method	miniImageNet		CIFAR-FS		CUB	
	1-shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
RandomCrop	66.72	82.98	75.55	88.79	78.12	90.85
Jigsaw	69.31	84.83	76.12	89.69	80.69	91.73
Rotation	69.48	84.72	76.98	89.42	80.31	90.06

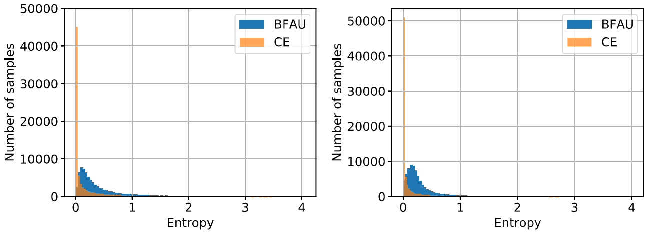
of producing soft labels for optimizing the model. LS, TF-KD, and ours are teacher-free methods while both RFS-distill and [45] require pre-training a teacher model in advance. LS and TF-KD artificially design the soft labels while RFS-distill, [45], and ours obtain the soft labels from the prediction of the samples. From the results, we observe that the distillation approaches perform much better than LS and TF-KD. We speculate that the soft labels provided by the distillation-based approaches contain structural information, which not only prevents the model from overfitting but also improves the model's generalization. We also observe that our one-stage method performs better than the two-stage knowledge distillation competitors on both miniImageNet and CIFAR-FS datasets, which indicates that the mutually distilling knowledge between intra-class samples is beneficial for the model's generalization on the novel classes. Besides, we observe that Label Smoothing [20] performs even worse than the baseline without the soft labels, which indicates that the label smoothing strategy would impair the model's generalization.

In terms of efficiency, LS, TF-KD, and our method do not require a pre-trained teacher network and thus are more efficient. Theoretically, the efficiency of our method is twice higher as that of RFS-distill during training and is the same as RFS-distill during the test.

Impacts of different data augmentation methods. In this experiment, we explore the effects of different self-supervised augmentation ways. Except for the rotation, we explore the other two augmentation ways, *i.e.*, Jigsaw, and random crop. Specifically, the jigsaw task rearranges the input image and uses the index of the permutation as the self-supervised label while random crop augmentation does not provide self-supervised labels. TAB. VI shows the results of three different augmentation ways. From the results, we observe that random crop augmentation performs the worst on three datasets. We speculate that no self-supervision loss is associated with the objective. Besides, the Jigsaw performs slightly better than Rotation on the CUB dataset. In contrast to Rotation, the Jigsaw has more self-supervised views and captures finer patterns for the fine-grained dataset. However, for the coarse-grained datasets, Rotation and Jigsaw perform neck-to-neck.

TABLE VII: Few-Shot classification accuracy (%) with and without pre-training stage on the three datasets.

Pretrain	miniImageNet		tieredImageNet		CUB	
	1-Shot	5-Shot	1-shot	5-Shot	1-shot	5-Shot
w/o	69.79	84.81	72.89	86.68	80.48	92.08
with	71.12	85.03	73.93	87.03	83.22	92.46



(a) miniImageNet (b) CIFAR-FS

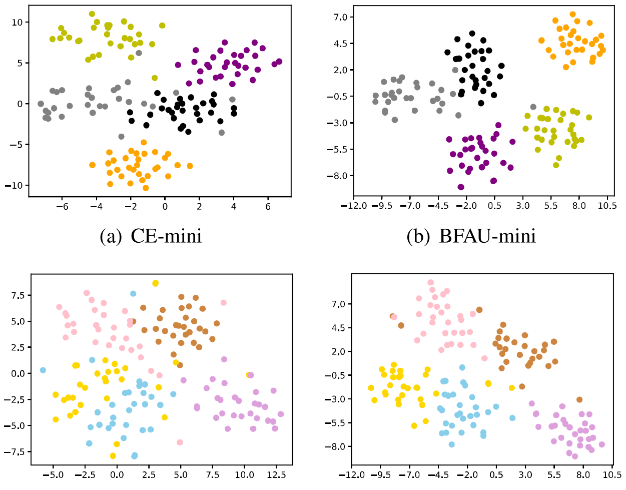
Fig. 4: Histogram of entropy values of the predicted probabilities on both miniImageNet and CIFAR-FS datasets. The networks are trained with ResNet12.

Pre-training for Few-shot Learning. As most of the competitors follow a two-stage pipeline, a pretraining, and a fine-tuning stage, we also apply a two-stage pipeline to train the feature extractor. Instead of training a supervised model in the first stage, we train the model in a self-supervised way. Specifically, we apply MOCO [76] to train the model in the first stage and then use the trained parameters to initialize the feature extractor. TAB. VII shows the results with and without the pre-training stage on the three datasets. From the results, we conclude that the pre-training stage could boost performance significantly, especially on the 1-shot task.

D. Qualitative Analysis

Fig. 4 illustrates the histogram of entropy values of the predicted probabilities on two benchmarks. We observe that the entropy values of the baseline with CE are smaller and more concentrated than those of our BFAU, which indicates that the baseline could confidently predict the instances from the base set into the correct classes but easily suffers from the overfitting issue. In contrast, our BFAU predicts the base instances less confidently and smooths the predictions, alleviating the overfitting issue correspondingly and preventing the feature collapse.

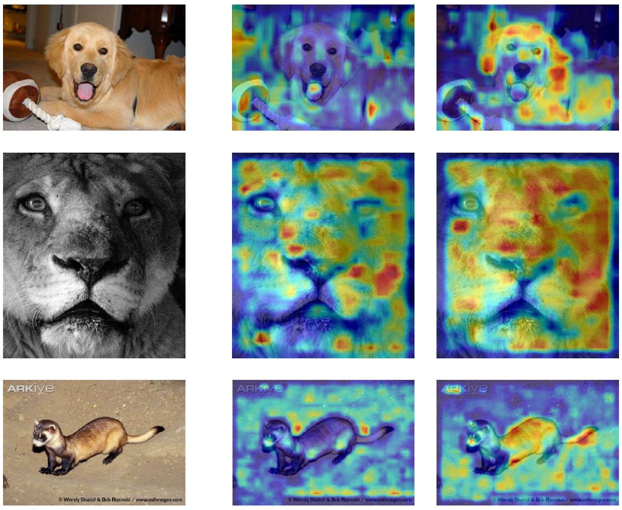
Fig. 5 illustrates the tSNE [75] visualization of the feature embeddings of both miniImageNet and CIFAR-FS datasets, respectively. We randomly select a query set to visualize, which consists of 150 samples from 5 novel classes. The baseline model trained with CE loss is taken for comparison. From the results, we observe that the feature embeddings of our BFAU are distributed more compactly and have fewer outliers than the baseline. Fig. 6 provides the Grad-CAM [77] visualization results on three novel classes from the



(a) CE-mini (b) BFAU-mini

(c) CE-cifar (d) BFAU-cifar

Fig. 5: tSNE [75] visualization of both miniImageNet and CIFAR-FS datasets. Different colors denote different classes.



(a) Input Image (b) Baseline (c) BFAU

Fig. 6: Grad-CAM [77] visualization results on the novel classes of the miniImageNet dataset.

miniImageNet dataset. We observe that our BFAU focuses more discriminative areas than the baseline, indicating that BFAU extracts more generalization features for the novel classes.

Fig. 7 illustrates the Top-3 predictions of some examples with the BFAU model at the twentieth epoch. For the easy instances (*e.g.*, the carrier), they are predicted very confidently and consistently between the input pairs, where BFAU brings in little benefits. For the hard instances (*e.g.*, the beetle), they are neither predicted confidently nor correctly, thus BFAU neither brings in benefits. For the cases where the input pair consists of an easy instance and a hard one, (*e.g.*, white fox), BFAU will force the easy one to help the hard one, and in turn, the hard one will soften the easy one. As the training progresses, each sample goes from hard to easy, thus BFAU would work at a certain stage for each sample.

V. Conclusion

In this paper, we have proposed an effective FSL approach to learning generalized feature representations by balancing the alignment and uniformity in the feature embedding space and developed a conceptually simple but methodologically effective framework. Our method delivers generalized feature representations for the disjoint target classes via smoothing the class predictions of the input class-wise pairs and augmenting more hard instances. The extensive experiments on four benchmark datasets demonstrate that the proposed approach effectively extracts compact intra-class feature representations for the novel classes and achieves competitive FSL results. Besides, we conclude that maximizing the mutual information between the feature representations and the input samples could improve the generalization of the model on the novel classes.

Appendix A
Theoretical Analysis

Inspired by [78], we show that our method pays different attention to different instances. When the hyperparameter β is fixed, the gradient in Eq. (7) with respect to logit value z_i is given as:

$$\frac{\partial\mathcal{L}_{tfd}(x)}{\partial z_i} = \partial_i^{tfd} = \alpha(p_i - y_i) + \beta(p_i - \frac{(p_i + p_j)}{2}). \tag{14}$$

For z_i where i equals to the target class G, *i.e.*, $y_i = 1$, ∂_i^{tfd} becomes:

$$\partial_i^{tfd} = (p_{i,G} - 1) - \beta(\frac{(p_{i,G} + p_{j,G})}{2} - 1). \tag{15}$$

For z_i where i does not equal to the target class, *i.e.*, $y_i = 0$, ∂_i^{tfd} becomes:

$$\partial_i^{tfd} = p_i - \beta\frac{(p_i + p_j)}{2}. \tag{16}$$

Considering that $0 < \beta < 1$, we set $\partial_i^{tfd} > 0$ for all i except the target class, *i.e.*, $p_i - \beta(p_i + p_j)/2 > 0$. As $\sum_i p_i = 1$, we conclude $(p_{i,G} - 1) - \beta((p_{i,G} + p_{j,G})/2 - 1) < 0$ and $\sum_{i\neq G} |p_i - \beta(p_i + p_j)/2| = (1 - p_{i,G}) - \beta(1 - (p_{i,G} + p_{j,G})/2)$. To this end, the L_1 norm of the gradient of $\mathcal{L}_{tfd}(x)$ is written as:

$$\sum_i |\partial_i^{tfd}| = 2(1 - p_{i,G}) - 2\beta(1 - \frac{p_{i,G} + p_{j,G}}{2}). \tag{17}$$

Similarly, we may obtain the L_1 norm of the gradient of $L_{CE}(x)$ in Eq. (3):

$$\sum_i |\partial_i^{CE}| = 2(1 - p_{i,G}). \tag{18}$$

Then we consider the gradient rescaling factor by applying TFD:

$$\begin{aligned}\frac{\sum_i |\partial_i^{tfd}|}{\sum_i |\partial_i^{CE}|} &= 1 - \beta\frac{1 - (p_{i,G} + p_{j,G})/2}{1 - p_{i,G}} \\ &= 1 - \beta\frac{(\gamma_i + \gamma_j)/2}{\gamma_i},\end{aligned} \tag{19}$$

where r_i denotes the probability of being incorrect.

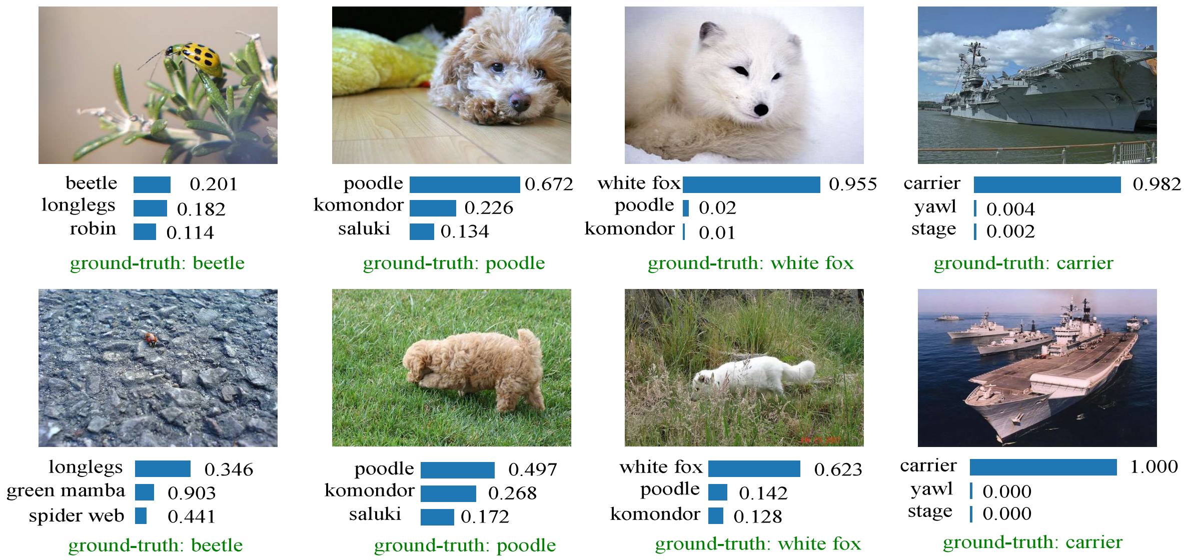


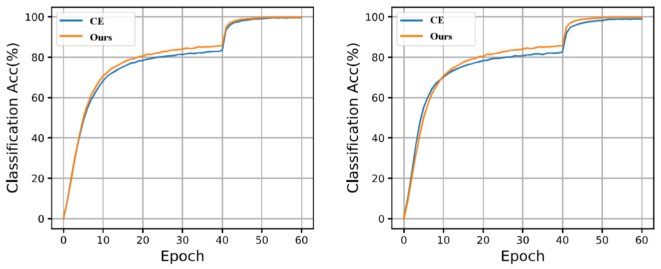
Fig. 7: Examples of Top-3 predictions with the BFAU model at the 20 epochs.

When the sample x_i is hard to classify, *i.e.*, γ_i is large and $\frac{(\gamma_i+\gamma_j)/2}{\gamma_i}$ is small, we obtain a large gradient rescaling factor, which means that the model would pay more attention to the hard sample. Conversely, if the sample is easy to classify, its gradient rescaling factor is small and the model would pay less attention to it. From this point of view, the proposed method is a dynamic attention strategy for the instances. As shown in Fig. 8, the proposed method has an advantage over the models with CE in convergence, which indirectly proves that our method is equipped with the capability to mine hard samples.

For a pair of input samples from the same class, their ratio of the gradient re-scaling factors is:

$$\begin{aligned}\eta_{ij} &= \frac{\sum_i |\partial_i^{tfd}|}{\sum_i |\partial_i^{CE}|} / \frac{\sum_j |\partial_j^{tfd}|}{\sum_j |\partial_j^{CE}|} \\ &= (1-\beta\frac{(\gamma_i+\gamma_j)/2}{\gamma_i})/(1-\beta\frac{(\gamma_i+\gamma_j)/2}{\gamma_j}) \\ &= \frac{(2-\beta)\gamma_i\gamma_j-\beta\gamma_j^2}{(2-\beta)\gamma_i\gamma_j-\beta\gamma_i^2}.\end{aligned} \tag{20}$$

If sample x_i is harder than sample x_j, *i.e.*, $\eta_{ij} > 1$, the harder sample may obtain more attention.



(a) miniImageNet (b) CIFAR-FS

Fig. 8: The classification performances of both miniImageNet and CIFAR-FS datasets on the base data.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 62002320, the Key R&D Program of Zhejiang Province, China (2021C01119), NSFC (U19B2043, 61672456), the Fundamental Research Funds for the Central Universities (2020QNA5010), and 2018203B036.

References

[1] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *ICML*, pp. 1126–1135, 2017. 1, 2
[2] M. Qi, J. Qin, X. Zhen, D. Huang, Y. Yang, and J. Luo, “Few-shot ensemble learning for video classification with slowfast memory networks,” in *ACM MM*, pp. 3007–3015, 2020. 1
[3] Y. Guo, R. Du, X. Li, J. Xie, Z. Ma, and Y. Dong, “Learning calibrated class centers for few-shot classification by pair-wise similarity,” *TIP*, vol. 31, pp. 4543–4555, 2022. 1
[4] B. Xi, J. Li, Y. Li, R. Song, D. Hong, and J. Chanussot, “Few-shot learning with class-covariance metric for hyperspectral image classification,” *TIP*, vol. 31, pp. 5079–5092, 2022. 1
[5] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *NeurIPS*, pp. 4077–4087, 2017. 1, 3, 6
[6] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *CVPR*, pp. 1199–1208, 2018. 1, 3
[7] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, “Meta-learning for semi-supervised few-shot classification,” in *ICLR*, 2018. 1, 6
[8] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, “Low-shot learning from imaginary data,” in *CVPR*, pp. 7278–7286, 2018. 1, 2
[9] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, “Negative margin matters: Understanding margin in few-shot classification,” in *ECCV*, pp. 438–455, 2020. 1, 3
[10] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, “Meta-baseline: Exploring simple meta-learning for few-shot learning,” in *ICCV*, pp. 9062–9071, 2021. 1, 3
[11] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, “Rethinking few-shot image classification: a good embedding is all you need?,” in *ECCV*, 2020. 1, 3, 7, 8, 9
[12] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” in *ICLR*, 2019. 1, 3, 4, 6, 7
[13] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, “A baseline for few-shot image classification,” in *ICLR*, 2020. 1

[14] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020. 1

[15] P. Izmailov, P. Kirichenko, N. Gruver, and A. G. Wilson, "On feature learning in the presence of spurious correlations," in *NeurIPS*, 2022. 1

[16] C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: spatially-aware few-shot transfer," in *NeurIPS*, pp. 21981–21993, 2020. 1

[17] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, 1988. 1, 4

[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NeurIPS Deep Learning and Representation Learning workshop*, 2015. 1, 3

[19] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *CVPR*, pp. 9163–9171, 2019. 1

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, pp. 2818–2826, 2016. 1, 8, 9

[21] L. Yuan, F. E. H. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *CVPR*, pp. 3903–3911, 2020. 1, 8, 9

[22] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one-shot learning," in *NeurIPS*, pp. 3630–3638, 2016. 2, 3, 6

[23] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *ICLR*, 2019. 2, 6

[24] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *ICLR*, 2018. 2

[25] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," in *ICLR*, 2019. 2

[26] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2016. 2, 6

[27] T. Munkhdalai and H. Yu, "Meta networks," *Proceedings of machine learning research*, vol. 70, p. 2554–2563, 2017. 2

[28] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *CVPR*, pp. 3018–3027, 2017. 2

[29] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes, and A. Bronstein, "Delta-encoder: an effective sample synthesis method for few-shot object recognition," in *NeurIPS*, pp. 2845–2855, 2018. 2

[30] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *CVPR*, pp. 13470–13479, 2020. 2, 7, 8

[31] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert, "Image deformation meta-networks for one-shot learning," in *CVPR*, pp. 8680–8689, 2019. 2

[32] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *ICCV*, pp. 8059–8068, 2019. 2, 3

[33] J.-C. Su, S. Maji, and B. Hariharan, "When does self-supervision improve few-shot learning?," in *ECCV*, pp. 645–666, 2020. 2, 3, 7

[34] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *CVPR*, pp. 403–412, 2019. 3

[35] N. Passalis and A. Tefas, "Unsupervised knowledge transfer using similarity embeddings," *TNNLS*, vol. 30, no. 3, pp. 946–950, 2018. 3

[36] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, pp. 4133–4141, 2017. 3

[37] E. J. Crowley, G. Gray, and A. J. Storkey, "Moonshine: Distilling with cheap convolutions," in *NeurIPS*, pp. 2888–2898, 2018. 3

[38] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *CVPR*, pp. 578–587, 2019. 3

[39] X. Zhu, S. Gong, *et al.*, "Knowledge distillation by on-the-fly native ensemble," in *NeurIPS*, pp. 7517–7527, 2018. 3

[40] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *AAAI*, pp. 5191–5198, 2020. 3

[41] C. Yang, L. Xie, S. Qiao, and A. L. Yuille, "Training deep neural networks in generations: A more tolerant teacher educates better students," in *AAAI*, pp. 5628–5635, 2019. 3

[42] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, pp. 4320–4328, 2018. 3

[43] S. Yun, J. Park, K. Lee, and J. Shin, "Regularizing class-wise predictions via self-knowledge distillation," in *CVPR*, pp. 13876–13885, 2020. 3

[44] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, "Self-supervised knowledge distillation for few-shot learning," *arXiv preprint arXiv:2006.09785*, 2020. 3, 7

[45] L. Liang, J. Weidong, and J. R. Yingkun HUANG, "Enhancing the generalization performance of few-shot image classification with self-knowledge distillation," *Studies in Informatics and Control*, vol. 31, no. 2, pp. 71–80, 2022. 3, 8, 9

[46] Z. Chen, J. Ge, H. Zhan, S. Huang, and D. Wang, "Pareto self-supervised training for few-shot learning," in *CVPR*, pp. 13663–13672, 2021. 3

[47] Y. An, H. Xue, X. Zhao, and L. Zhang, "Conditional self-supervised learning for few-shot classification.," in *IJCAI*, pp. 2140–2146, 2021. 3

[48] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *ICCV*, pp. 360–368, 2017. 4

[49] Y. Yu, D. Zhang, Y. Li, and Z. Zhang, "Multi-proxy learning from an entropy optimization perspective," in *IJCAI*, 2022. 4

[50] Y. Yu, D. Zhang, Y. Li, and Z. Zhang, "Multi-proxy learning from an entropy optimization perspective," in *IJCAI*, pp. 1594–1600, 2022. 4

[51] M. Boudiaf, J. Rony, I. M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed, "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in *ECCV*, pp. 548–564, 2020. 4

[52] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *PNAS*, vol. 117, no. 40, pp. 24652–24663, 2020. 4

[53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011. 6

[54] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning," *arXiv preprint arXiv:1911.04623*, 2019. 7

[55] A. Afrasiyabi, J.-F. Lalonde, and C. Gagné, "Associative alignment for few-shot image classification," in *ECCV*, pp. 18–35, 2020. 7

[56] A. Afrasiyabi, J.-F. Lalonde, and C. Gagne, "Mixture-based feature space learning for few-shot image classification," in *ICCV*, pp. 9041–9051, 2021. 7

[57] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, and C. Gagné, "Matching feature sets for few-shot image classification," in *CVPR*, pp. 9014–9024, 2022. 7

[58] Y. Liu, W. Zhang, C. Xiang, T. Zheng, D. Cai, and X. He, "Learning to affiliate: Mutual centralized learning for few-shot classification," in *CVPR*, pp. 14411–14420, 2022. 7

[59] A. Ravichandran, R. Bhotika, and S. Soatto, "Few-shot learning with embedded class models and shot-free meta training," in *ICCV*, pp. 331–339, 2019. 7

[60] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *CVPR*, pp. 10657–10665, 2019. 7

[61] J. Kim, H. Kim, and G. Kim, "Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning," in *ECCV*, pp. 599–617, 2020. 7

[62] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *CVPR*, pp. 8808–8817, 2020. 7

[63] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *CVPR*, pp. 4136–4145, 2020. 7

[64] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *CVPR*, pp. 12203–12213, 2020. 7, 8

[65] J. Xie, F. Long, J. Lv, Q. Wang, and P. Li, "Joint distribution matters: Deep brownian distance covariance for few-shot classification," in *CVPR*, pp. 7972–7981, 2022. 7

[66] D. Wertheimer, L. Tang, and B. Hariharan, "Few-shot classification with feature map reconstruction networks," in *CVPR*, pp. 8012–8021, 2021. 7, 8

[67] C. Xu, Y. Fu, C. Liu, C. Wang, J. Li, F. Huang, L. Zhang, and X. Xue, "Learning dynamic alignment via meta-filter for few-shot learning," in *CVPR*, pp. 5182–5191, 2021. 7

[68] M. Jeong, S. Choi, and C. Kim, "Few-shot open-set recognition by transformation consistency," in *CVPR*, pp. 12566–12575, 2021. 7

[69] D. Kang, H. Kwon, J. Min, and M. Cho, "Relational embedding for few-shot classification," in *ICCV*, pp. 8822–8833, 2021. 7

[70] Z. Zhou, X. Qiu, J. Xie, J. Wu, and C. Zhang, "Binocular mutual learning for improving few-shot classification," in *ICCV*, pp. 8402–8411, 2021. 7, 8

[71] X. Luo, L. Wei, L. Wen, J. Yang, L. Xie, Z. Xu, and Q. Tian, "Rectifying the shortcut learning of background for few-shot learning," in *NeurIPS*, pp. 13073–13085, 2021. 7

[72] J. Li, Z. Wang, and X. Hu, "Learning intact features by erasing-inpainting for few-shot classification," in *AAAI*, pp. 8401–8409, 2021. 7

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016. 6

[74] M. Chen, Y. Fang, X. Wang, H. Luo, Y. Geng, X. Zhang, C. Huang, W. Liu, and B. Wang, "Diversity transfer network for few-shot learning," in *AAAI*, pp. 10559–10566, 2020. 7

[75] L. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. 9, 10

[76] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv:2003.04297*, 2020. 9

[77] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *CVPR*, pp. 618–626, 2017. 9, 10

[78] K. Kim, B. Ji, D. Yoon, and S. Hwang, "Self-knowledge distillation with progressive refinement of targets," in *ICCV*, pp. 6567–6576, 2021. 10