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

In the supplementary material, we first introduce more details of our framework. Then, we introduce more implementation details, including the details of the datasets, the pre-training strategy, and the hyper-parameters on the four datasets for reproducing the results. Afterwards, we show more visualization results on the part masks learned by TPMN. Finally, we discuss the differences between the proposed TPMN and the relevant methods. The overall architecture of the proposed model is presented in Figure 1.

1. More Details of Our Framework

In this section, we introduce the complete architecture of the meta filter learner and task-aware part filters (see Figure 2). As a 1×1 convolution kernel, the parameter set of the part filter is comprised of the kernel weight parameters and **kernel bias** parameters. The main text only presents the formulation of the kernel weight, for the simplicity of the notation. To generate the bias parameter of task-aware part filter, the meta filter learner $\mathbf{G}_p$ includes an additional bias generator $\mathbf{g}^b$ apart from the weight generators $\{\mathbf{g}_i^p\}_{i=1}^k$ that are introduced in the main text, i.e., $\mathbf{G}_p = \{\{\mathbf{g}_i^p\}_{i=1}^k, \mathbf{g}^b\}$. The bias generator $\mathbf{g}^b$ takes the task embedding $e^{\mathcal{T}}$ as input, and predicts the bias parameters of part filters for the current task $\mathcal{T}$. Denote the parameters of $\mathbf{g}^b$ as θ^b, the bias parameter set $b^{\mathcal{T}}$ of task-aware part filters can be derived as

$$b^{\mathcal{T}} = \mathbf{g}^b(e^{\mathcal{T}}; \theta^b), \tag{1}$$

where $b^{\mathcal{T}} \in \mathbb{R}^k$. The i-th dimension of the bias parameter set $b^{\mathcal{T}}$ is the bias parameter of the i-th task-aware part filter, $i = 1, 2, \cdots, k$. In this way, the meta filter learner $\mathbf{G}_p$ can produce task-aware part filters that can discover multiple task-specific local parts. Similar with the architecture of the weight generators, the bias generator $\mathbf{g}^b$ consists of two fully connected layers, with the first layer followed by a ELU activation layer.

2. More Implementation Details

In this section, we provide more implementation details of our proposed TPMN. Specifically, we present more details of the dataset, the pre-training strategy, and hyper-parameters used to reproduce results.

2.1. More Dataset Details.

To demonstrate the effectiveness of the proposed model, we conduct experiments on four standard datasets including: *mini*ImageNet [13], *tiered*ImageNet [9], CIFAR-FS [1], and Fewshot-CIFAR100 (FC100) [7]. Here, we introduce more details of these four datasets. As shown in Table 1, we list details for the number of images, the number of classes, image resolution and train/val/test splits.

***Mini*ImageNet** [13] is a widely used benchmark dataset for FSL. There are 100 categories with 600 samples per category chosen from the ILSVRC-2012 [11]. The size of each image is 84×84. Following the split in [8], these categories are randomly split into 64 training classes, 16 validation classes and 20 test classes.

***Tiered*ImageNet** [9] is a larger subset of ILSVRC-12, containing 608 classes from 34 super-classes and 779,165 images in total. We split it into 20/351 training classes, 6/97 validation classes and 8/160 test classes, as in [9]. The splits are set according to the super-classes to enlarge the domain difference between the training and testing sets. All images of the two datasets are resized to 84×84.

CIFAR-FS [1] is built upon CIFAR100 [6] and contains 100 classes. These classes are split into 64, 16 and 20 classes for training, validation, and testing, respectively, following the criteria of previous work in [1]. There are 600 samples per class and the resolution of every image is 32×32.

FC100 [7] is also derived from CIFAR100 [6], which contains 100 classes grouped into 20 super-classes. We follow the split division proposed in [7], where base, validation and novel splits contain 60, 20, 20 classes belonging to 12, 5, and 5 super-classes, respectively. The image resolution and the number of samples per class are the same as that of CIFAR-FS dataset. It is worth noting that, compared with CIFAR-FS, the class overlap in the partition of training set, validation set and test set of FC100 is lower, because the classes are selected from different superclasses. Therefore, FC100 is a more challenging dataset than CIFAR-FS.

*Corresponding Author

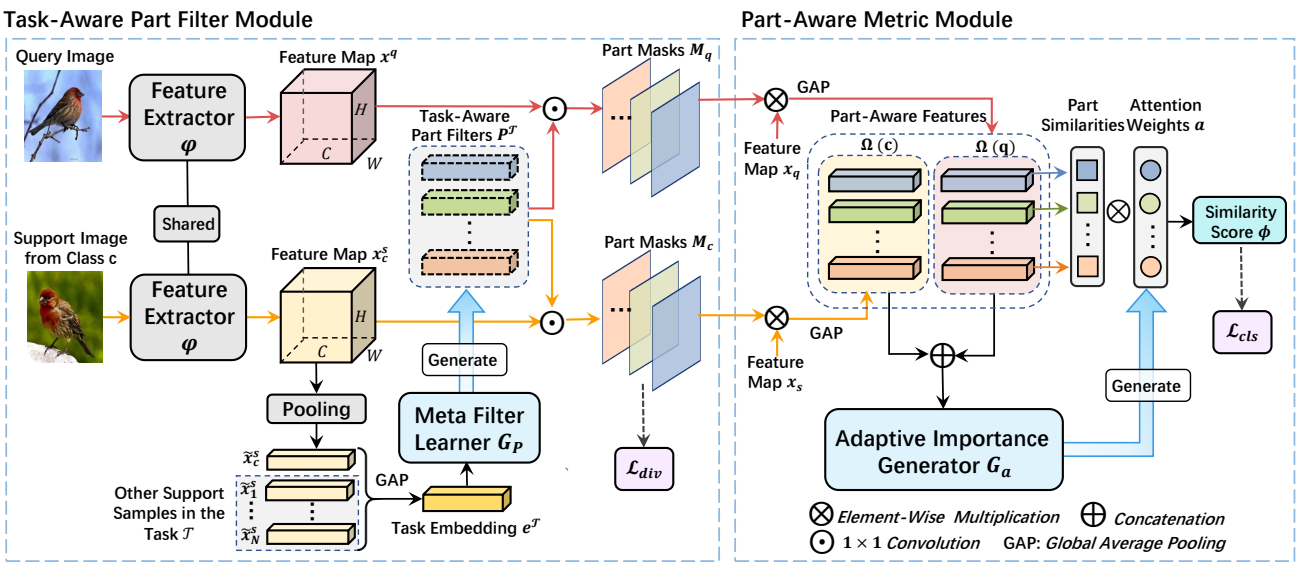

Figure 1. The architecture of our method (illustrated in the 1-shot setting): (1) The task-aware part filter module takes in the query and support images to extract their feature maps. Then the meta filter learner $\mathbf{G}_p$ produces the task-aware part filters $\mathbf{P}^{\mathcal{T}}$ conditioned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

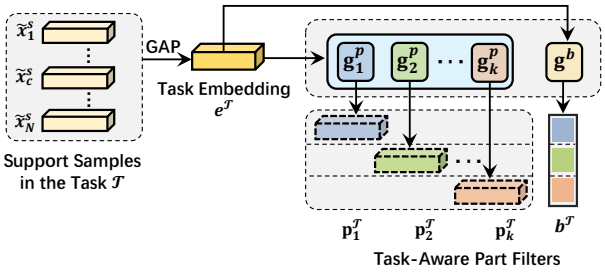

Figure 2. The illustration of the meta filter learner $\mathbf{G}_p$. $\mathbf{G}_p$ consists of a sequence of weight generators $\{\mathbf{g}_i^p\}_{i=1}^k$ and a bias generator $\mathbf{g}^b$, which generate the kernel weight parameters and kernel bias parameters of the corresponding task-aware part filters, respectively.

Dataset	Images	Classes	Train-val-test	Resolution
*Mini*ImageNet	60000	100	64/16/20	84×84
*Tiered*ImageNet	779165	608	351/97/160	84×84
CIFAR-FS	60000	100	64/16/20	32×32
FC100	60000	100	60/20/20	32×32

Table 1. The details of benchmark datasets used in FSL.

2.2. Pre-training Strategy.

Instead of optimizing from scratch, we apply a pre-training strategy for the backbone φ to accelerate the training process, as suggested in [12, 15]. We adopt ResNet-12 as the backbone network for feature extraction. During the pre-training, the global average pooling layer of the backbone network is preserved, and is appended with a softmax layer. Then the backbone is trained by all the SEEN classes (e.g., 64 classes in the *mini*ImageNet) in the training set with a cross-entropy loss. The feature embeddings of the penultimate layer of the backbone are utilized to evaluate the classification performance. The evaluation is conducted on multiple randomly sampled 1-shot tasks from the validation set. Then the best pre-trained model is selected and is used to initialize the feature extractor backbone φ in TPMN.

2.3. Hyper-parameters on Different Datasets.

We introduce more detailed hyper-parameters on different datasets, for the convenience of reproducing the results. The number of the part masks, denoted as N_f, is set as different values on different datasets.

On the *mini*ImageNet Dataset:

- The evaluation frequency is 50 episodes.
- Base learning rate is 0.0001.
- The weight decay in SGD optimizer is 5e-4.
- The loss coefficient λ_{div} of part diversity loss $\mathcal{L}_{div}$ is set as 0.1.
- The number of the part masks N_f is set as 15 in 1-shot setting.
- The number of the part masks N_f is set as 20 in 5-shot setting.

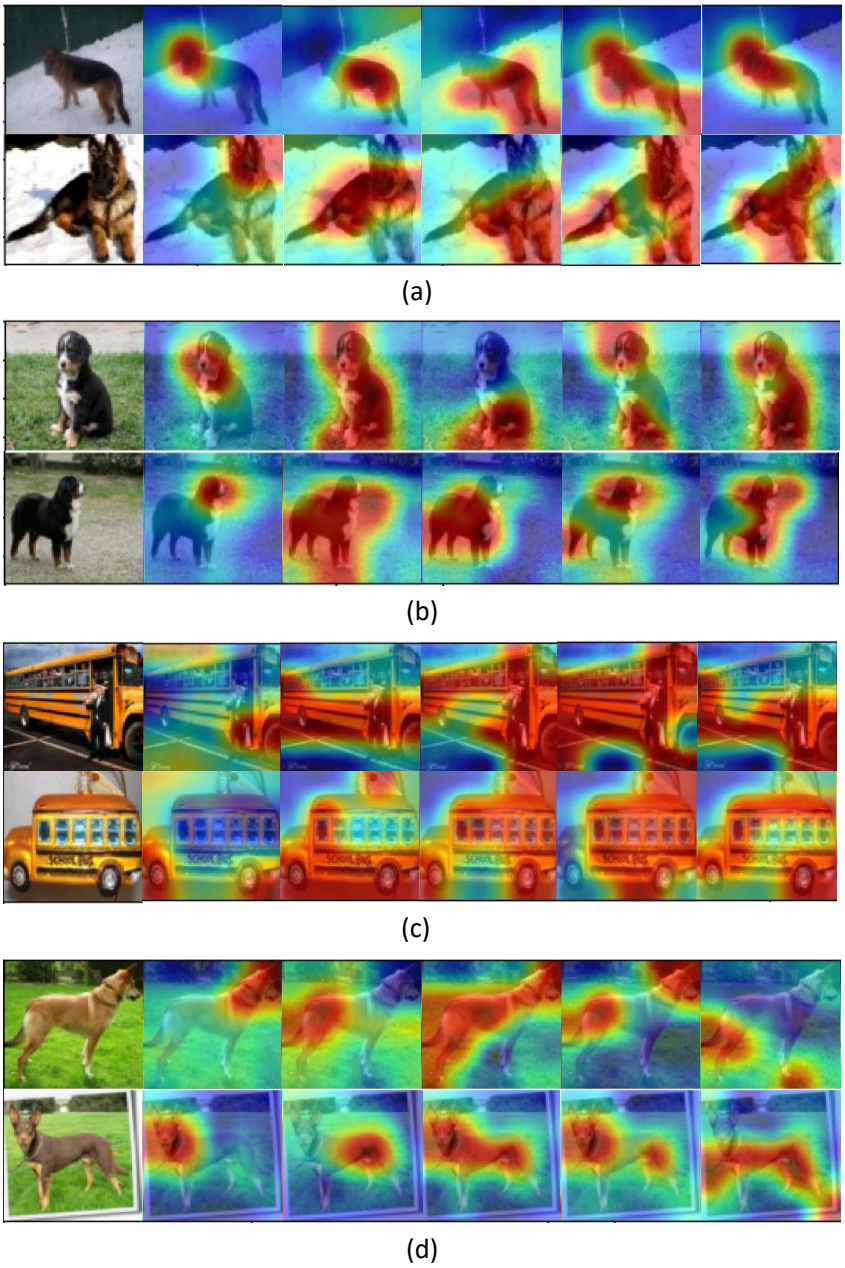

Figure 3. The visualization of the learned local parts (taking five local parts as examples) of two pairs of support and query images. Each pair of images is from the same class. We can observe the explicit semantic correspondence between the local parts.

- The number of training epoch is 300.

On the *tiered*ImageNet Dataset:

- The evaluation frequency is 50 episodes.
- Base learning rate is 0.0001.
- The weight decay in SGD optimizer is 5e-4.
- The loss coefficient λ_{div} of part diversity loss $\mathcal{L}_{div}$ is set as 0.1.
- The number of the part masks N_f is set as 10 in 1-shot setting.
- The number of the part masks N_f is set as 17 in 5-shot setting.
- The number of training epoch is 300.

On the FC100 Dataset:

- The evaluation frequency is 25 episodes.

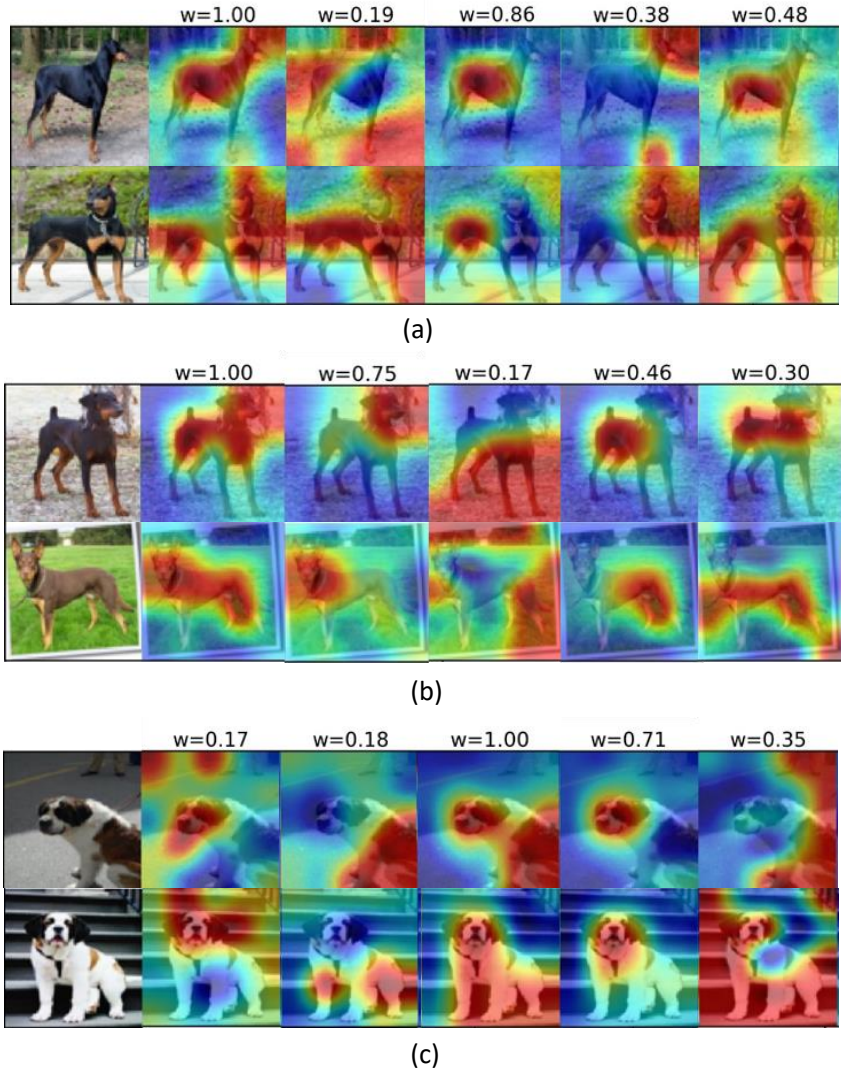

Figure 4. The visualization of the local parts and their normalized importance weights (normalizing the maximum importance weight to 1) in a pair of query and support images. Larger weights are assigned to the more discriminative parts.

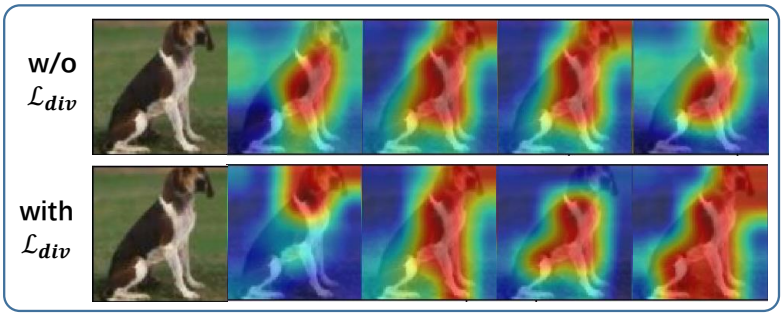

Figure 5. The part visualizations with $\mathcal{L}_{div}$ and without $\mathcal{L}_{div}$.

- Base learning rate is 0.00017.
- The weight decay in SGD optimizer is 5e-4.
- The loss coefficient λ_{div} of part diversity loss $\mathcal{L}_{div}$ is set as 2.4.
- The number of the part masks N_f is set as 10 in 1-shot

N_T	50	500	1000	2000	3000	6000
TPMN	65.20	66.10	66.42	66.93	67.22	67.64

Table 2. The effect of number of tasks (N_T) on *mini*ImageNet in 5w-1s setting. We utilize the pre-trained ResNet-12.

setting.

- The number of the part masks N_f is set as 10 in 5-shot setting.
- The number of training epoch is 300.

On the CIFAR-FS Dataset:

- The evaluation frequency is 25 episodes.
- Base learning rate is 0.0002.
- The weight decay in SGD optimizer is 5e-4.
- The loss coefficient λ_{div} of part diversity loss $\mathcal{L}_{div}$ is set as 1.0.
- The number of the part masks N_f is set as 13 in 1-shot setting.
- The number of the part masks N_f is set as 15 in 5-shot setting.
- The number of training epoch is 300.

3. The Effects of the Number of Tasks

We study the effects of the number of training tasks (denoted as N_T) on *mini*ImageNet in 5-way 1-shot setting. As shown in Table 2, as more tasks are trained, the performance is gradually improving, with the accuracy increasing from 65.20 to 67.64. This verifies that meta filter learner is improved during the training on large amounts of tasks. The meta filter learner learns how to produce part filters that best fit the needs of the current task by meta-learning on the numerous training tasks.

4. More Visualization Results

Here, we give more visualization results of the learned part masks on *mini*ImageNet and *tiered*ImageNet. First, we visualize the part correspondences between the part masks from the query and support images. Four groups of results are presented, and the images in each group are from the same category. As shown in Figure 3, there are clear semantic correspondences between the part masks obtained from the same task-aware part filter, which justifies the effectiveness of our task-aware part filters. Also, we provide more visualization results of the part importance weights to testify the effectiveness of the adaptive importance generator $\mathbf{G}_a$. We use the min-max normalization to normalize the maximum importance weights to 1. The normalized importance weights along with the matched part masks of the query and support image are shown in Figure 4. It can be observed that the discriminative and well-matched parts are assigned with larger importance weights. On the contrary, the local parts that contain too many background noises enjoy smaller importance weights. Besides, we provide the comparisons of part visualizations with the part diversity loss $\mathcal{L}_{div}$ and the visualizations without $\mathcal{L}_{div}$. As shown in Figure 5, without $\mathcal{L}_{div}$, all parts gather in similar regions, ignoring other fine-grained local regions that also provide discriminative clues. However, $\mathcal{L}_{div}$ can help discover diverse and complementary parts, which improves the transfer and generalization abilities of our method.

5. Discussion

In this section, we discuss the differences between TPMN and several relevant methods including ATL-Net [2], TAFE-Net [14], CNAPs [10], and some attention-based FSL methods including CAN [5], STANet [4] and AWGIM [3].

(1) ATL-Net [2] proposes an episodic attention mechanism to weight different key local patches based on the relationship between local patches. Their so-called "task-aware local representations" are in fact the shared local patches weighted by different coefficients. This is similar to our design of assigning adaptive weights to the local representations. However, our method further takes into consideration the categorical information to produce task-related local representations that dynamically cover multi-scale regions for different tasks. Compared with the fixed local patches in ATL-Net that may contain considerable background noises, the local representations learned by our task-aware part-filters are more flexible to satisfy the needs of different tasks, even the tasks with unseen classes.

(2) TAFE-Net [14] and CNAPs [10] utilize a meta learner to generate task-specific parameters for convolution layers and linear classification layers in the network, which is similar to our idea of generating parameters in a meta-learning way. However, both of them attempt to learn a global image-level representation, which loses considerable local information. Unlike these methods, our method mainly focuses on discovering task-aware local representation that enjoys favorable transferability and consistency across different tasks.

(3) CAN [5] proposes a cross-attention module to localize the regions of the target object for few-shot learning. It utilizes the correlation map between each pair of query and support samples to generate cross attention maps. CAN learns single object-centric representation, while TPMN learns the task-specific, complementary and transferable lo-

cal representations that can generalize better to the unseen tasks. Similar to CAN, STANet [4] only localizes a single object region by spatial attention. Also, STANet lacks the task-aware ability to adapt the model to unseen tasks, while TPMN can customize the task-aware part filters for novel tasks. AWGIM [3] uses multi-head attention to model the relations between samples within one task, while TPMN focuses on local part mining to obtain transferable local features.

(4) Compared to the above methods, our TPMN utilizes the meta-learning strategy to generate the part filers that can discover discriminative local parts for the unseen tasks. By taking advantage of the automatically-mined and task-aware local representations, our method acquires fast adaptation and good generalization abilities across tasks.

References

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

[2] 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.

[3] Guo et al. Attentive weights generation for few shot learning via information maximization. In *CVPR*, 2020.

[4] Yan et al. A dual attention network with semantic embedding for few-shot learning. In *AAAI*, 2019.

[5] 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.

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

[7] 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*, volume 31, pages 721–731, 2018.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Xin Wang, Fisher Yu, Ruth Wang, Trevor Darrell, and Joseph E Gonzalez. Tafe-net: Task-aware feature embeddings for low shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[15] 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.