BlockMix: Meta Regularization and Self-Calibrated Inference for Metric-Based Meta-Learning

Hao Tang, Zechao Li, Zhimao Peng and Jinhui Tang
School of Computer Science and Engineering, Nanjing University of Science and Technology
{tanghao0918,zechao.li,zhimaopeng,jinhuitang}@njust.edu.cn

ABSTRACT

Most metric-based meta-learning methods learn only the sophisticated similarity metric for few-shot classification, which may lead to the feature deterioration and unreliable prediction. Toward this end, we propose new mechanisms to learn generalized and discriminative feature embeddings as well as improve the robustness of classifiers against prediction corruptions for meta-learning. For this purpose, a new generation operator *BlockMix* is proposed by integrating interpolation on the images and labels within metric learning. Based on the above *BlockMix*, we propose a novel regularization method *Meta Regularization* as an auxiliary task branch with its own classifier to better constraint the feature embedding module and stabilize the meta-learning process. Furthermore, a novel inference scheme *Self-Calibrated Inference* is proposed to alleviate the unreliable prediction problem by calibrating the prototype of each category with the confidence-weighted average of the support and generated samples. The proposed mechanisms can be used as supplementary techniques alongside standard metric-based meta-learning algorithms without any pre-training. Experimental results demonstrate the insights and the efficiency of the proposed mechanisms respectively, compared with the state-of-the-art methods on the prevalent few-shot benchmarks.

CCS CONCEPTS

• **Information systems** → **Clustering and classification**.

KEYWORDS

Few-shot learning; representation learning; metric-learning

ACM Reference Format:
Hao Tang, Zechao Li, Zhimao Peng and Jinhui Tang. 2020. BlockMix: Meta Regularization and Self-Calibrated Inference for Metric-Based Meta-Learning . In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413884

1 INTRODUCTION

For many challenging tasks in the field of multimedia, deep learning methods have achieved great success [25, 26, 55]. To successfully learn a deep neural network model, millions of labeled data in thousands of categories are required. In practice, not only it is very expensive to collect such a large number of labeled data [46, 47], but also the data of some novel categories are scarce. In the meanwhile, few training examples often result in overfitting in the deep network, which greatly limits the applicability of learned models. However, human-level intelligence has a better generalization ability to classify categories that have been never seen before or observed with only few examples. Consequently, few-shot learning has attracted widespread attention due to relieving the above gap by training models to generalize to novel categories from few examples.

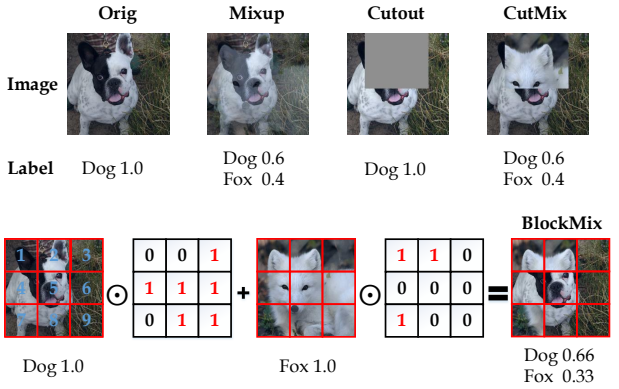


Figure 1: Visual comparisons of the proposed BlockMix (e.g., 3×3) with the Mixup [13], Cutout [8] and CutMix [57].

The most representative paradigm in existing works for few-shot learning is meta-learning [19, 48, 49]. Meta-learning methods train a network that can generalize to novel categories by learning some transferable knowledge from base classes with vast amounts of examples. These transferable knowledge has good network initialization [10], discriminative similarity metric [41, 51], efficient optimization strategy [36], etc. Among these meta-learning approaches, the most representative research direction is metric-based algorithm [30, 41, 44] which prompts the frontier of few-shot classification. Specifically, the features are extracted firstly from the support and query examples by the feature embedding module and then are used to build the nearest neighbor classifier based on the calculated distances on them. Besides, fine-tuning based methods [4, 11, 27] with standard transfer learning are proposed and also achieve encouraging results on unseen categories, compared with current state-of-the-art meta-learning methods. Therefore, the notable difference from the former is that the latter further explores the knowledge learned from base categories and generalizes it to the novel categories without meta-learning paradigm.

Corresponding author: Zechao Li.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413884

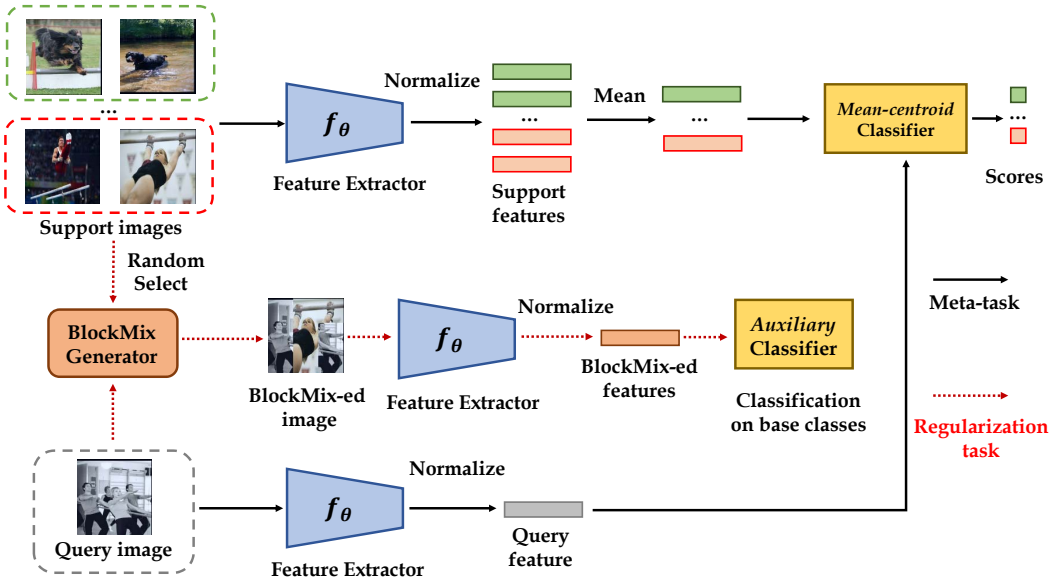


Figure 2: Framework illustration of the proposed Meta Regularization. The proposed method learns the generalized and discriminative feature embeddings as well as improves the robustness of classifiers against prediction corruptions. The solid lines branch learns the mean-centroid classifier for meta-task. Meanwhile, the dotted lines branch learns the auxiliary classifier for regularization task. Both branches are co-trained in a meta-learning paradigm.

In this paper, we are interested to take a close look at the metric-based meta-learning algorithms and come up with our motivations from two important observations: (1) **Feature deterioration**. Learning a robust and general-purpose category representation is the first step of the metric-based approach and the prototype of each support category is usually calculated by averaging the corresponding feature representations. However, the feature embedding module pre-trained on the base categories may lose its generalized and discriminative property on the novel categories. (2) **Unreliable prediction**. In the most metric-based approaches [3, 41, 44], each query sample is assigned to the support prototype with the closest distance to itself, where the support prototype is often directly calculated by averaging the feature representations. But this straightforward comparison may result in unreliable confidence and incorrectness in predictions. Thus how to represent the data distribution of each support category as true as possible with limited labeled samples remains challenging.

To alleviate the above issues, this work firstly proposes a new generation operator *BlockMix*: sub-blocks are randomly selected and mixed between two various images, where the ground truth labels are mixed proportionally according to the number of the combined blocks (See Figure 1). Then, a novel regularization method *Meta Regularization* is proposed to improve the generalization ability and reduce the uncertainty in predictions for metric-based meta-learning algorithms. In *Meta Regularization*, the sub-blocks between support and query examples are mixed in an episode where the ground truth labels are also mixed proportionally to the number of the combined blocks. Specifically, this method only uses the resources of the network itself to regularize the feature embedding module. It is common knowledge that the explicit regularization for posterior probability has been proven as an effective means [8, 32] to increase the robustness of the model. Therefore, we employ an auxiliary task co-training branch with its own classifier to constraint the feature embedding module and stabilize the meta-learning process, as illustrated in Figure 2. In practice, the feature embedding module is enforced to be shared between the meta-learning branch and the regularization branch, and the experimental results show that both branches are complementarity and force the predictions of each other not to be over-confident. To some extent, the *BlockMix* can be seen as a kind of regional dropout method sharing similarity with CutMix [57]. It essentially makes full use of the block-level pixels of training examples and maintains the effect of regional dropout regularization during the training.

In addition, we further propose a novel inference scheme called *Self-Calibrated Inference* for metric-based meta-learning algorithms, in which *BlockMix* can be explored to effectively alleviate the problem of unreliable prediction in the inference phase. Adopting the idea of pseudo-labeling [21] for reference, as a sample generation operator, *BlockMix* can adaptively generate diverse *BlockMix-ed* examples with semantically similar to the support example. The generated *BlockMix-ed* examples, as well as the support examples, are weighted summed as the enhanced prototype of the corresponding novel category. Specifically, this inference scheme has the **calibration property**: (1) *Confidence-based Mixing*. The mixing process of *BlockMix* is decided by predicted confidences from the softmax function, which makes generated semantically similar examples follow the true distribution of the novel category. (2) *Confidence-weighted Updating*. The generated examples are not treated equally and the category prototype is updated by the sum of weighted *BlockMix-ed* examples to further alleviate the intrinsic unreliability. Finally, the meta-classifier is discriminative enough to classify novel query examples correctly with the category prototype updated by multiple iterations.

Overall, the key contributions of this paper are summarized as follows:

- We propose a novel generation operator *BlockMix*: the sub-blocks are randomly selected and mixed between two different images while the labels are also mixed proportionally to the number of the mixed blocks. It can also be seen as a simple yet effective dropout strategy that can discard global structure and keep local details.
- An elaborative regularization method *Meta Regularization* is developed to conduct in an episodic manner, which can be as a flexible plug-and-play technique alongside standard metric-based meta-learning algorithms. To our best knowledge, it is the first time to explore the regularization strategy for metric-based meta-learning.
- A novel inference scheme *Self-Calibrated Inference* is proposed to alleviate the unreliable prediction issue. Specifically, the semantically similar example will be adaptively generated to enrich the prototype and make the calibrated meta-classifier correctly classify the novel categories, which is less explored in the inference for meta-learning algorithms and achieves state-of-the-art results.

2 RELATED WORKS

2.1 Regularization

Regularization is a common technique for training deep networks which makes deep models with better robustness and generalization. This is generally done by adding a few noises to some parameters during the training according to various modes. Dropout [42] and its variants have been proved to avoid over-fitting predictions and improve the generalization performance of models. Besides, label smoothing [45] and knowledge distillation [17] make efforts to alleviate the above issues by regularizing the posterior probability. Recently, regional dropout strategy[8] and its variants[13, 43, 57] have been proposed as a class of effective regularization, which is in the form of data augmentation. For example, Mixup [13] and CutMix [57] effectively perturb the input information to enhance the performance of the convolutional neural network during the training. In this paper, we mainly investigate a training framework that integrates a novel regularization method based on the proposed novel operator called *BlockMix* for meta-learning. Similar to Mixup [13] and CutMix [57], the proposed regularization method only needs to operate on the input data and easily is applied to any meta-learning paradigm for the few-shot classification.

2.2 Few-shot Learning

Few-shot learning aims to learn classifiers to generalize novel categories from few examples under given a large amount of labeled data from base categories. Some works [4, 19, 54] have shown tremendous success. In the view of the availability of the unlabeled query set, we can divide the few-shot classification researches into two groups: *inductive settings* and *transductive settings*.

Inductive Algorithm. Currently the main perspective for inductive few-shot learning is the meta-learning paradigm[48, 49], which designs the meta-learner to learn some meta-knowledge on the base tasks that can be quickly transferred to similar new tasks with scarce labeled data. Meta-learning methods can be roughly divided into several categories. *Optimization-based methods* [10, 36] aim to learn a good initialization that makes the model easily generalize to new tasks with only a small amount of gradient. *Parameter-generating based methods* [12, 35] usually learn to predict the classifier weights from the feature representations of the novel categories. *Metric-learning based methods* [41, 44, 51] learn a common feature space where the query set examples can be correctly classified based on the distance metric. In this work, our proposed mechanisms are built on the metric-learning based methods which consider the mean of feature embeddings as the category prototype to directly compare the distance between the query and support examples with a certain metric. Specifically, our work proposes to alleviate the feature deterioration by introducing the *Meta Regularization* method to restrain the feature embedding module for each episode.

In addition to the above methods, recent works [5, 6, 15, 53, 58] based on sophisticated sample generation gradually become another popular direction for few-shot classification. How to generate semantic-similar and high-quality data based on a few examples remains a challenging open problem. The proposed method shares similarity with chen et al. [5] that replaces the blocks between two samples, while the critical difference is that the latter uses a fixed augmented set to augment all images and needs to manually find the optimal replacement number. By contrast, the proposed *Meta Regularization* mixes the sub-blocks and labels between support and query examples on the fly in each training episode, and *BlockMix* can also be seamlessly integrated with the pseudo-labels and confidences computed from the model to enhance category prototypes in the inference phase. In addition, the proposed method is mainly for the meta-learning paradigm, but the latter is not.

Transductive Algorithm. Since the limited amount of labeled support examples in the few-shot classification, some works try to make full use of the unlabeled query examples, which is referred to as *transductive inference* [50]. Liu et al. [28] firstly introduces transductive inference in the few-shot classification by constructing a graph both on support and query sets and propagating labels to the unlabeled query examples within the graph. The work proposed by Kim et al. [20] is very similar to the [28], but the former utilizes the features of both edge and node in the update steps. In addition to the above methods, Ren et al. [37] proposes meta-learning for semi-supervised few-shot classification by utilizing unlabeled data to adjust the learned category prototype. Recently, Hou et al. [18] proposes to enrich the category features by picking top-k confident query samples with confidence criterion. The idea behind *Self-Calibrated Inference* is inspired by the *pseudo-labeling* strategy [21] in semi-supervised learning, where predictions of models are converted to hard labels and only be retained when the largest class probability is sufficiently confident. We propose to generate the semantic-similar examples with the *Blockmix* operator applied to pseudo-labeled examples to augment the labeled support set and the diversity of category prototypes can be self-calibrated enhanced in the inference stage.

3 THE PROPOSED METHOD

3.1 Preliminary

Given a dataset $\mathcal{D}$, it is divided into three subsets: a training set $\mathcal{D}_{train}$ with large amount of labeled examples, a support set $\mathcal{D}_{support}$

Algorithm 1 The proposed *Meta Regularization*

Input: The set of all query examples Q_x and labels Q_y
The set of all support examples S_x and labels S_y
The maximum number of mixed blocks N_{max}

for each episode **do**
  S_x, S_y, Q_x, Q_y = **get_episode**(dataset)
  if mode == training **then**
    $\tilde{S}_x, \tilde{S}_y$ = **shuffle** (S_x, S_y)
    $\mathbf{W} \in \{0,1\}^{N\times N} \longleftarrow N_{mix}$ = **Randint** $(1, N_{max}+1)$
    $\alpha = \frac{\sum\sum \mathbf{W}(i,j)}{N\times N} = 1 - \frac{N_{mix}}{N\times N}$
    input = $\mathbf{W} \odot Q_x + (\mathbf{1} - \mathbf{W}) \odot \tilde{S}_x$
    target = $\alpha * Q_y + (1-\alpha) * \tilde{S}_y$
  end if
  output = **model_forward**(input)
  $\mathcal{R}$ = **compute_loss**(output, target)
end for

with a few labeled examples and a query set $\mathcal{D}_{query}$ with unlabeled examples. The training set has a separate category space while the support set and query set share the same category space. The categories in $\mathcal{D}_{train}$ are defined as base categories C_{base}, and the categories in the $\mathcal{D}_{support}$ as well as $\mathcal{D}_{query}$ are defined as novel categories C_{novel}. That is, C_{base} is disjoint with C_{novel}. Meta learning aims to correctly classify images in query set $\mathcal{D}_{query}$ with prior meta-knowledge learned on the training set D_{train}, when given the support set $\mathcal{D}_{support}$. In particular, if the support set contains N unique categories and each category has K labeled examples, this few-shot learning problem is termed as the standard N-way K-shot classification scenario which firstly defined by Vinyals et al. [51]. In general, most previous meta-learning methods develop episodic training to mimic the few-shot learning setting. Specifically, in each training iteration process, an episode is constructed by a support set $\mathcal{S}_{train}$ and a query set $\mathcal{Q}_{train}$ that are sampled from the training set, which simulates the mode of the episodic meta-testing.

3.2 Sample Generation via BlockMix Operator

The goal of the *BlockMix* operator is to introduce new semantic information by generating a new training sample $(\tilde{x}, \tilde{y})$ given two distinct training samples (x_A, y_A) and (x_B, y_B). Here, $x \in \mathbb{R}^{W\times H\times C}$ and y denote one image and its label, respectively. Specifically, images are resized to the same fixed size and divided into $N \times N$ sub-blocks, and the *BlockMix* operator can be defined as

$$\begin{aligned} \tilde{x} &= \mathbf{W} \odot x_A + (\mathbf{1} - \mathbf{W}) \odot x_B, \\ \tilde{y} &= \alpha y_A + (1-\alpha) y_B, \end{aligned} \tag{1}$$

where $\mathbf{W} \in \{0,1\}^{N\times N}$ denotes a binary mask indicating each block in the new sample belongs to which of the two examples, $\mathbf{1}$ is a binary mask all filled with ones and $\odot$ denotes the element-wise multiplication. Specifically the blocks in x_A where the regions are filled with zeros in binary mask $\mathbf{W}$ are cutting out and replaced by the blocks cropped from x_B where the regions are filled with ones in binary mask $\mathbf{W}$. The combination ratio α between two samples is determined by the proportion of replaced regions to the original image, so the α is set to $\frac{\sum\sum \mathbf{W}(i,j)}{N\times N}$. In fact, the *BlockMix* operator is also as simple as other data augmentation methods [8, 13, 57] with negligible computational overhead, and we select the version of 3×3 sub-blocks (i.e., $N = 3$) as the final operator is shown in the bottom of Figure 1.

Algorithm 2 The proposed *Self-Calibrated Inference*

Input: The set of all query examples Q_{test}
The set of support examples S^c_{test}, for each category $c = 1, \ldots, C$
The number of update steps T
Output: Category prototype P^c_t updated after T steps.

for $c = 1, \ldots, C$ **do**
  Compute initial ($t = 0$) prototype $\mathcal{P}^c_0$ by Eq. (10);
end for
for $x \in Q_{test}$ **do**
  Compute initial ($t = 0$) pseudo-label $\hat{y}$ and confidence score $K(\hat{y}|x)$ by Eq. (11);
  Incorporated $(x, \hat{y}, K(\hat{y}|x))$ into corresponding pseudo-labeled set Q^c_{test};
end for
for $c = 1, \ldots, C$ **do**
  Generate corresponding BlockMix-ed set $\mathcal{S}^c_{mix}$;
end for
for $t = 1, \ldots, T$ **do**
  for $c = 1, \ldots, C$ **do**
    Update prototype P^c_t by Eq. (12);
  end for
  Compute pseudo-labeled query set $Q^{c,t}_{test}$, for all $c = 1, \ldots, C$
  Generate BlockMix-ed set $\mathcal{S}^{c,t}_{mix}$, for all $c = 1, \ldots, C$
end for

3.3 Metric Learning with Meta Regularization

The overall co-training pipeline of the metric-based meta-learning algorithm with *Meta Regularization* is shown in Figure 2, and the inputs are organized into two task branches: *meta-task* (solid lines) and *regularization task* (dotted lines).

The inputs of the *meta-task* branch consist of the following two parts: support set $\mathcal{S}$ and query set $\mathcal{Q}$, where $\mathcal{S}, \mathcal{Q} \in \mathcal{D}_{train}$. Following [31, 33, 35], all images are mapped to L_2–normalized feature vectors $f_\theta(\mathbf{x}) \in \mathbb{R}^C$ (i.e., $\|f_\theta(\mathbf{x})\|_2 = 1$) by the feature embedding module f_θ, where the parameters θ represent the weights of the feature embedding module. In practice, the *meta-task* is a N-way classification task, so we build a *mean-centroid classifier* $\mathbf{M} \in \mathbb{R}^{N\times C}$, each row $m_n \in \mathbb{R}^C$ in $\mathbf{M}$ represents the category prototype. The category prototype for the n-th category is the L_2–normalized average feature of all the K support examples that can be defined as follow:

$$\mathcal{P}_n = \frac{1}{K} \sum_{k=1}^{K} f_\theta(\mathbf{x}), \tag{2}$$

Finally, a query example $\mathbf{x}_q \in \mathcal{Q}$ is fed into constructed *mean-centroid classifier* and assigned to the nearest centroid's category by computing the distance between the query feature $f_\theta(\mathbf{x}_q)$ and category prototype $\mathcal{P}_n$. Maximizing the inner-product is equivalent to minimizing the Euclidean Distance between corresponding normalized vectors

$$\min d(f_\theta(\mathbf{x}_q), \mathcal{P}_n) \triangleq \max \mathcal{P}_n^\top f_\theta(\mathbf{x}_q), \tag{3}$$

Thus, the *meta-task* is trained by minimizing the following objective function:

$$\mathcal{L}_{\text{meta}} = -\log \frac{\exp\left(\eta \mathcal{P}_n^\top f_\theta(\mathbf{x}_q)\right)}{\sum_{i=1}^{N} \exp\left(\eta \mathcal{P}_i^\top f_\theta(\mathbf{x}_q)\right)}, \tag{4}$$

Here, η is a learnable parameter that can increase stability and robustness of the classifier. Obviously, the *mean-centroid classifier* is differentiable and can be updated by standard back-propagation according to the Eq. (4).

The inputs of the *regularization task* branch are generated by the *BlockMix Generator*, as shown in Figure 2. In each training episode, we first exploit *BlockMix* operator on the query set $\mathcal{Q}$ to generate a BlockMix-ed set $\mathcal{Q}_{mix}$. Specifically, for each image $(\mathbf{x}_q, \mathbf{y}_q) \in \mathcal{Q}$, we randomly select another image $(\mathbf{x}_s, \mathbf{y}_s) \in \mathcal{S}$ with a non-trivial probability that two different images have the same label, where $\mathbf{y}_s, \mathbf{y}_q \in C_{base}$. Then, a BlockMix-ed sample $(\tilde{x}, \tilde{y}) \in \mathcal{Q}_{mix}$ is generated by *BlockMix Generator* according to Eq. (1). In practice, we set a threshold N_{max} in *BlockMix Generator* which indicates the maximum number of blocks allowed to be mix, so the number N_{mix} of blocks to be mixed is sampled according to:

$$N_{mix} = \textbf{Randint}\,(1, N_{max} + 1)\,, \tag{5}$$

In order to alleviate the risk of mixing procedure biased towards a frequent pattern, we randomly determine the blocks to be mixed. The binary mask $\mathbf{W} \in \{0,1\}^{N \times N}$ is generated by filling with zeros within the corresponding mixed blocks, remaining parts are filled with ones. So, the lables of two examples are also mixed proportionally to the number of mixed blocks:

$$\begin{aligned} \alpha &= \frac{\sum_i \sum_j \mathbf{W}(i,j)}{N \times N} = 1 - \frac{N_{mix}}{N \times N}, \\ \tilde{y} &= \alpha \mathbf{y}_q + (1-\alpha)\mathbf{y}_s, \end{aligned} \tag{6}$$

In order to accelerate the convergence of training and get better generalization performance, the consistency with *meta-task* branch is maximized. To classify each BlockMix-ed sample among all available categories on the $\mathcal{D}_{train}$, the *regularization task* is jointly trained and shared the same feature embedding module with the *meta-task* branch. Different from the above *mean-centroid classifier*, the auxiliary classifier in *regularization task* is randomly initialized and updated via back-propagation. With the feature vectors $f_\theta(\tilde{x}) \in \mathbb{R}^C$ and the auxiliary classifier parameter w_c for category $c \in C_{base}$, the classification score s_c can be computed as follows:

$$s_c = \beta \cdot \frac{w_c^\top f_\theta(\tilde{x})}{\left\|w_c^\top\right\| \cdot \|f_\theta(\tilde{x})\|}, \tag{7}$$

β is also a learnable parameter. Thus, with the batch size B, the regularization task is trained by minimizing the following objective function:

$$\mathcal{R} = \alpha \sum_{\tilde{x},\tilde{y}}^{B} \left[-s_{\mathbf{y}_q} + \log \sum_{c=1}^{C_{\text{base}}} e^{s_c} \right] + (1-\alpha) \sum_{\tilde{x},\tilde{y}}^{B} \left[-s_{\mathbf{y}_s} + \log \sum_{c=1}^{C_{\text{base}}} e^{s_c} \right], \tag{8}$$

Finally, incorporating Eq. (4) and Eq. (8), the objective function is to minimize the following equation:

$$\mathcal{L} = \gamma_1\, \mathcal{L}_{\text{meta}} \;+\; \gamma_2\, \mathcal{R}, \tag{9}$$

where γ_1 and γ_2 are positive weighted coefficients between $\mathcal{L}_{\text{meta}}$ and $\mathcal{R}$. Please refer to Figure 2 for an illustrative representation of the proposed *Meta Regularization* strategy and the specific procedure is summarized in Algorithm 1.

3.4 Inference with Calibration Property

In the few-shot classification task, a single or few of labeled examples can not accurately represent the true data distribution. How to tackle this problem remains very challenging. Inspired by the concept of *Pseudo-labeling* [21] in semi-supervised learning, which assumes that unlabeled query exampls can be accessed and used, we propose a simple and effective inference algorithm called *Self-Calibrated Inference*. Specially, this inference scheme has calibration property which can adaptively mix support examples with semantically similar query examples and utilize the BlockMix-ed examples to enhance category prototype. The overall pipeline of the proposed algorithm is shown in Algorithm 2.

In the formula, $\mathcal{S}_{test}^c \in \mathcal{D}_{support}$ is defined as the support set for category c and $\mathcal{Q}_{test}$ is the query set consisting of all unlabeled query examples in one testing episode. In the inference process, the feature embedding module and *mean-centroid classifier* are supposed that they are all trained reasonably good in meta-learning with regularization strategy. Firstly, the original support set $\mathcal{S}_{test}^c$ is fed into the feature embedding module and *mean-centroid classifier*, the initial prototype $\mathcal{P}_0^c$ of each category $c \in \{1, \ldots, C\}$ can be computed according to Eq. (2):

$$\mathcal{P}_0^c = \frac{1}{|\mathcal{S}_{test}^c|} \sum_{x \in \mathcal{S}_{test}^c} f_\theta(x), \tag{10}$$

Given the normalized feature vectors $f_\theta(x)$ of query example $x \in \mathcal{Q}_{test}$ and the corresponding category prototypes, the pseudo label $\hat{y}$ and its confidence score $K(\hat{y}|x)$ are assigned according to Eq. (3) and Eq. (4):

$$\begin{aligned} \hat{y} &\triangleq \text{one-hot}\left(\arg\max_{c \in C} \eta \mathcal{P}_0^{c\top} f_\theta(x)\right), \\ K(\hat{y}|x) &= \frac{\exp\left(\eta \mathcal{P}_0^{\hat{y}\top} f_\theta(x)\right)}{\sum_{c'=1}^{C} \exp\left(\eta \mathcal{P}_0^{c'\top} f_\theta(x)\right)}, \end{aligned} \tag{11}$$

Next, the *mean-centroid* classifier iterates over all of the unlabeled query examples, and the pseudo-labeled query set $\mathcal{Q}_{test}^c = \{(x, \hat{y} = c, K(\hat{y}|x))\}$ can be obtained for each category $c \in \{1, \ldots, C\}$.

According to Eq. (1), for each support example $x^c \in \mathcal{S}_{test}^c$, the blocks will be mixed by each pseudo-labeled query example $(x, \hat{y})$ with $\hat{y} = c$ to generate BlockMix-ed sets $\mathcal{S}_{mix}^c = \{(\tilde{x}, \tilde{y}, K(\tilde{y}|\tilde{x})\}$ in which new example $\tilde{x}$ remains corresponding confidence score (i.e., $K(\tilde{y}|\tilde{x}) = K(\hat{y}|x)$) and pseudo label (i.e., $\tilde{y} = \hat{y} = c$). Then, the prototype of category c is calibrated based on the confidence scores for all $\tilde{x} \in \mathcal{S}_{mix}^c$:

$$\mathcal{P}_1^c = \frac{\sum_{x \in \mathcal{S}_{test}^c} 1 \cdot f_\theta(x) + \sum_{\tilde{x}, K \in \mathcal{S}_{mix}^c} K(\tilde{y} = c|\tilde{x}) \cdot f_\theta(\tilde{x})}{\sum_{\mathbf{x} \in \mathcal{S}_{test}^c} 1 + \sum_{\tilde{x}, K \in \mathcal{S}_{mix}^c} K(\tilde{y} = c|\tilde{x})}, \tag{12}$$

Finally, $\mathcal{P}_1^c$ is then used to re-assign pseudo label for each unlabeled query example. The above process iterates step T to progressively generate a more representative and robust category prototype $\mathcal{P}_t^c$.

Table 1: The average Few-shot classification results with 95% confidence intervals on the MiniImageNet dataset.

Methods	Ref.	Backbone	1-shot 5-way	5-shot 5-way
MatchNet [51]	NeurIPS'16	Conv-64F	43.56 ± 0.84	55.31 ± 0.73
ProtoNet [41]	NeurIPS'17	Conv-64F	49.42 ± 0.78	68.20 ± 0.66
MM-Net [1]	CVPR'18	Conv-64F	53.37 ± 0.48	66.97 ± 0.35
RelationNet [44]	CVPR'18	Conv-64F	50.44 ± 0.82	65.32 ± 0.70
TADAM [30]	NeurIPS'18	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
STANet [56]	AAAI'18	ResNet-12	58.35 ± 0.57	71.07 ± 0.39
DN4 [24]	CVPR'19	Conv-64F	51.24 ± 0.74	71.02 ± 0.64
DCEM [9]	ICCV'19	ResNet-18	58.71 ± 0.62	77.28 ± 0.46
LEO [39]	ICLR'19	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12
VMS [2]	AAAI'20	ResNet-12	60.16 ± 0.47	77.25 ± 0.15
DTN [3]	AAAI'20	ResNet-12	60.72 ± 0.72	76.58 ± 0.65
MetaGAN [59]	NeurIPS'18	Conv-32F	52.71 ± 0.64	68.63 ± 0.67
Dual TriNet [7]	TIP'19	ResNet-18	58.80 ± 1.37	76.71 ± 0.69
Δ-encoder [40]	NeurIPS'18	VGG-16	59.50	69.70
IDeMe-Net [6]	CVPR'19	ResNet-18	59.14 ± 0.86	74.63 ± 0.74
SalNet [58]	CVPR'19	ResNet-101	62.22 ± 0.87	77.95 ± 0.65
AFHN [23]	CVPR'20	ResNet-18	62.38 ± 0.72	78.16 ± 0.56
PN_Cos	Ours	ResNet-12	59.11 ± 0.85	72.29 ± 0.67
PN_Cos+*MR*	Ours	ResNet-12	**62.48 ± 0.85**	**78.20 ± 0.59**
TPN [28]	ICLR'19	ResNet-12	59.46	75.65
TEAM [34]	ICCV'19	ResNet-18	60.07	72.04
CAN [18]	NeurIPS'19	ResNet-12	67.19	**80.64**
PN_Cos+*MR*+*SCI*^{1}	Ours	ResNet-12	67.45	80.02
PN_Cos+*MR*+*SCI*^{2}	Ours	ResNet-12	69.15	80.12
PN_Cos+*MR*+*SCI*^{3}	Ours	ResNet-12	**69.79**	80.19

MR: Meta Regularization, SCI^t: Self-Calibrated Inference ($t \geq 1$)

More importantly, the proposed inference scheme has the **calibration property** without further training to robustly against the wrong predictions on unlabeled query exampls (i.e., $\hat{y} \neq c$).

(1) **Confidence-based Mixing.** Different from meta-learning with regularization strategy, the mixing process in *BlockMix Generator* is controlled adaptively according to the confidence score $K(\tilde{y}|\tilde{x})$. We randomly determine the position of mixed blocks, but the number of mixed blocks is computed as:

$$n_{mix} = \sum_i \sum_j \mathbf{W}(i,j) = \textbf{Round}\left(K \times N \times N\right). \tag{13}$$

(2) **Confidence-weighted Updating.** To further alleviate the intrinsic unreliability, we add a weighted sum for the all support examples to more robustly update the category prototype. Specifically, we consider the weight of each generated example in $\tilde{x} \in \mathcal{S}^c_{mix}$ as the same as the confidence score of corresponding pseudo-labled query example. Note that the weight of the original support example is always set to 1.

4 EXPERIMENTS

4.1 Datasets

Extensive experiments are conducted on the standard few-shot classification dataset: MiniImageNet [51] and another widely used fine-grained benchmark: CUB-200-2011 [52] to evaluate the effectiveness of the proposed method for metric-based meta-learning. To verify the method comprehensively, these datasets cover from general objects to fine-grained images performed with two settings (i.e., 1-shot and 5-shot classification).

MiniImageNet. This dataset is a mini-version of the full ImageNet dataset [38], and contains 100 different categories with 600 images per category. We follow the settings in [36] by splitting 64, 16 and 20 categories as the training set, validation set, and testing set, respectively.

Table 2: The average Few-shot classification results with 95% confidence intervals on the CUB-200-2011 dataset.

Methods	Ref.	Backbone	1-shot 5-way	5-shot 5-way
MatchNet [51]	NeurIPS'16	Conv-64F	49.34	59.31
ProtoNet [41]	NeurIPS'17	Conv-64F	45.27	56.35
RelationNet [44]	CVPR'18	ResNet-18	67.59 ± 1.02	82.75 ± 0.58
DN4 [24]	CVPR'19	Conv-64F	53.15 ± 0.84	81.90 ± 0.60
Baseline++ [4]	ICLR'19	ResNet-18	67.68 ± 0.23	82.26 ± 0.15
SAML [14]	ICCV'19	Conv-64F	69.33 ± 0.22	81.56 ± 0.15
DTN [3]	AAAI'20	ResNet-12	72.0	85.1
DualTriNet [7]	TIP'19	ResNet-18	69.61 ± 0.46	84.10 ± 0.35
Δ-encoder [40]	NeurIPS'18	VGG-16	69.80 ± 0.46	82.60 ± 0.35
AFHN [23]	CVPR'20	ResNet-18	70.53 ± 1.01	83.95 ± 0.63
PN_Cos	Ours	ResNet-12	70.78 ± 0.89	82.47 ± 0.62
PN_Cos+*MR*	Ours	ResNet-12	**75.31 ± 0.79**	**88.53 ± 0.49**
TEAM [34]	ICCV'19	ResNet-18	80.16	87.17
PN_Cos+*MR*+*SCI*^{1}	Ours	ResNet-12	81.40	90.07
PN_Cos+*MR*+*SCI*^{2}	Ours	ResNet-12	83.13	90.17
PN_Cos+*MR*+*SCI*^{3}	Ours	ResNet-12	**83.82**	**90.22**

MR: Meta Regularization, SCI^t: Self-Calibrated Inference ($t \geq 1$)

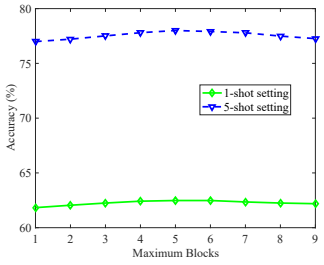
CUB-200-2011. This dataset is one widely-used image dataset for fine-grained object recognition and contains 11,788 images from 200 bird species in total. We follow the commonly-used evaluation protocol proposed in [4], which randomly splits 100, 50, 50 categories to construct the training set, validation set, and testing set, respectively.

4.2 Implementation Details

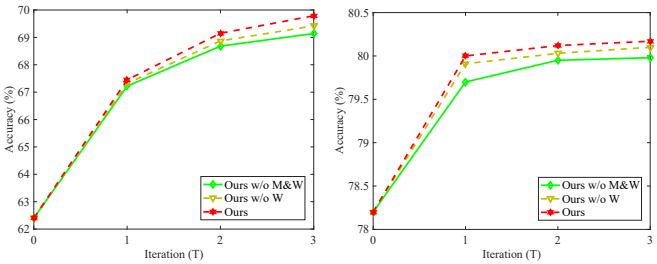
Following the basic experimental settings in [3], the ResNet-12 [16] network is used as the visual feature extractor with the same structure as former methods [22, 29, 30] for a fair comparison. The ResNet-12 network is composed of 4 residual blocks, each of which consists of 3 convolution layers. The number of filters is set to 32, 64, 128 and 256 respectively and the size of all filters is set to 3×3. Besides, all the convolutional layers are followed by a Batch Normalization layer, a LeakyReLU nonlinearity layer, and each residual block is followed by a 2×2 max-pooling layer. In all experiments, we adopt the SGD optimizer with Nesterov momentum of 0.9 and weight decay of 0.0005 to train 100 epoch from scratch. The initial learning rate is set to 0.05 and change to 0.4 times of itself every 20 epoch. In each episode, we randomly sample 5 examples and 15 examples from each novel category as the query set during meta-training and meta-testing, respectively. The coefficients γ_1 and γ_2 in Eq. (9) are set to the same value (i.e., $\gamma_1 = \gamma_2 = 0.5$) for all experiments. Considering both the performance and efficiency synthetically, the number of sub-blocks $N \times N$ in *BlockMix* operator is set to 3×3. In order to reduce calculational cost and speed up the inference, the hyper-parameter N_{max} in *Meta Regularization* is set to 5 and the maximum of iterations T in *Self-Calibrated Inference* is set to 3 by default. Note that, the meta-trained model and hyper-parameters are chosen based on the validation set with 5-way 1-shot test accuracy.

4.3 Experimental Results

To verify the effectiveness of the proposed method for the metric-based meta-learning, we conduct the proposed *Meta Regularization*



(a) the impact of the hyper-parameter



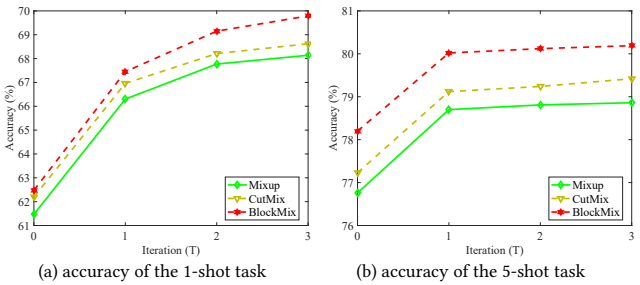
(b) the effectiveness of the calibration property (left: 1-shot task, right: 5-shot task)

Figure 3: The results of ablation studies on the Meta Regularization shown in (a) and Self-Calibrated Inference shown in (b). Best viewed with zoom.

strategy and *Self-Calibrated Inference* on a baseline and compare it with some state-of-the-art methods. All results are summarized in Table 1 and Table 2.

MiniImageNet. Table 1 presents comparison results on MiniImageNet. It can be easily observed that the proposed PN_Cos (*baseline*) with *Meta Regularization* (*MR*) achieves competitive performance compared with several state-of-the-art few-shot learning approaches for 1-shot and 5-shot classification tasks. After applying the *Meta Regularization* strategy to the baseline, the absolute promotion is 3.37% for 1-shot task and 5.91% for 5-shot task respectively. The results well show the effectiveness of the proposed method and the rationality of our motivation. Note that the comparisons with MetaGAN [59], DualTriNet [7], Δ-encoder [40], IDeMe-Net [6], SalNet [58], AFHN [23] are a little unfair for the proposed method, since these methods belong to generation-based methods while our approach (PN_Cos+*MR*) has no any additional examples to expand the diversity of novel category. Even so, the proposed approach is still slightly better than AFHN [23]. In addition to the above settings, we also conduct some experiments with the *Self-Calibrated Inference* (*SCI*) which can augment the novel category with the example generation. The bottom rows of Table1 show the results of our proposed PN_Cos+*MR*+SCI^t ($t \geq 1$), compared with current transductive few-shot learning approaches in which the model utilizes the entire query set in each episode. It can be seen that the improvement margin of 1-shot task is considerable where the labeled support example is extremely limited. Especially, the accuracy of 5-shot task in CAN [18] is slightly higher than the proposed method, but the former relies on its complex attention components. The proposed method achieves the best performance on the 1-shot task.

CUB-200-2011. Fine-grained few-shot classification task is more challenging than the generic one due to the smaller inter-class and



(a) accuracy of the 1-shot task (b) accuracy of the 5-shot task

Figure 4: Quantitative comparisons of the proposed BlockMix operator with Mixup [13] and CutMix [57]. In Meta Regularization and Self-Calibrated Inference, BlockMix provides better results that demonstrate the effectiveness of this technique.

larger intra-class variations. From Table 2, it shows that the performance of our baseline with *Meta Regularization* strategy (PN_Cos+*MR*) is far better than the previous methods. Compared with the baseline, the absolute promotion is 4.53% for 1-shot task and 6.06% for 5-shot task respectively, which further indicates the effectiveness of our *Meta Regularization* strategy. When using the *Self-Calibrated Inference* during meta-testing, the PN_Cos+*MR*+SCI^3 achieves significant performance over 1-shot and 5-shot tasks. Note that the margin of the improvement on CUB-200-2011 is larger than that on MiniImageNet. The reason may be that *Meta Regularization* mainly mixes the sub-blocks between the support and query examples which leads to discard global structure and keep local details. Based on the above analysis, the proposed *BlockMix* can force the model to identify and focus on the discriminative local regions for fine-grained recognition.

Experiments cross two different benchmark datasets indicate that: (1) The proposed method can consistently improve the performance of few-shot classification on different datasets. This shows that the metric-based meta-learning algorithm can efficiently alleviate feature deterioration and unreliable prediction with the improvement of the proposed method. (2) The more available support examples, the less improved performance. Especially, the performance promotion of our method in 1-shot task is more significant than that in 5-shot task and this similar phenomena also appears in [28, 34], which conforms to the nature of transduction inference.

5 ABLATION STUDIES

We further conduct ablation studies on the MiniImageNet dataset to verify the effect of the key components in the proposed method, and the results are shown in Figure 3.

Impact of hyper-parameter: Considering that the position and number of blocks are randomly sampled in *Meta Regularization*, the hyper-parameter N_{max} indicating the maximum number of mixed blocks in Eq. 5 needs to be tuned for the optimal performance of *Meta Regularization*. The impact of varying hyper-parameter across the range of 1 to 9 is presented in Figure 3 (a) and the best performance trade-off is achieved when $N_{max} = 5$. It may be that there exists a non-trivial probability that the mixed blocks contain the objects of interest rather than the background. The proposed method achieves a reasonable balance between replacement and

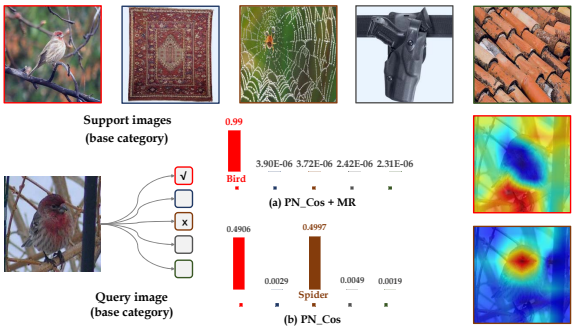


Figure 5: The effectiveness of the BlockMix operator within the Meta Regularization on the base (seen) category. We find that the Meta Regularization can force the model to focus on more discriminative regions under a given object for the 1-shot task. As a result, the metric-based method can correctly classify the above query image. Best viewed in color with zoom.

reservation of block-level information on the images, which makes the *BlockMix* more robust on images.

Effectiveness of calibration property: To verify the effectiveness of the *Self-Calibrated Inference* method, two different variants: without Mixing and Weighting (i.e., w/o **M&W**), and without Weighting (i.e., w/o **W**) are evaluated. The former is seen as a pseudo labeling approach and the latter can be seen as a straightforward data generation approach. The contribution of each component and the evaluation results on MiniImagenet are shown in Figure 3 (b), and the proposed method achieves the best performance obviously. Although the results of pseudo labeling gradually become better and more stable as the iteration T increases, it is still worse than the results obtained by the proposed algorithm with calibration property. Especially, it can be observed that the accuracy gap among the three different inference configurations is subtle when firstly updating the category prototype for 1-shot settings, but this gap is widening as the iteration T increases. This supports our hypothesis: (1) Calibration property enables the model to robustly alleviate wrong predictions under the help of confidence-based mixing and confidence-weighted updating. (2) The semantic information introduced from the *BlockMix-ed* examples is also robust enough, which can be used to represent the real data distribution of the novel category.

6 DISCUSSION

Though the proposed *BlockMix* shows the significant improvement for the metric-based meta-learning algorithms. However, it's hard to fathom the theoretical foundation of its effectiveness.

For the *BlockMix* operator in the proposed *Meta Regularization*, the input image is firstly divided into local blocks and then the sub-blocks between the support and query examples are mixed, which discards the global structure of main classification object and makes efficient use of training pixels. Moreover, the blocks are selected from random locations and mixed onto the same area in the BlockMix-ed image as in the original image. To correctly recognize BlockMix-ed images, the feature embedding module has to pay more attention to discriminative regions under a given object. This

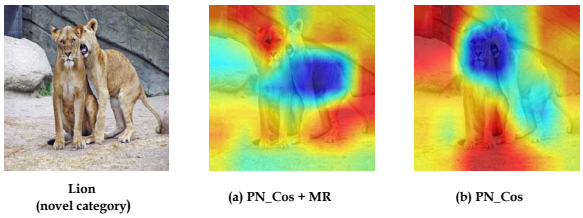


Figure 6: The effectiveness of the BlockMix operator within the Meta Regularization on the novel (unseen) category. Although deep model is biased toward the base (seen) objects, the metric-based embedding module with the proposed Meta Regularization strategy can enhance the discrimination and richness of information that might be useful for unseen identities.

regularization strategy increases the difficulty of metric learning and makes it harder for overfitting. A representative comparison of the proposed *BlockMix* with other strategies is shown in Figure 1, and the *BlockMix* can be seen as a complex version of CutMix with multi-regions. To demonstrate the effectiveness of *BlockMix*, we further replace the *BlockMix* operator in the proposed *Meta Regularization* and *Self-Calibrated Inference*, and the quantitative results are shown in Figure 4. It can be observed that the proposed BlockMix operator outperforms Mixup [13] and CutMix [57] in all cases, especially in *Self-Calibrated Inference*.

In addition, the effect visualizations of the *BlockMix* are shown in Figure 5 and Figure 6. The class activation maps on the base category and novel category show that the *BlockMix* can help the metric-based meta-learning model to focus on more informative regions. We believe that this lightweight operator can also be easily plugged into other meta-learning algorithms and achieve similar gains, since the robust and discriminative features extracted from the images are also the main foundation for the few-shot classification task.

7 CONCLUSIONS

In this work, we first define a novel and feasible generation operator *BlockMix*. Then, a scalable regularization strategy *Meta Regularization* and a novel inference scheme *Self-Calibrated Inference* are proposed for the metric-based meta-learning. The former leverages the mixed sub-blocks of support examples and query examples to help learn a better feature embedding space for generalizing to novel categories. The latter can generate semantically similar examples under the help of the entire query set and take a weighted combination of examples to enrich the category prototype progressively. The proposed method can be easily plugged into existing metric-based meta-algorithms with negligible computational overhead. Comprehensive experimental results show that the proposed method achieves competitive performance against the state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China under Grant 2018AAA0102002, the National Natural Science Foundation of China (Grant No. 61720106004 and 61732007) and the Natural Science Foundation of Jiangsu Province (Grant BK20170033).

REFERENCES

[1] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. 2018. Memory Matching Networks for One-Shot Image Recognition. In *CVPR*.

[2] Jiaxin Chen, Li-Ming Zhan, Xiao-Ming Wu, and Fu-lai Chung. 2020. Variational Metric Scaling for Metric-Based Meta-Learning. In *AAAI*.

[3] Mengting Chen, Yuxin Fang, Xinggang Wang, Heng Luo, Yifeng Geng, Xinyu Zhang, Chang Huang, Wenyu Liu, and Bo Wang. 2020. Diversity Transfer Network for Few-Shot Learning. In *AAAI*.

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A Closer Look at Few-shot Classification. In *ICLR*.

[5] Zitian Chen, Yanwei Fu, Kaiyu Chen, and Yu-Gang Jiang. 2019. Image Block Augmentation for One-Shot Learning. In *AAAI*.

[6] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. 2019. Image Deformation Meta-Networks for One-Shot Learning. In *CVPR*.

[7] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. 2019. Multi-Level Semantic Feature Augmentation for One-Shot Learning. *IEEE Transactions on Image Processing* 28, 9 (2019), 4594–4605.

[8] Terrance Devries and Graham W. Taylor. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv preprint arXiv:1708.04552* (2017).

[9] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. 2019. Diversity With Cooperation: Ensemble Methods for Few-Shot Classification. In *ICCV*.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

[11] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. 2019. Boosting Few-Shot Visual Learning With Self-Supervision. In *ICCV*.

[12] Spyros Gidaris and Nikos Komodakis. 2018. Dynamic Few-Shot Visual Learning Without Forgetting. In *CVPR*.

[13] Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. MixUp as Locally Linear Out-of-Manifold Regularization. In *AAAI*.

[14] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. 2019. Collect and Select: Semantic Alignment Metric Learning for Few-Shot Learning. In *ICCV*.

[15] Bharath Hariharan and Ross B. Girshick. 2017. Low-Shot Visual Recognition by Shrinking and Hallucinating Features. In *ICCV*.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[17] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* (2015).

[18] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. 2019. Cross Attention Network for Few-shot Classification. In *NeurIPS*.

[19] Muhammad Abdullah Jamal and Guo-Jun Qi. 2019. Task Agnostic Meta-Learning for Few-Shot Learning. In *CVPR*.

[20] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. 2019. Edge-Labeling Graph Neural Network for Few-Shot Learning. In *CVPR*.

[21] Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshops*.

[22] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-Learning With Differentiable Convex Optimization. In *CVPR*.

[23] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. 2020. Adversarial Feature Hallucination Networks for Few-Shot Learning. In *CVPR*.

[24] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. 2019. Revisiting Local Descriptor Based Image-To-Class Measure for Few-Shot Learning. In *CVPR*.

[25] Zechao Li and Jinhui Tang. 2015. Weakly Supervised Deep Metric Learning for Community-Contributed Image Retrieval. *IEEE Transactions on Multimedia* 17, 11 (2015), 1989–1999.

[26] Zechao Li, Jinhui Tang, and Tao Mei. 2018. Deep Collaborative Embedding for Social Image Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 9 (2018), 2070–2083.

[27] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. 2019. Dense Classification and Implanting for Few-Shot Learning. In *CVPR*.

[28] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. 2019. Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning. In *ICLR*.

[29] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A Simple Neural Attentive Meta-Learner. In *ICLR*.

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

[31] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. 2019. Few-Shot Image Recognition With Knowledge Transfer. In *ICCV*.

[32] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. In *ICLR*.

[33] Hang Qi, Matthew Brown, and David G. Lowe. 2018. Low-Shot Learning With Imprinted Weights. In *CVPR*.

[34] Limeng Qiao, Yemin Shi, Jia Li, Yonghong Tian, Tiejun Huang, and Yaowei Wang. 2019. Transductive Episodic-Wise Adaptive Metric for Few-Shot Learning. In *ICCV*.

[35] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. 2018. Few-Shot Image Recognition by Predicting Parameters From Activations. In *CVPR*.

[36] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-Shot Learning. In *ICLR*.

[37] Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. 2018. Meta-Learning for Semi-Supervised Few-Shot Classification. In *ICLR*.

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[39] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. 2019. Meta-Learning with Latent Embedding Optimization. In *ICLR*.

[40] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogério Schmidt Feris, Raja Giryes, and Alexander M. Bronstein. 2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *NeurIPS*.

[41] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *NeurIPS*.

[42] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[43] Cecilia Summers and Michael J. Dinneen. 2019. Improved Mixed-Example Data Augmentation. In *WACV*.

[44] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *CVPR*.

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*.

[46] Jinhui Tang, Xiangbo Shu, Zechao Li, Yu-Gang Jiang, and Qi Tian. 2019. Social Anchor-Unit Graph Regularized Tensor Completion for Large-Scale Image Retagging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2019), 2027–2034.

[47] Jinhui Tang, Xiangbo Shu, Guo-Jun Qi, Zechao Li, Meng Wang, Shuicheng Yan, and Ramesh C. Jain. 2017. Tri-Clustered Tensor Completion for Social-Aware Image Tag Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2017), 1662–1674.

[48] Sebastian Thrun. 1998. Lifelong Learning Algorithms. In *Learning to Learn*. Springer, 181–209.

[49] Sebastian Thrun and Lorien Y. Pratt. 1998. Learning to Learn: Introduction and Overview. In *Learning to Learn*. Springer, 3–17.

[50] Vladimir Vapnik. 1998. *Statistical learning theory*. Wiley.

[51] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *NeurIPS*.

[52] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. *The Caltech-UCSD Birds-200-2011 Dataset.* Technical Report CNS-TR-2011-001. California Institute of Technology.

[53] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. 2018. Low-Shot Learning From Imaginary Data. In *CVPR*.

[54] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys* 53, 3 (2020), 63:1–63:34.

[55] Rui Yan, Jinhui Tang, Xiangbo Shu, Zechao Li, and Qi Tian. 2018. Participation-Contributed Temporal Dynamic Model for Group Activity Recognition. In *ACM MM*.

[56] Shipeng Yan, Songyang Zhang, and Xuming He. 2019. A Dual Attention Network with Semantic Embedding for Few-Shot Learning. In *AAAI*.

[57] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *ICCV*.

[58] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. 2019. Few-Shot Learning via Saliency-Guided Hallucination of Samples. In *CVPR*.

[59] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. 2018. MetaGAN: An Adversarial Approach to Few-Shot Learning. In *NeurIPS*.