Task-aware Part Mining Network for Few-Shot Learning

Jiamin Wu, Tianzhu Zhang*, Yongdong Zhang, Feng Wu
University of Science and Technology of China
jiaminwu@mail.ustc.edu.cn, {tzzhang, zhyd73, fengwu}@ustc.edu.cn

Abstract

Few-Shot Learning (FSL) aims at classifying samples into new unseen classes with only a handful of labeled samples available. However, most of the existing methods are based on the image-level pooled representation, yet ignore considerable local clues that are transferable across tasks. To address this issue, we propose an end-to-end Task-aware Part Mining Network (TPMN) by integrating an automatic part mining process into the metric-based model for FSL. The proposed TPMN model enjoys several merits. First, we design a meta filter learner to generate task-aware part filters based on the task embedding in a meta-learning way. The task-aware part filters can adapt to any individual task and automatically mine task-related local parts even for an unseen task. Second, an adaptive importance generator is proposed to identify key local parts and assign adaptive importance weights to different parts. To the best of our knowledge, this is the first work to automatically exploit the task-aware local parts in a meta-learning way for FSL. Extensive experimental results on four standard benchmarks demonstrate that the proposed model performs favorably against state-of-the-art FSL methods.

1. Introduction

Deep Convolutional Neural Networks (CNNs) have achieved tremendous success in a wide range of computer vision tasks [43, 14, 20, 37, 29, 54, 55]. However, the training of CNNs requires large amounts of annotated images, which are prohibitively expensive to collect [2]. Few-Shot Learning (FSL) [12, 11, 48, 44] is promising in reducing the need for human annotation, which aims at learning a model with good generalization capability such that it can classify unlabeled samples (query set) into new unseen classes with only one or a few examples (support set). Usually, a few-shot classifier is trained to solve N-way K-shot tasks that consist of N classes and K support samples per class.

To tackle the FSL problem, a series of previous work [44, 46, 13, 23, 22, 52] adopts a metric-based learning model, which firstly learns a good embedding space, and then selects a distance metric to directly compute the distances between the query and support images for classification. However, most of these methods utilize the image-level pooled representation for classification, which could lose considerable discriminative local clues that enjoy favorable transferability across classes [53]. Recently, several methods [24, 25, 53, 9] consider taking advantage of the local representation for FSL. These methods decompose images into a set of local patches by grid dividing on the feature map. Then the image-level distance measurement between a pair of the query and the support images is achieved by aggregating the similarities between local patches.

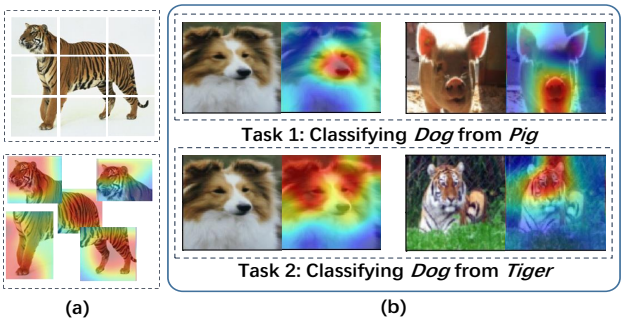

Figure 1. Illustration of our motivation. (a) shows the comparison of different local representations. The first row presents the local patches derived by grid dividing in the previous methods. The second row shows the local parts learned in an automatic way. The latter achieves better discriminability and contains fewer background noises. (b) shows the task-aware mechanism. For the same image of *dog*, the attended local parts change with different tasks.

By studying the previous FSL methods based on local representations [24, 25, 53, 9], we sum up three characteristics that are imperative for building a robust few-shot model. (1) **Automaticity.** In the previous methods, the local patches obtained by the pre-defined decomposition strategy suffer from large randomness. Some local patches have high possibilities to cover only a small part of the semantic regions of the object, or even completely the background regions (see the first row in Figure 1 (a)). This randomness could cause misalignments when matching the local patches. However, the human can quickly recognize a new category by automatically segmenting the objects into mul-

*Corresponding Author

tiple local parts (see the second row in Figure 1 (a)) and comparing them to the similar object parts that have already been seen. The automaticity in the part mining process ensures natural semantic correspondences between the local parts. (2) **Task-Aware Mechanism.** In previous methods, a common set of local regions is shared across tasks, which may not handle the diverse tasks with large distribution differences well. As each individual task consists of a unique set of categories, the local regions that are effective for the current task may not always satisfy the needs of other tasks. For example, when differentiating *dog* from *tiger*, humans pay more attention on the texture of part *face*, while when recognizing *dog* and *pig*, the structure of part *nose* is obviously more essential (see Figure 1 (b)). Even for the same image, the importance degrees of the local parts vary from task to task. Therefore, without a task-aware mechanism, it is challenging to make the learned model generalize to novel classes well. (3) **Adaptive Weights.** When making the final predictions, suppressing the importance of irrelevant regions can avoid introducing noises, while more discriminative regions should enjoy higher weights. Therefore the importance weights of different local parts should be adaptively assigned.

Inspired by the above insights, we propose an end-to-end Task-aware Part Mining Network (**TPMN**) to integrate the automatic local part mining process into the metric-based FSL. To achieve the **automaticity** in discovering local part regions, we firstly introduce a set of Part Filters (PFs) to automatically generate part-aware activation maps. The PFs are parameterized by multiple learnable 1×1 convolution kernels, which can activate the spatial attention that covers a certain local part on the feature map. The resulted activation maps are used as part masks to obtain multiple discriminative part-aware features for both support and query images. However, the above PFs, once are learned, are fixed and shared across tasks, which are incapable of handling various tasks well. Therefore, to equip the network with the **task-aware mechanism**, we design a Meta Filter Learner (MFL) to flexibly generate the parameters of the PFs that are customized for an individual task, in a meta-learning way. This is achieved by incorporating the task embedding, which expresses the unique categorical information of the specific task. MFL establishes the connection between the task embedding and part filters. Intuitively speaking, the parameters of the PFs will be derived by transforming the task embedding into the parameter space. In this way, the task-aware PFs are able to adapt from task to task and can provide the most desired information for any tasks, even the tasks with unseen classes. Then, to determine the image-level similarity for the given query and support image, we assign **adaptive weights** to local parts by an adaptive importance generator, such that the less relevant parts will be suppressed while the discriminative ones are highlighted. Finally, the image-level similarity is derived as the weighted aggregation of local similarities between part-aware features from different images.

The contributions of our method could be summarized into three-fold: (1) We propose an end-to-end Task-aware Part Mining Network by jointly exploiting the automatic local part mining process and the meta-learning strategy. (2) We design a meta filter learner to generate task-aware part filters, which can discover task-related local parts even for the unseen tasks. Also, an adaptive importance generator is proposed to assign importance weights for local parts. To our best knowledge, this is the first work to exploit discriminative local parts in a meta-learning way for FSL. (3) Extensive experimental results on four challenging benchmarks demonstrate that our method performs favorably against state-of-the-art FSL methods.

2. Related Work

In this section, we introduce several lines of research in few-shot learning, local representation learning in FSL, and part-aware attention mechanism.

Few-Shot Learning. Existing FSL methods can be generally divided into three groups: **(1) Metric-based** methods [48, 44, 46, 47, 26, 42, 52] learn a discriminative embedding space for their chosen distance metrics. MatchingNet [48] and ProtoNet [44] perform classification by computing the similarities or distances between support and query samples. There are also interesting methods that directly learn a deep distance metric, *e.g.*, using the CNN-based relation module to produce the relation score [46], utilizing graph neural network to infer the edge strength [41, 18, 50]. **(2) Gradient-based** methods [12, 34, 3, 45, 16] design the meta-learner as an optimizer to adapt model parameters to new tasks in the low-shot regime. MAML [12] and many of its variants [3, 45, 16] aim to learn a good model initialization so that the learner can rapidly adapt to novel tasks. In [34], an LSTM-based meta-learner is trained as the optimizer that learns to update model parameters, for replacing the SGD optimizer. **(3) Generation-based** methods [5, 33, 32, 30, 38] usually develop a meta-learner as a parameter predicting network to generate task-specific networks given few novel class samples. [32] and [33] generate the weights of the classification layer from extracted features. [30] and [38] modulate the feature maps with scaling and shifting parameters predicted from the current task input. Our proposed method belongs to the metric-based methods. However, a key difference lies in that the above methods generally adopt a global representation for metric-based classification, while our method focuses on automatically mining multiple local object parts.

Local Representation Learning in FSL. Some FSL methods [25, 24, 9, 53] attempt to exploit the discriminative power of local representations. The basic idea is to consider each spatial location in the feature map as a local

patch, and gather the patch-level distances as image-level distance. DC [25] performs dense classification on each local feature and fuses the results for the final prediction. DN4 [24] adopts k-NN selection on the patch distance matrix to fuse highly-related distances, and ATL-Net [9] makes a refinement by applying an episodic attention mechanism to select important patches. DeepEMD [53] performs a many-to-many matching between local patches and gets the global distance by solving an optimization problem of Earth Mover's distance. However, none of these methods explicitly design a task-adaptive mechanism to dynamically adapt the local features to different tasks. Our method utilizes a meta-learning strategy to automatically localize multiple object parts as the local representations, which is task-aware and can generalize to an arbitrary unseen task well.

Part-Aware Attention Mechanism. Attention mechanisms aim to highlight important regions to extract more discriminative features. Several methods [31, 56, 8] utilize attention mechanisms to exploit multiple object parts and learn complementary representation from these parts. In [31], selective search is adopted to mine candidate local patches and remove noisy patches by threshold filtering. [56] clusters the feature channels to produce multiple part attention maps. In [8], dynamic sparse attentions are learned to focus on the informative regions. However, these methods are not designed for FSL, where test tasks comprise different sets of novel classes. Furthermore, these methods lack the task-adaptive mechanism, thus cannot generalize to the test images from unseen classes well. Different from these existing methods, in this paper, we design a task-aware mechanism through a meta filter learner, which can help the FSL model adapt to new tasks and discover task-related local parts.

3. Our Method

In this section, we first formulate the task of few-shot learning. Then we describe each modules of the proposed Task-aware Part Mining Network (TPMN) in detail. As shown in Figure 2, our TPMN consists of two modules. (1) The task-aware part filter module is responsible for discovering local parts through the task-aware part filters that are generated by a meta filter learner. (2) The part-aware metric module aims at calculating the final similarity score by weighted aggregation of the part-level similarities.

3.1. Problem Definition

The few-shot classification is conducted on a set of tasks $\mathcal{T}$ (also called episodes). The training set $\mathbb{D}_{train}$ is segmented into a set of tasks $\mathcal{T}_{train}$ to mimic the test setting, in the hope of acquiring the generalization ability across tasks. The testing set $\mathbb{D}_{test}$ is composed of testing tasks $\mathcal{T}_{test}$, and contains classes disjoint from the training set $\mathbb{D}_{train}$. Each few-shot task $\mathcal{T}$ consists of a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. Specifically, an N-way K-shot task means the model is trained over the support set $\mathcal{S}$ consisting of N classes with K samples per class, *i.e.*, $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{NK}$, where $y_i^s \in \{1, 2, \cdots, N\}$. The query set is composed of M samples per class, *i.e.*, $\mathcal{Q} = \{(x_i^q, y_i^q)\}_{i=1}^{MN}$. Our goal is to classify a query sample $x_i^q \in Q$ into one of the N support classes given a few labeled samples from $\mathcal{S}$.

3.2. Task-Aware Part Filter Module

Different from previous approaches that extract local features by manually grid-dividing [24, 25, 53, 9], we automatically explore the diverse object part regions to flexibly focus on the multi-scale local information, without any bounding boxes or part annotations. To achieve this goal, we design a part mining process to obtain part-aware features, and a meta filter learner to augment the part mining process with task-aware ability.

Part Mining Process. We introduce multiple Part Filters (PFs) to filter out the noisy background regions and retain the discriminative part regions with high objectness. Specifically, the feature map $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is extracted from feature extractor φ, where C, H and W denote the number of channels, height, and width, respectively. We assume that the object is composed of k meaningful local parts. Thus we design k PFs with each of them responsible for dealing with one specific part, denoted as $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k\}$. Each PF is parameterized by a 1×1 convolution kernel weight (the kernel bias is omitted to simplify notation), *i.e.*, the i-th part filter $\mathbf{p}_i \in \mathbb{R}^{1 \times 1 \times C}$. We apply PFs to convolve with feature map $\mathbf{x}$ and derive the activation map:

$$A_i = \mathbf{p}_i \odot \mathbf{x}, i = 1, 2, \cdots, k, \tag{1}$$

where $A_i \in \mathbb{R}^{H \times W}$ is the i-th part-aware activation map, and $\odot$ is the convolution operation. Then, the i-th part mask M_i can be generated by applying Sigmoid function σ on A_i, *i.e.*, $M_i = \sigma(A_i), i = 1, 2, \cdots, k$. M_i covers the region of the i-th object part by activating the pixels belonging to it.Then we can obtain k corresponding part-aware feature maps $\mathbf{F} = \{F_1, F_2, \cdots F_k\}$ w.r.t. the input feature map $\mathbf{x}$:

$$F_i = \mathbf{x} \otimes R(M_i), i = 1, 2, \cdots, k, \tag{2}$$

where R reshapes the mask M_i to be the same dimension as $\mathbf{x}$, *i.e.*, $R(M_i) \in \mathbb{R}^{H \times W \times 1}$, and $\otimes$ denotes the element-wise multiplication at each spatial location in every channel. $F_i \in \mathbb{R}^{H \times W \times C}$ is the resulted i-th part-aware feature map. Then, by applying global average pooling on each F_i in the spatial dimensions, we derive k part-aware features: $\Omega(x) = \{f_i\}_{i=1}^k$, where $f_i \in \mathbb{R}^C$. These part-aware features encapsulate the discriminative local part information, thus can be viewed as multi-scale and complementary representations of the given image.

Meta Filter Learner. The above generic PFs are shared across different tasks and are fixed after end-to-end training. However, the training tasks and testing tasks are sam-

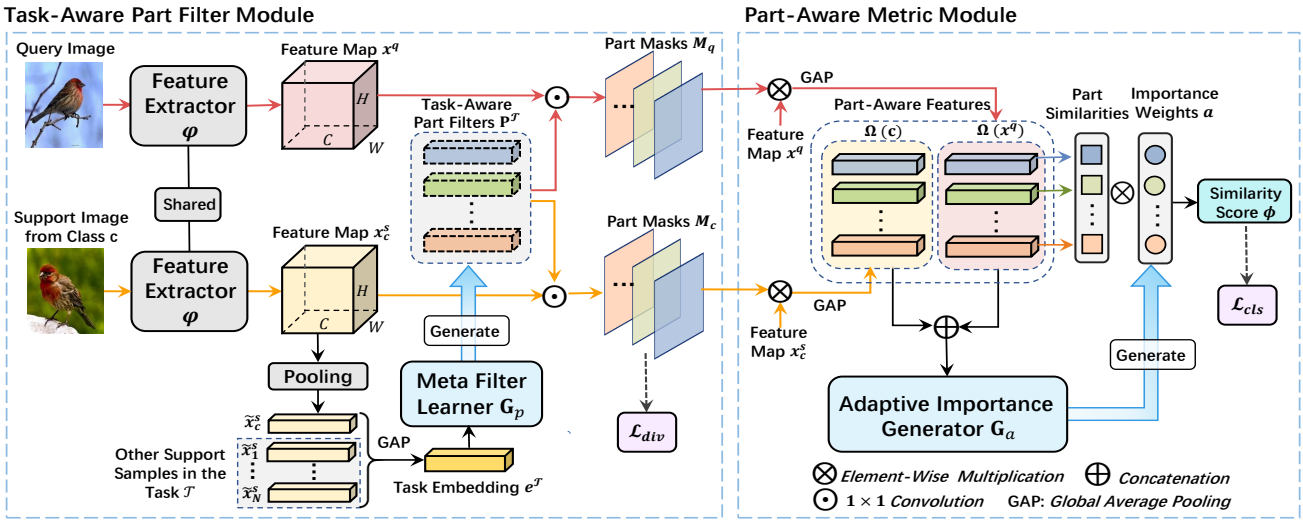

Figure 2. The architecture of TPMN (illustrated in the 1-shot setting): (1) The task-aware part filter module takes in the query and support images to derive their part features. The meta filter learner produces the task-aware part filters $\mathbf{P}^{\mathcal{T}}$ conditioned on the task embedding. $\mathbf{P}^{\mathcal{T}}$ are used to generate multiple part masks for each image. (2) The part-aware metric module computes the part similarities, which are then weighted by the importance weights produced by the adaptive importance generator for the final similarity score.

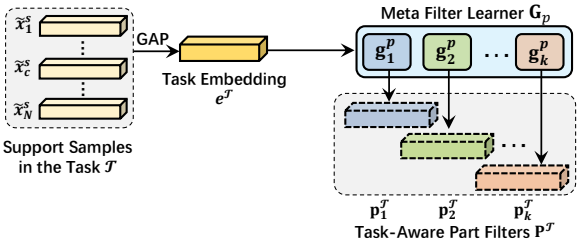

Figure 3. The illustration of the meta filter learner $\mathbf{G}_p$. $\mathbf{G}_p$ consists of a sequence of weight generators $\{g_i^p\}_{i=1}^k$, which generate the parameters of the corresponding task-aware part filter.

pled from different categories with large distribution differences. The generic PFs are not able to accommodate the need of these diverse tasks, as each task involves distinguishing a potentially unique set of classes. Therefore, we design a Meta Filter Learner (MFL) to augment the task-aware ability of the model. The MFL, denoted as $\mathbf{G}_p$, is responsible for adaptively generating the PFs conditioned on the categorical information of the specific task. Specifically, the MFL consists of a sequence of weight generators $\{\mathbf{g}_1^p, \mathbf{g}_2^p, \cdots, \mathbf{g}_k^p\}$ (see Figure 3), which can generate the parameters of the k corresponding task-aware PFs respectively (see **Supplementary Material** for details of the kernel bias generation). $\mathbf{G}_p$ accepts as input the task embedding $\mathbf{e}^{\mathcal{T}}$ to contextualize the categorical information involved in the current task $\mathcal{T}$. $\mathbf{e}^{\mathcal{T}}$ is defined as the mean vector of the feature vectors $\tilde{\mathbf{x}}^s$ of all the support instances in the task $\mathcal{T}$, *i.e.*, $\mathbf{e}^{\mathcal{T}} = \frac{1}{NK}\sum_{m=1}^{NK}\tilde{\mathbf{x}}_m^s$. Here, $\tilde{\mathbf{x}}^s \in \mathbb{R}^C$ is derived by applying global average pooling on the support feature map $\mathbf{x}^s$. $\mathbf{G}_p$ establishes the mapping from the task embedding $\mathbf{e}^{\mathcal{T}}$ to the parameter space of part filters. In this way, the formulation of PFs is conditioned on the contextualized information of the specific task, such that the PFs are aware of the task characteristics and can exploit the most task-related parts even for the unseen tasks. These task-aware PFs are denoted as $\mathbf{P}^{\mathcal{T}} = \{\mathbf{p}_1^{\mathcal{T}}, \mathbf{p}_2^{\mathcal{T}}, \cdots, \mathbf{p}_k^{\mathcal{T}}\}$, where $\mathbf{p}_i^{\mathcal{T}} \in \mathbb{R}^{1\times 1\times c}$ can be produced by the corresponding generators $\mathbf{g}_i^p$ in $\mathbf{G}_p$:

$$\mathbf{p}_i^{\mathcal{T}} = \mathbf{g}_i^p(\mathbf{e}^{\mathcal{T}}; \theta_i^p), i = 1, 2, \cdots, k, \tag{3}$$

where θ_i^p denotes the parameters of the $\mathbf{g}_i^p$. Then, we apply $\mathbf{P}^{\mathcal{T}}$ to Eq. (1) and Eq. (2) to produce the adapted part-aware features $\{f_i\}_{i=1}^k$. The MFL can be trained on large amounts of tasks to learn how to produce part filters that best fit the current task, in a meta-learning way. This is achieved by minimizing classification errors on query samples. The meta-learned MFL enables good generalization and fast adaptation on the completely new tasks in testing.

Diversified Local Parts. Without part-level supervision, it is likely the case that all of the part masks gather in the most discriminative regions and consequently produce identical part-aware features. To help the part-aware features target for distinct object part regions, we propose a part diversity loss motivated by [27, 51], which is formulated as:

$$\mathcal{L}_{div} = \sum_{i=1}^{k}\sum_{j=1, j\neq i}^{k}\frac{< f_i, f_j >}{\|f_i\|_2\|f_j\|_2}. \tag{4}$$

The intuition behind $\mathcal{L}_{div}$ is that, if the i-th and j-th part-aware features simultaneously give high activation responses in a similar area, then the $\mathcal{L}_{div}$ will have a large value. By minimizing $\mathcal{L}_{div}$, the part-aware features are prevented from having high similarities with each other.

3.3. Part-Aware Metric Module

After going through the task-aware part filter module, each instance is structurally reframed as a set of part-aware features: $\Omega(x) = \{f_i\}_{i=1}^k$. To predict the class of the query sample, we design a part-aware metric module to calculate and merge the part-level similarities based on the natural semantic correspondences between local parts. Then we use global similarities to perform k-NN classification.

Specifically, given the part-aware feature set of query sample x^q: $\Omega(x^q) = \{f_i^q\}_{i=1}^k$, we wish to obtain its similarities with all the categories $\{c\}_{c=1}^N$ in the support set, which can then be transformed as the predicted class probabilities. For the 1-shot setting, the support sample x_c^s from class c can directly represent its class: $\Omega(c) = \{f_i^s\}_{i=1}^k$. Notably, for the many-shots setting, we average the part-aware features of the support instances in the same class as the category part-aware features: $f_i^c = \frac{1}{K}\sum_{n=1}^K f_i^{s,n}, i = 1, 2, \cdots, k$, where $f_i^{s,n}$ denotes the n-th support sample in the class c. Then the category c can be represented as: $\Omega(c) = \{f_i^c\}_{i=1}^k$. For the convenience of the expression, we use $\Omega(c) = \{f_i^c\}_{i=1}^k$ as the category part-aware features for both the 1-shot and many-shot settings. Afterwards, the part-aware features obtained from the same task-aware PF form a natural semantic correspondence, as they are most likely to describe the same local parts. Thus we match f_i^q and f_i^c to form a part-aware feature pair.

The global image-level similarity between query x^q and category c can be determined by aggregating the part similarities. However, naively combining the local similarities with equal weights is suboptimal, as the contribution of different local parts varies a lot. Some parts may contain background noises and need to be suppressed. To achieve this goal, we design an adaptive importance generator $\mathbf{G}_a$ to assign the proper importance weights to the part-aware features. Specifically, the part-ware feature pairs are concatenated and fed into $\mathbf{G}_a$ to derive the importance weights:

$$a_i = \mathbf{G}_a(f_i^q \oplus f_i^c; \theta_a), i = 1, 2, 3, \cdots, k, \tag{5}$$

where $\oplus$ represents the concatenation, and a_i is the importance weight for the i-th part-aware feature pair, θ_a denotes the parameters of $\mathbf{G}_a$. Constrained by ground-truths, the end-to-end trained $\mathbf{G}_a$ could learn to assign higher weights to the parts that have a large contribution to the classification, e.g., the well-matched and discriminative parts. Denote the comparison module as $\mathbf{\Phi}$, the final global similarity is defined as the weighted sum of local similarities:

$$\mathbf{\Phi}(x^q, c) = \sum_{i=1}^k a_i (f_i^c \cdot f_i^q). \tag{6}$$

3.4. Training Objectives

Based on the global similarities computed by $\mathbf{\Phi}$, our network can compute the probability over class $c \in \{1, 2, \cdots, N\}$ for each query point x^q in the current task by a softmax function:

$$p(y = c|x^q) = \frac{exp(\mathbf{\Phi}(x^q, c))}{\sum_{c'=1}^N exp(\mathbf{\Phi}(x^q, c'))}, \tag{7}$$

The classification loss can be formulated as the negative log-probability:

$$\mathcal{L}_{cls} = -\frac{1}{|Q|} \sum_{(x^q, y^q) \in \mathcal{Q}} \log p(y = y^q | x^q), \tag{8}$$

As such, the final loss for our TPMN is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda_{div} \mathcal{L}_{div}. \tag{9}$$

where λ_{div} is the weight of $\mathcal{L}_{div}$. Please refer to the **Supplementary Material** for the discussion of the differences between TPMN and the relevent methods.

4. Experiments

In this section, we first introduce implementation details and datasets. Then, we show experimental results and some visualizations. Please refer to the **Supplementary Material** for more implementation details and results.

4.1. Implementation Details

For a fair comparison with previous works [52, 15, 53], we choose ResNet-12 as the backbone of the feature extractor φ, and remove the last global average pooling layer. Specifically, the input images, resized as $84 \times 84 \times 3$, are fed into φ to get the feature map x with the size of $5 \times 5 \times 640$. The weight generator $\mathbf{g}_i^p$ $(i = 1, 2, \cdots, k)$ in $\mathbf{G}_p$ is composed of 2 fully connected (FC) layers, each of which is followed by a ELU activation function. The adaptive importance generator $\mathbf{G}_a$ also consists of 2 FC layers. Before the meta-training, we apply a pre-training strategy for the backbone φ to accelerate the training process, as in [40, 52]. Then the model is trained in an episodic way. Each episode is comprised of an N-way K-shot task including 15 query samples for each class. We mainly experiment with 5-way 1-shot and 5-way 5-shot settings. The SGD optimizer is employed with the learning rate of 0.0001. We adopt image augmentations including horizontal flip, random crop and color jitter for training. The number of local parts is set as 15 and 20 for 1-shot and 5-shot settings on *mini*ImageNet, which is selected by episodic cross validation. The default value of λ_{div} is set as 0.1. During the testing stage, we report the average classification accuracy with $\pm 95\%$ confidence intervals in 1000 randomly sampled tasks.

4.2. Dataset Descriptions

We evaluate our model on four challenging datasets including *mini*ImageNet [48], *tiered*ImageNet [36], Fewshot-CIFAR100 (FC100) [30] and CIFAR-FS [4]. ***Mini*ImageNet** [48] has 100 categories with 600 samples

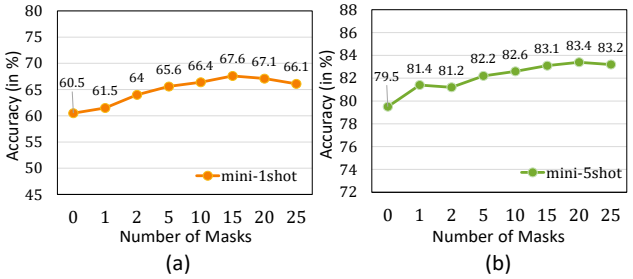

Figure 4. The effect of number of masks under (a) 5-way 1-shot setting and (b) 5-way 5-shot setting on ***mini*ImageNet**.

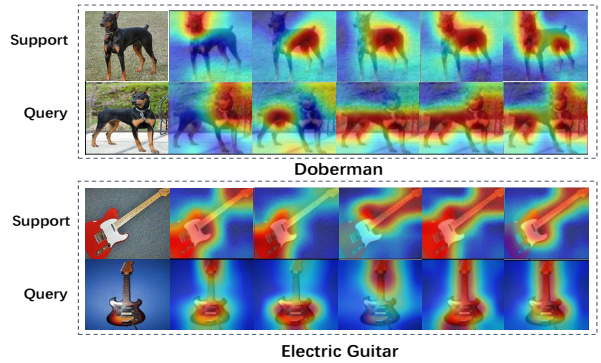

Figure 5. The visualization of the learned local parts (taking five local parts as examples) of two pairs of support and query images from the class *Doberman* and class *Electric Guitar*. Each pair of images are from the same class. We can observe the explicit semantic correspondences between the local parts.

per category chosen from the ILSVRC-2012 [39]. Following the split in [34], these categories are split into 64, 16, and 20 for training, validation and testing, respectively. ***Tiered*ImageNet** [36] is a larger subset of ILSVRC-12 [39], containing 608 classes that are split into 351 training classes, 97 validation classes and 160 test classes, as in [36]. **CIFAR-FS** [4] is built upon CIFAR100 [19] and contains 100 classes, with 600 samples per class. These classes are split into 64, 16 and 20 classes for training, validation, and testing, respectively. **FC100** [30] is also derived from CIFAR100 [19], which contains 100 classes grouped into 20 superclasses to minimize class overlap. These classes are split into 60, 20, and 20 for training, validation, and testing, respectively.

4.3. Comparison with Other Methods

Comparison with the baseline. We first compare our method with the baseline ProtoNet [44], which is based on the image-level representation. ProtoNet calculates the distances of the query sample with every class prototype (the mean representation of the support samples) as the class probabilities for the prediction. We re-implement the ProtoNet with the same pre-trained backbone and training strategy with our TPMN. As shown in Table 1, our TPMN significantly outperforms the ProtoNet that relies on the global image representations under both settings, achieving a whopping accuracy gain of **7.17%** in 5-way 1-shot setting and **3.97%** in 5-way 5-shot setting on *mini*ImageNet. This verifies the effectiveness of our method in discovering local parts and utilizing the local features. The accuracy improvements also show the superiority of the local part representation over the global representation, as local features are more fine-grained and transferable across tasks.

Comparison with methods based on local representations. We also compare our method with several metric-based baselines that are based on local embedding, including DN4 [24], ATL-Net [9], DC [25] and DeepEMD [53]. The results are summarized in Table 1. Our method outperforms all of these methods and achieves significant accuracy gains than the best local-based method (*i.e.*, DeepEMD). This is because previous methods simply adopt the patch representation divided from the feature map and suffer from large randomness. The patches could cover large noises and lose discriminative information contained in the irregular part regions like the *head*, *leg* and so on. However, our model can automatically exploit multiple local parts by part filters, and percept the most task-related local features for the current task with a task-aware mechanism.

Comparison with the state-of-the-arts. We compare TPMN with some state-of-the-art methods. (1) **Results on *mini*ImageNet and *tiered*ImageNet** (see Table 1 (a)). The compared state-of-the-art methods are divided into three groups: gradient-based, generation-based, and metric-based. Our TPMN achieves the new state-of-the-art performance on both benchmarks under all settings, which strongly proves the effectiveness of our method. Compared with the best generation-based method (DTN), TPMN achieves a large margin of **4.19%** in 1-shot setting, and **5.53%** in 5-shot setting. The previous generation-based methods generally produce the parameters of linear classifiers and convolution layers, while our method focuses on the generation of the more generalizable task-aware part filters. Compared with the metric-based methods, our method also has a clear lead. This is because our method claims better transferability by utilizing the task-aware local parts. (2) Results on **FC100 and CIFAR-FS** (see Table 1 (b) and Table 1 (c)). The proposed method also achieves superior performance in all the tasks on FC100 and CIFAR-FS. In particular, our results outperform the state-of-the-art performance by a significant margin of 1.6% in the 5-way 1-shot task on CIFAR-FS, which further demonstrates the effectiveness of our method in exploiting task-aware local parts.

4.4. Ablation Study

In this section, we perform detailed ablation studies to evaluate the effect of each design.

Analysis of Model Components. We perform detailed analysis of model components of TPMN on *mini*ImageNet, as shown in Table 2. To investigate the contribution of the task-aware part filters (TAPFs), we compare with our re-

Method	Backbone	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
Gradient-based					
MAML [12]	ConvNet	48.70 ± 0.84	55.31 ± 0.73	51.67 ± 1.81	70.30 ± 1.75
MTL [45]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80	65.62 ± 1.80	80.61 ± 0.90
MetaOptNet [21]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.71	81.56 ± 0.63
E3BM [28]	ResNet-12	63.8 ± 0.4	80.1 ± 0.3	71.2 ± 0.4	85.3 ± 0.3
Generation-based					
TADAM [30]	ResNet-12	58.50 ± 0.30	76.60 ± 0.38	-	-
Dynamic [13]	ConvNet	56.20 ± 0.86	73.00 ± 0.64	-	-
LEO [40]	WRN-28	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
DTN [6]	ResNet-12	63.45 ± 0.86	77.91 ± 0.62	-	-
Metric-based					
MatchingNet [48]	ResNet-12	43.56 ± 0.84	55.31 ± 0.73	-	-
RelationNet [46]	ResNet-12	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.32 ± 0.78
CAN [15]	ResNet-12	63.85 ± 0.48	79.44 ± 0.34	69.89 ± 0.51	84.23 ± 0.37
DN4§ [24]	ConvNet	51.24 ± 0.74	71.02 ± 0.64	-	-
ATL-Net§ [9]	ConvNet	54.30 ± 0.76	73.22 ± 0.63	-	-
Distill [47]	ResNet-12	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
DSN [42]	ResNet-12	62.64 ± 0.66	78.83 ± 0.45	66.22 ± 0.75	82.79 ± 0.48
FEAT [52]	ResNet-12	66.78 ± 0.20	82.05 ± 0.14	-	-
TRAML [22]	ResNet-12	67.10 ± 0.52	79.54 ± 0.60	-	-
DC§ [25]	ResNet-12	62.53 ± 0.19	78.95 ± 0.13	-	-
DeepEMD§ [53]	ResNet-12	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
ProtoNet‡ [44]	ResNet-12	60.47 ± 0.62	79.47 ± 0.43	70.46 ± 0.69	83.78 ± 0.65
TPMN (ours)	ResNet-12	**67.64** ± 0.63	**83.44** ± 0.43	**72.24** ± 0.70	**86.55** ± 0.63

(a) Results on ***mini*ImageNet** and ***tiered*ImageNet** datasets.

Method	Backbone	Fewshot-CIFAR100	
		1-shot	5-shot
TADAM [30]	ResNet-12	40.10 ± 0.40	56.10 ± 0.40
MetaOptNet [21]	ResNet-12	41.10 ± 0.60	55.50 ± 0.60
MatchingNet [48]	ResNet-12	43.88 ± 0.75	57.05 ± 0.71
MTL [45]	ResNet-12	45.10 ± 1.8	57.60 ± 0.9
Distill [47]	ResNet-12	44.60 ± 0.7	60.90 ± 0.6
Centroid [1]	ResNet-18	45.83 ± 0.48	59.74 ± 0.56
DC§ [25]	ResNet-12	42.04 ± 0.17	57.05 ± 0.16
DeepEMD§ [53]	ResNet-12	46.47 ± 0.26	63.22 ± 0.71
ProtoNet‡ [44]	ResNet-12	42.66 ± 0.76	58.92 ± 0.76
TPMN (ours)	ResNet-12	**46.93** ± 0.71	**63.26** ± 0.74

(b) Results on **Fewshot-CIFAR100** dataset.

Method	Backbone	CIFAR-FS	
		1-shot	5-shot
Shot-Free [35]	ResNet-12	69.2 ± n/a	84.7 ± n/a
MetaOptNet [21]	ResNet-12	72.0 ± 0.7	84.2 ± 0.5
MABAS [17]	ResNet-12	73.5 ± 0.8	85.7 ± 0.7
Distill [47]	ResNet-12	73.9 ± 0.8	86.9 ± 0.5
DSN [42]	ResNet-12	72.3 ± 0.8	85.1 ± 0.6
ProtoNet‡ [44]	ResNet-12	70.3 ± 0.7	83.5 ± 0.5
TPMN (ours)	ResNet-12	**75.5** ± 0.9	**87.2** ± 0.6

(c) Results on **CIFAR-FS** dataset.

Table 1. Comparison of our method with the state-of-the-art methods on (a) ***mini*ImageNet**, ***tiered*ImageNet**, (b) **Fewshot-CIFAR100** and (c) **CIFAR-FS**. The **bold font** indicates the highest result. ‡ means the results are from our re-implemented version, and § denotes the methods based on local representations.

Method	1-shot	5-shot
ProtoNet‡	60.47	79.47
ProtoNet‡+PF	64.11	80.75
ProtoNet‡+TAPF	65.94	81.47
ProtoNet‡+TAPF+$\mathcal{L}_{div}$	66.53	82.84
ProtoNet‡+TAPF+$\mathbf{G}_a$	66.83	82.39
ProtoNet‡+TAPF+$\mathbf{G}_a$+$\mathcal{L}_{div}$(TPMN)	**67.64**	**83.44**

Table 2. Ablation results on ***mini*ImageNet** in 5-way 1-shot and 5-way 5-shot settings.

Method	1-shot	5-shot
cosmax [7]	43.06 ± 1.01	64.38 ± 0.86
ProtoNet‡ [44]	47.51 ± 0.72	67.96 ± 0.70
Diverse 20 [10]	-	66.17 ± 0.55
centroid [1]	46.85 ± 0.75	70.37 ± 1.02
FEAT‡ [52]	50.67 ± 0.78	71.08 ± 0.73
TPMN(ours)	**52.83** ± 0.65	**72.69** ± 0.52

Table 3. Cross-domain results from ***mini*ImageNet** to **CUB** in 1-shot and 5-shot settings. ‡ denotes our implementation.

implemented ProtoNet and the generic part filters (PFs) that are end-to-end learned and shared across tasks, using the same number of masks. Then, we test the performance improvement of $\mathcal{L}_{div}$ and the adaptive importance generator $\mathbf{G}_a$. The complete version of TPMN gives the highest results in both settings. The results are analyzed as follows: (1) The introduction of TAPFs achieves remarkable performance gains compared with ProtoNet, *e.g.*, **5.47%** in 1-shot setting. The improvements can be mainly ascribed to the strong ability of TAPFs to discover complementary local parts. Also, the TAPFs significantly outperform the generic PFs, with a lead of 1.83% in 1-shot setting. This justifies the superiority of our task-aware mechanism. Compared with generic PFs, TAPFs can adapt to arbitrary tasks and produce more transferable and task-related part features. (2) With the utilization of $\mathcal{L}_{div}$, further improvements can be observed, *e.g.*, 1.37% in 5-shot setting. $\mathcal{L}_{div}$ prevents the filters from focusing on similar local parts. The diversified local regions help the model to form a more comprehensive understanding of the object. (3) The addition of $\mathbf{G}_a$ also contributes to a certain performance lift compared with the combination of ProtoNet and TAPF, leading by 0.92% in 5-shot setting. This proves that the $\mathbf{G}_a$ is productive in discerning the discriminative parts and less-discriminative ones. Thus $\mathbf{G}_a$ can assign proper part weights to promote a more effective global metric.

Analysis of the Number of Local Parts. We study the influence of the number of local parts (denoted as N_f) on *mini*ImageNet (see Figure 4). In 1-shot setting, the best performance is achieved when $N_f = 15$. With the growth of N_f, the accuracy presents a rising trend, since more part regions can bring in more complementary semantic information of the object. However, after reaching the peak value at $N_f = 15$, the accuracy falls as N_f increases. This is because the available few samples cannot support the learning of the massive model parameters brought by increasing

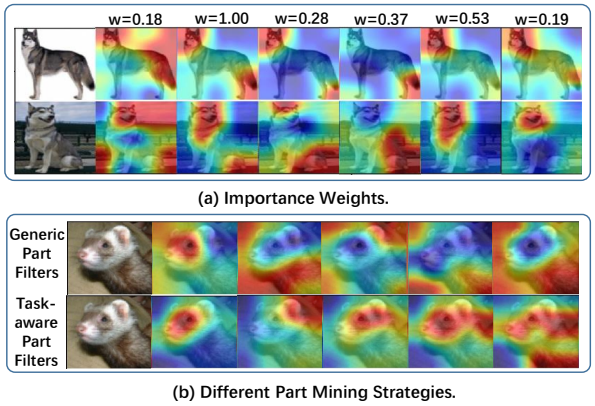

Figure 6. (a) The visualization of the local parts and their normalized importance weights. Larger weights are assigned to the more discriminative parts. (b) The local parts obtained from different part mining strategies on the unseen tasks. The task-aware PFs have better generalization and localization abilities than the generic PFs that are end-to-end learned and shared across tasks.

masks. Too many PFs may introduce redundant semantic information and even background noises. A similar performance variation trend can be observed in the 5-shot setting. The difference is that the best performance is achieved at $N_f = 20$. Overall speaking, the accuracy changes relatively smoothly under different N_f, which indicates the robustness of our model.

4.5. Cross-Domain Transfer Experiments

Following the experimental setups in [7], we perform cross-domain transfer experiments where the model is meta-trained on *mini*ImageNet dataset, but is meta-tested on CUB dataset [49]. This experimental setting allows for a larger domain gap and better evaluation of the knowledge transfer ability of different methods. As shown in Table 3, our method demonstrates the superiority over other methods and shows absolute accuracy gains of 2.16% in 1-shot setting, and 1.61% in 5-shot setting over the best method (*i.e.*, FEAT [52]). This indicates that our method effectively mines the object part regions that provide more transferable local information across different domains. Furthermore, the task-aware mechanism in meta filter learner allows for the customization of the part mining process for any task. Therefore, our model can overcome the domain gap and generalize to the novel classes well.

4.6. Visualizations

Visualization of Part Correspondence. To qualitatively evaluate the task-aware part filters, we visualize several groups of part masks of the query and support images on *mini*ImageNet and *tiered*ImageNet. The images in each group are from the same category (see Figure 5). As we can see, clear semantic correspondences are established between the pair of part masks obtained from the same task-aware PF. For example, in the category *Doberman*, the part *head*, *hip* and *lower limbs* of the query image can accurately match with the corresponding part masks of the support image, even though the objects in the two images are in the opposite direction. This proves the efficiency of our PFs. After meta-training on numerous tasks consisting of diverse images, each PF can capture a specific semantic pattern well, so the resulted part alignments are robust to the view and scale variations. Another interesting fact is that our PFs can discover not only the small regions like the part *head*, but also the large regions like the *the upper part of the body*. The multi-scale information can further strengthen the discriminative power and robustness of our model.

Visualization of Part Importance Weights. To vividly present the working mechanism of adaptive importance generator $\mathbf{G}_a$, we visualize the part masks with their normalized importance weights assigned by $\mathbf{G}_a$, in the pair of query and support images belonging to the same category. As shown in Figure 6 (a), some well-matched and discriminative parts are highlighted, occupying larger weights (*i.e.*, the second, fourth and fifth parts), while some parts that are not well-aligned or contain large background noises are assigned smaller weights (*i.e.*, the first and the last parts).

Comparison between different part mining strategies. We compare the part masks in the unseen task learned by the generic PFs and task-aware PFs in a more intuitive way. As shown in Figure 6 (b), the generic PFs can hardly attend to the objects, with most of the masked regions full of irrelevant backgrounds. Because of the absence of an explicit mechanism for feature adaptation, the generic PFs cannot generate discriminative part regions for the images from novel classes. By contrast, the task-aware part filters, be equipped with the task-adaptive ability brought by the meta filter learner, have a good performance in mining diverse and high-quality local parts for the unseen classes.

5. Conclusion

In this paper, we propose a Task-aware Part Mining Network for FSL. We automatically explore object parts in the metric-based model in a meta-learning way. A meta filter learner is proposed to generate task-aware part filters, which can adapt to any individual task and mine discriminative local parts for FSL. Experiments show the effectiveness.

6. Acknowledgement

This work was partially supported by the National Key Research and Development Program under Grant No. 2018YFB0804204, National Defense Basic Scientific Research Program (JCKY2020903B002), Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC02050500), National Nature Science Foundation of China (Grant 62022078, 62021001), Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 202000019, and Youth Innovation Promotion Association CAS 2018166.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *European Conference on Computer Vision*, pages 18–35. Springer, 2020.

[2] Maria-Luiza Antonie, Osmar R Zaiane, and Alexandru Coman. Application of data mining techniques for medical image classification. In *Proceedings of the Second International Conference on Multimedia Data Mining*, pages 94–101, 2001.

[3] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *International Conference on Learning Representations*, 2018.

[4] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2018.

[5] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, 2016.

[6] Mengting Chen, Yuxin Fang, Xinggang Wang, Heng Luo, Yifeng Geng, Xinyu Zhang, Chang Huang, Wenyu Liu, and Bo Wang. Diversity transfer network for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10559–10566, 2020.

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

[8] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6599–6608, 2019.

[9] Chuanqi Dong, Wenbin Li, Jing Huo, Zheng Gu, and Yang Gao. Learning task-aware local representations for few-shot learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2020.

[10] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

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

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, pages 4003–4014, 2019.

[16] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[17] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *European Conference on Computer Vision*, 2020.

[18] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[21] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[22] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12576–12584, 2020.

[23] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2019.

[24] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[25] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9258–9267, 2019.

[26] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision*, pages 438–455, 2020.

[27] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, June 2019.

[28] Yaoyao Liu, Bernt Schiele, and Qianru Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *European Conference on Computer Vision*, pages 404–421, 2020.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[30] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018.

[31] Yuxin Peng, Xiangteng He, and Junjie Zhao. Object-part attention model for fine-grained image classification. *IEEE Transactions on Image Processing*, 27(3):1487–1500, 2017.

[32] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[33] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[35] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 331–339, 2019.

[36] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

[38] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems*, pages 7959–7970, 2019.

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

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

[41] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018.

[42] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4136–4145, 2020.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[45] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[46] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[47] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, pages 266–282, 2020.

[48] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[50] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13390–13399, 2020.

[51] Wenfei Yang, Tianzhu Zhang, Zhendong Mao, Yongdong Zhang, Qi Tian, and Feng Wu. Multi-scale structure-aware network for weakly supervised temporal action detection. *IEEE Transactions on Image Processing*, 30:5848–5861, 2021.

[52] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[53] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020.

[54] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Learning multi-task correlation particle filters for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):365–378, 2019.

[55] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Robust structural sparse tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):473–486, 2019.

[56] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.