A Multi-Mode Modulator for Multi-Domain Few-Shot Classification

Yanbin Liu[1,2], Juho Lee[3,4], Linchao Zhu[2], Ling Chen[2], Humphrey Shi[5], Yi Yang[2*]
[1]Baidu Research, [2]AAII, University of Technology Sydney, [3]KAIST, [4]AITRICS
[5]University of Oregon & Picsart AI Research (PAIR)
{csyanbin, shihonghui3}@gmail.com, juholee@kaist.ac.kr
{linchao.zhu, ling.chen, yi.yang}@uts.edu.au

Abstract

*Most existing few-shot classification methods only consider generalization on one dataset (i.e., **single-domain**), failing to transfer across various seen and unseen domains. In this paper, we consider the more realistic **multi-domain** few-shot classification problem to investigate the cross-domain generalization. Two challenges exist in this new setting: (1) how to efficiently generate multi-domain feature representation, and (2) how to explore domain correlations for better cross-domain generalization. We propose a parameter-efficient multi-mode modulator to address both challenges. First, the modulator is designed to maintain multiple modulation parameters (one for each domain) in a single network, thus achieving single-network multi-domain representation. Given a particular domain, domain-aware features can be efficiently generated with the well-devised separative selection module and cooperative query module. Second, we further divide the modulation parameters into the domain-specific set and the domain-cooperative set to explore the intra-domain information and inter-domain correlations, respectively. The intra-domain information describes each domain independently to prevent negative interference. The inter-domain correlations guide information sharing among relevant domains to enrich their own representation. Moreover, unseen domains can utilize the correlations to obtain an adaptive combination of seen domains for extrapolation. We demonstrate that the proposed multi-mode modulator achieves state-of-the-art results on the challenging META-DATASET benchmark, especially for unseen test domains.*

1. Introduction

Few-shot classification aims to train a model that can generalize on unseen novel classes with only few labeled examples in each novel class. Recent progress has been made by the meta-learning paradigm: instead of learning about any training class in particular, few-shot algorithms exploit the training classes to learn to recognize new classes with few examples. Excellent results are achieved on common benchmarks (*e.g.*, Omniglot [21], *mini*ImageNet [33]) by a series of methods [28, 50, 53, 51, 52, 4, 43, 7, 18]. Despite their success, most of them train and evaluate on only one dataset (*i.e.*, *single-domain*), failing to learn generalized model across different visual domains (*i.e.*, *multi-domain*). In fact, the need for cross-domain generalization is prevalent in practical applications [44, 13, 24, 23, 14, 11, 26]. For example, we would expect a model trained on ImageNet [6] to be applied on TrafficSigns [17] without collecting extra target training examples (Figure 1).

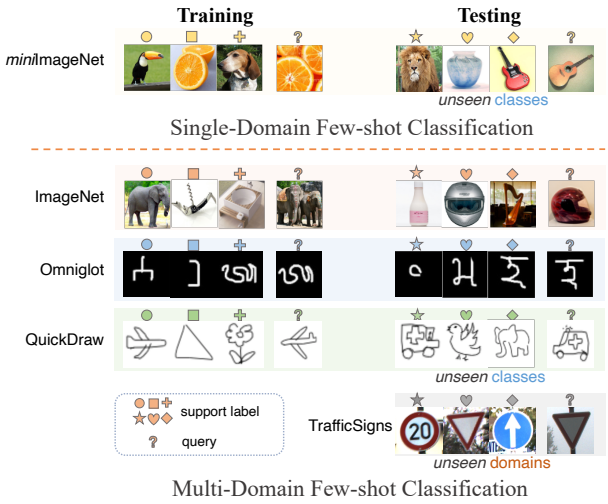

Figure 1. Multi-domain few-shot classification differs from single-domain few-shot classification in two aspects: (1) it contains multiple diverse datasets for training and extra unseen domains for test; (2) there exists potential correlations across multiple domains, *e.g.*, both Omniglot and QuickDraw contain simple shapes.

To break the limitations of existing few-shot classification methods and benchmarks, [44] have proposed a new

*Part of this work was done when Yanbin Liu was an intern at Baidu Research. Yi Yang is the corresponding author.

benchmark, META-DATASET, consisting of multiple diverse datasets and raised the new problem of multi-domain few-shot classification. It differs from conventional single-domain few-shot classification in two aspects (as shown in Figure 1): (1) It contains multiple diverse datasets for training and extra unseen domains for test; (2) potential correlations exist across multiple domains, *e.g*., both Omniglot and QuickDraw contain simple shapes.

These differences pose two challenges for multi-domain few-shot classification: (1) how to efficiently generate multi-domain representation, and (2) how to explore domain correlations for better cross-domain generalization. Current multi-domain methods can not address these challenges well. For example, CNAPs [34] trains a general adaptation network using all training datasets, leading to the single-mode and general-purpose adaptation. For substantially different domains (*e.g*., ImageNet and Omniglot in Figure 1), this single-mode adaptation network may be insufficient to handle all domains and potential interference may occur. In contrast, SUR [10] pre-trains multiple independent feature extraction networks to obtain the multi-domain feature representation. However, it is inefficient to maintain multiple replications of the feature extraction networks and domain-level information sharing is prohibited.

To address the drawbacks of the above methods, we propose a **Multi-Mode Modulator** (***tri*-M**) to simultaneously model the multi-domain feature representation and cross-domain correlations in a single network. First, the modulator is devised to achieve multi-domain representation by incorporating multiple modulation parameters in a single network, where each parameter describes a particular domain (called a mode). Given a dataset, the domain-aware features are efficiently generated by the well-designed separative selection and cooperative query mechanism.

Second, to explore the domain correlations, the modulation parameters are further divided into two sets: the domain-specific set and the domain-cooperative set, which work complementarily to explore both the intra-domain and inter-domain information. Concretely, the domain-specific set describes each domain independently to prevent negative interference among distant domains, *e.g*., ImageNet and Omniglot. The domain-cooperative set captures the inter-domain relations to guide beneficial information sharing among relevant domains to enrich their own domain-specific representation. Moreover, with the learned domain relations, the unseen domains can be described by an adaptive combination of the relevant seen domains, showing the extrapolation ability of our model.

Moreover, by design, our modulator is flexible to change the number of modes to deal with varying numbers of datasets and the number of modulation layers to satisfy desired model capacity. In experiments, we show the effectiveness of each component in our model and visually interpret how the selection and query mechanism work on the domain-specific and domain-cooperative sets of parameters. In summary, our main contributions are three-fold:

- We propose a multi-mode modulator to deal with the multi-domain few-shot classification problem. The domain-aware features can be efficiently generated with our single-network multi-domain model.
- We explicitly model the domain correlations by the domain-specific and domain-cooperative parameter sets. They work complementarily to extract both the intra-domain and inter-domain information.
- We achieve state-of-the-art performance on the challenging META-DATASET benchmark, especially for unseen test domains.

2. Related Work

Meta-learning. Recent few-shot learning methods rely on the meta-learning [42, 38, 37] paradigm. Most of them are divided into two types: metric-based methods and optimization-based methods. Metric-based methods [47, 40, 51, 41, 31] utilize a feature encoder to extract features from both the labeled and unlabeled images and employ a metric function (*e.g*., euclidean distance [40]) to calculate the similarity scores for predicting the category of unlabeled images. Optimization-based methods [12, 33, 35, 48] learn an update rule for the parameters of a base-learner model with the few examples from a sequence of episodes.

Multi-domain and cross-domain few-shot classification. Chen *et al*. [4] recently found that current meta-learning approaches do not generalize well to the unseen domains. To mitigate this, [44] proposed a more realistic, large-scale and diverse benchmark: META-DATASET and raised the new problem of multi-domain few-shot classification (see Figure 1). META-DATASET provides a well-defined evaluation test-bed and inspired a series of new few-shot learning methods [34, 10, 2, 27, 3, 8, 1, 45, 36].

Similarly, [13] proposed the cross-domain few-shot learning (CD-DSL) benchmark, in which ImageNet [6] is used for source training, and domains of varying dissimilarity from ImageNet (ranging from crop disease, satellite, and medical images) are used for target evaluation. Cross-domain few-shot learning differs from multi-domain few-shot learning in that it focuses on the domain-shift from the source training domain to different target evaluation ones while multi-domain few-shot learning tries to learn a well-performing model for both seen and unseen domains.

Multimodal feature representations for few-shot learning. A straightforward way to use multiple representations is simply training N individual models and apply a feature-level or prediction-level fusion. [9] designs an ensemble of deep networks to leverage the variance of classi-

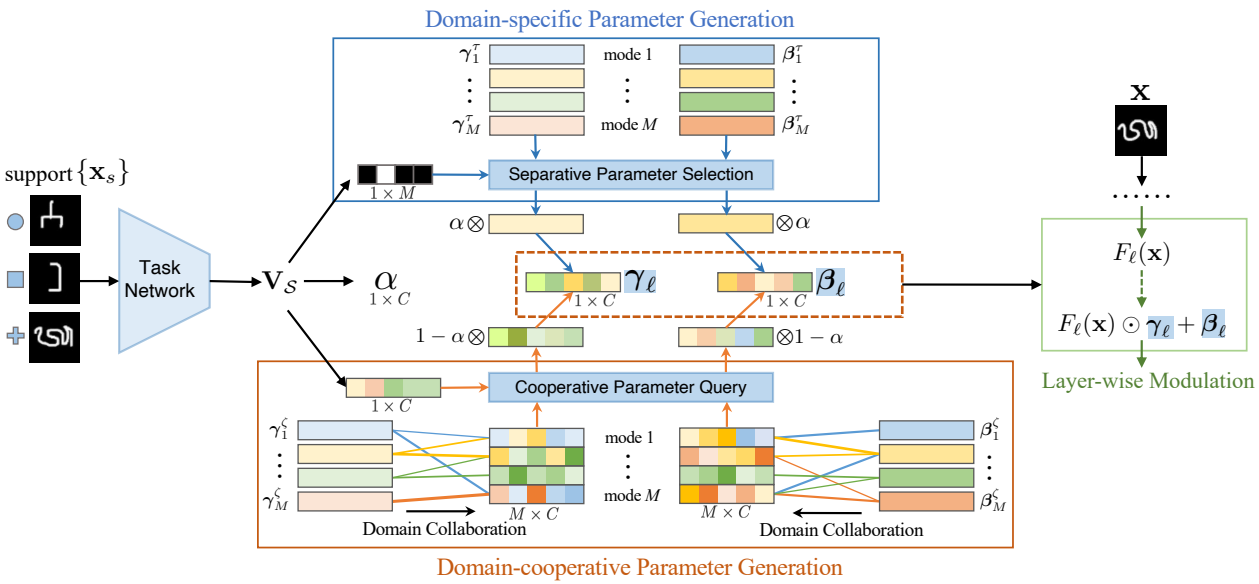

Figure 2. The proposed ***tri*-M** framework. We first input the support images into to the task network to obtain a domain descriptor $\mathbf{V}_\mathcal{S}$. The domain descriptor is followed by three heads to generate a hard-gating, a fusion parameter α, and a query vector. The hard-gating is used to guide the *Separative Parameter Selection* module to select from the domain-specific parameters. The query vector is used as a probe for the *Cooperative Parameter Query* module to query from the domain-cooperative parameters, which are generated by applying domain collaboration on individual parameters. Then, the selected and queried parameters are fused channel-wisely with weights α and $1 - \alpha$ to get the ℓ-th layer parameters $\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell$. Finally, the layer-wise feature modulation is applied on any support or query images.

fiers for few-shot classification. [10] obtains a multi-domain representation by pre-training multiple neural networks and re-weights the multiple features at inference time. [48] proposes a multimodal MAML (MMAML) framework to modulate its meta-learned priors with parameters generated from the modulation network. The multimodal tasks share the same modulation network to learn a general adaptation mechanism.

3. Problem Definition

Conventional few-shot classification is usually formulated as a meta-learning problem: instead of sampling a mini-batch of examples from the training classes, we create a series of learning tasks (*i.e.*, episodes) with each task composed of few labeled examples called a *support set* and several unlabeled examples called a *query set*. Specifically, in each episode, a small subset of N classes are sampled from all training classes to construct a support set and a query set. The support set contains K examples for each of the N classes (*i.e.*, N-way K-shot setting) denoted as $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{N\times K}, y_{N\times K})\}$, while the query set $\mathcal{Q} = \{(\mathbf{x}_1^*, y_1^*), \ldots, (\mathbf{x}_q^*, y_q^*)\}$ includes different samples from the same N classes. The performance is evaluated on the $(\mathcal{S}, \mathcal{Q})$ tasks sampled from the unseen test classes.

The multi-domain few-shot classification shares the basic structure with conventional few-shot classification, but has some crucial differences. In conventional few-shot classification, $(\mathcal{S}, \mathcal{Q})$ are sampled from only one dataset D for both training and test. In the multi-domain few-shot classification, during training, $(\mathcal{S}, \mathcal{Q})$ are sampled from multiple datasets $D_{\text{tr}} = \{D_1, D_2, \ldots, D_n\}$, and during test, $(\mathcal{S}, \mathcal{Q})$ are sampled from $D_{\text{te}} = \{D_1, D_2, \ldots, D_n, D_1^{\text{un}}, \ldots, D_u^{\text{un}}\}$ including both the training datasets D_{tr} and unseen datasets $\{D_i^{\text{un}}\}_{i=1}^u$. This means that a model to solve this task must generalize to multiple datasets including unseen datasets. Another difference is that unlike the typical N-way K-shot setting, the sampled tasks in multi-domain few-shot classification can have diverse number of classes and imbalanced number of images per class. For instance, in META-DATASET [44], N is sampled from the range $[5, 50]$, and K and $\mathcal{Q}$ are sampled with a complex procedure. These two differences make it non-trivial to directly apply the existing few-shot learning methods for the multi-domain few-shot classification problem.

4. Our Framework

We visualize the proposed ***tri*-M** framework in Figure 2. In our framework, we fix the backbone network and utilize the multi-mode modulator to generate the scale and translation parameters $\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell$ for layer-wise feature modulation. Specifically, we first feed a support set into the task network to obtain a domain descriptor $\mathbf{V}_S$, which is followed by three heads to generate a hard-gating, a fusion parameter α, and a query vector. The hard-gating selects parameters

from the domain-specific parameter set and the query vector queries parameters from the domain-cooperative parameter set. Then, they are fused with α and $1-\alpha$ to get the layer-wise parameters $\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell$ for feature modulation. Overall, our framework achieves the single-network multi-domain feature representation in a parameter-efficient way. In the following, we first describe the layer-wise feature modulation, and then explain how the modulation parameters are generated and fused. Finally, we describe the classifier.

4.1. Layer-wise Feature Modulation

Feature adaptation is a critical issue in few-shot learning since the model has to quickly generalize after seeing very few examples. Existing methods [12, 33] address this issue by adapting all the network parameters using few support examples, which are usually slow and prone to overfitting [44].

To adapt the network parameters in a parameter-efficient manner, we utilize a Feature-wise Linear Modulation (FiLM) layer [32]. The main idea is to freeze the parameters of a pre-trained backbone network and apply a channel-wise linear transformation for feature modulation. Specifically, for an input image $\boldsymbol{x}$, FiLM scales and shifts its l-th layer feature map $\boldsymbol{F}_\ell(\boldsymbol{x}) \in \mathbb{R}^{H\times W\times C}$ as

$$\hat{\boldsymbol{F}}_\ell(\boldsymbol{x}) = \boldsymbol{F}_\ell(\boldsymbol{x}) \odot \boldsymbol{\gamma} + \boldsymbol{\beta}\,, \tag{1}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^C, \boldsymbol{\beta} \in \mathbb{R}^C$ are the learnable parameters, H, W and C indicates the height, width and channel number of feature maps.

In our implementation, the FiLM layer is applied to every convolution block between batch normalization (BN) and ReLU. Intuitively, since the linear transformation is applied after BN layer, the pre-trained BN statistics (*i.e.*, mean and variance) can be adapted to match the target dataset. Therefore, the distribution of the output feature maps can be well-aligned to the target dataset. Moreover, for multiple datasets, it is parameter-efficient to achieve multi-mode feature adaptations with different $(\boldsymbol{\gamma}, \boldsymbol{\beta})$.

Formally, if we denote f as the neural network function, θ as the pre-trained network parameters. The feature representation for an image $\boldsymbol{x}$ can be denoted as $\boldsymbol{z} = f_\theta(\boldsymbol{x}; \{\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell\}_{\ell=1}^L)$, where $\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell$ are the modulation parameters of layer ℓ.

4.2. The Multi-Mode Modulator

Given M datasets, a straightforward way to implement multi-domain representation is to pre-train M individual networks [10]. Although being simple to implement, it is inefficient to train M models and inference M times. Moreover, the domain relations are ignored for potential knowledge transfer across datasets. Another way is to train a general feature adaptation network for all datasets [34, 48]. This single-mode adaptation can not be equally effective for substantially different datasets and may cause interference, *e.g.*, ImageNet and Omniglot.

In contrast, we apply a single-network multi-mode feature modulation. Each mode represent a particular dataset/domain, and has its own learnable parameters that are further divided into two sets: the *domain-specific set* $\{(\boldsymbol{\gamma}_i^{\boldsymbol{\tau}}, \boldsymbol{\beta}_i^{\boldsymbol{\tau}})\}_{i=1}^M$ and the *domain-cooperative set* $\{(\boldsymbol{\gamma}_i^{\boldsymbol{\zeta}}, \boldsymbol{\beta}_i^{\boldsymbol{\zeta}})\}_{i=1}^M$. The specific set provides separated adaptation for each domain to prevent interference while the cooperative set explores useful correlations to encourage information sharing. These two sets work complementarily to achieve effective intra-domain and inter-domain adaptation.

4.2.1 Task Network

To generate domain-aware features, we utilize a task network to obtain domain-level description $\mathbf{V}_S$ for each learning task. Specifically, we first feed the support set images $\mathcal{S} = \{\boldsymbol{x}_s\}_{s=1}^N$ to a lightweight network with 5 sequential blocks (each block contain a 3×3 convolution with 64 channels followed by BN, ReLU, and 2×2 max pooling). Then the outputs are average-pooled in both *spatial* and *batch* dimension to get a single descriptor $\boldsymbol{V}_{\mathcal{S}} \in \mathbb{R}^{1\times64}$ of the support set $\mathcal{S}$. The descriptor is invariant to the permutation of the support set due to the average pooling. More details are in the supplementary material.

4.2.2 Domain-specific Parameter Generation

Separative parameter selection. The domain descriptor $\boldsymbol{V}_{\mathcal{S}}$ encodes the necessary information to infer the domain identity of the support set $\mathcal{S}$. Therefore, we can employ a hard-gating mechanism to choose the proper domain-specific parameters from M existing modes. Specifically, let $\boldsymbol{W}^g \in \mathbb{R}^{64\times M}$ and $\boldsymbol{b}^g \in \mathbb{R}^{1\times M}$ be learnable parameters. Then we can construct the selection gates as

$$\boldsymbol{g} = \mathrm{softmax}(\boldsymbol{V}_{\mathcal{S}}\boldsymbol{W}^g + \boldsymbol{b}^g) \in \mathbb{R}^{1\times M}, \tag{2}$$

where $\boldsymbol{g}_i$ indicates the probability that the support set $\mathcal{S}$ belongs to the i-th mode. We choose the mode index k with highest probability, *i.e.*, $k = \arg\max_i \boldsymbol{g}_i$ and select the domain-specific parameters as $(\boldsymbol{\gamma}^g, \boldsymbol{\beta}^g) = (\boldsymbol{\gamma}_k^{\boldsymbol{\tau}}, \boldsymbol{\beta}_k^{\boldsymbol{\tau}})$. To ensure each member in the domain-specific parameter set exclusively describe the corresponding domain, we introduce a domain classification loss. During training, the domain identity of the the support set $\mathcal{S}$ is known in advance. Using this as the ground truth $\boldsymbol{y}_{\text{domain}}$, we define the domain classification loss as

$$L_{\text{domain}} = \lambda \mathcal{L}^{\text{CE}}(\boldsymbol{g}, \boldsymbol{y}_{\text{domain}})\,, \tag{3}$$

where $\mathcal{L}^{\text{CE}}$ denotes cross-entropy loss, $\boldsymbol{y}_{\text{domain}} \in \mathbb{R}^{1\times M}$ denotes the one-hot ground-truth vector of domain, and $\lambda > 0$ is a hyperparameter controlling the effect of the loss.

4.2.3 Domain-cooperative Parameter Generation

Domain collaboration. Although the hard-gating and domain loss prevent interference among various modes of the domain-specific parameter set, they also block effective information sharing across domains. To resolve this issue, we explore the mode correlations through another set of parameters: domain-cooperative set $\{(\boldsymbol{\gamma}_i^{\boldsymbol{\zeta}}, \boldsymbol{\beta}_i^{\boldsymbol{\zeta}})\}_{i=1}^M$. These parameters are originally uncorrelated with random initialization, so we use Transformer [46] to explicitly learn the potential correlations among domains. An attention function generates the correlated transformation of the inputs as

$$\mathrm{Attn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top / \sqrt{d_k})\boldsymbol{V},$$

where d_k is the feature dimension of $\boldsymbol{Q}, \boldsymbol{K}$. To increase the expressive power, multi-head attention is usually applied as

$$\mathrm{MHAttn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)\boldsymbol{W}^O,$$

where $\mathrm{head}_i = \mathrm{Attn}(\boldsymbol{Q}\boldsymbol{W}_i^{\boldsymbol{Q}}, \boldsymbol{K}\boldsymbol{W}_i^{\boldsymbol{K}}, \boldsymbol{V}\boldsymbol{W}_i^{\boldsymbol{V}}), \boldsymbol{W}_i^{\boldsymbol{Q}} \in \mathbb{R}^{d\times d_k}, \boldsymbol{W}_i^{\boldsymbol{K}} \in \mathbb{R}^{d\times d_k}, \boldsymbol{W}_i^{\boldsymbol{V}} \in \mathbb{R}^{d\times d_v}$, and $\boldsymbol{W}^O \in \mathbb{R}^{hd_v\times d}$. To apply the multihead attention, we first pack $\{\boldsymbol{\gamma}_i^{\boldsymbol{\zeta}}\}_{i=1}^M$ into a matrix $\boldsymbol{\gamma}^{\boldsymbol{\zeta}} \in \mathbb{R}^{M\times C}$ and $\{\boldsymbol{\beta}_i^{\boldsymbol{\zeta}}\}_{i=1}^M$ into $\boldsymbol{\beta}^{\boldsymbol{\zeta}} \in \mathbb{R}^{M\times C}$. Then, the correlated parameters are computed as $\boldsymbol{\gamma}^{\mathrm{corr}} = \mathrm{MHdAttn}(\boldsymbol{\gamma}^{\boldsymbol{\zeta}}, \boldsymbol{\gamma}^{\boldsymbol{\zeta}}, \boldsymbol{\gamma}^{\boldsymbol{\zeta}})$ and $\boldsymbol{\beta}^{\mathrm{corr}} = \mathrm{MHAttn}(\boldsymbol{\beta}^{\boldsymbol{\zeta}}, \boldsymbol{\beta}^{\boldsymbol{\zeta}}, \boldsymbol{\beta}^{\boldsymbol{\zeta}})$. Now, $(\boldsymbol{\gamma}^{\mathrm{corr}}, \boldsymbol{\beta}^{\mathrm{corr}})$ are the correlated parameters taking domain collaboration into account. For example, relevant domains such as Omniglot and QuickDraw may have similar parameters.

Cooperative parameter query. We employ a quey strategy to obtain the domain-cooperative parameter. At first, the query vector is obtained $\boldsymbol{Q}_{\mathcal{S}} = \boldsymbol{V}_{\mathcal{S}}\boldsymbol{W}^a + \boldsymbol{b}^a \in \mathbb{R}^{1\times C}$. Then, the query scores of $\boldsymbol{\gamma}^{\mathrm{corr}}$ is computed as $s^\gamma = \mathrm{softmax}(\boldsymbol{Q}\boldsymbol{\gamma}^{\mathrm{corr}\top}/\sqrt{C}) \in \mathbb{R}^{1\times M}$. Finally, the queried parameter is $\boldsymbol{\gamma}^a = s^\gamma\boldsymbol{\gamma}^{\mathrm{corr}}$. Similarly, $\boldsymbol{\beta}^a$ can be obtained. For a seen domain, the query of domain-cooperative parameters can activate all its related modes by properly setting the query scores. Thus, the parameters of all activated modes can be jointly learned, which potentially increases the training data of all related domains. For an unseen domain, although it is not shown at training time, it can still utilize the learned query mechanism to find a weighted combination of the relevant existing domain-cooperative parameters for appropriate feature modulation. Therefore, our model provides an effective way of extrapolating to unseen domains.

4.2.4 Parameter Fusion

Having selected the domain-specific parameters $(\boldsymbol{\gamma}^g, \boldsymbol{\beta}^g)$ and computed the domain-cooperative parameters $(\boldsymbol{\gamma}^a, \boldsymbol{\beta}^a)$, we combine them into the final modulation parameters. Due to the diverse nature of the training tasks and datasets, a naïve average fusion may not be optimal. Instead, we use the adaptive fusion scheme for channel-wise fusion. We first compute the adaptive fusion ratio as $\boldsymbol{\alpha} = \mathrm{sigmoid}(\boldsymbol{V}_{\mathcal{S}}\boldsymbol{W}^f + \boldsymbol{b}^f) \in \mathbb{R}^{1\times C}$, and then combine the parameters as $\boldsymbol{\gamma} = \boldsymbol{\alpha}\boldsymbol{\gamma}^g + (1-\boldsymbol{\alpha})\boldsymbol{\gamma}^a, \boldsymbol{\beta} = \boldsymbol{\alpha}\boldsymbol{\beta}^g + (1-\boldsymbol{\alpha})\boldsymbol{\beta}^a$. Doing so, the model can choose proper fusion ratio according to the characteristic of the support set.

4.3. Classifier

Metric-based classifier has been widely-used in few-shot learning [47, 40, 41, 2] and reported to improve performance. Following [2], we use the structured Mahalanobis distance to formulate our classifier since it shows promising performance. We first compute the adapted features for the support set, $\{\boldsymbol{z}_s\}_{s=1}^N = f_\theta(\{\boldsymbol{x}_s\}_{s=1}^N; \{\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell\}_{\ell=1}^L)$. Then for each class, we compute the class mean $\boldsymbol{\mu}_k$ and regularized covariance matrix $\boldsymbol{Q}_k$. Given a query feature $\boldsymbol{z}_q = f_\theta(\boldsymbol{x}_q; \{\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell\}_{\ell=1}^L)$, the class probability is constructed as

$$p(y_q = k|\boldsymbol{x}_q) \propto \exp(-(\boldsymbol{z}_q - \boldsymbol{\mu}_{\boldsymbol{k}})^T\boldsymbol{Q}_k^{-1}(\boldsymbol{z}_q - \boldsymbol{\mu}_k)).$$

5. Experiments

We present the experiments to analyze the performance of our multi-mode modulator. We first describe the datasets being used and implementation details, and present a comparison of ours to the recent state-of-the-art methods. Next, we show the effectiveness of the domain-specific and domain-cooperative parameter sets, accuracy under various choice of the number of modes and number of modulation layer groups. Finally, we present interpretable visualizations of the selection gates (domain-specific parameter set) and query scores (domain-cooperative parameter set).

5.1. Datasets and Implementation Details

Benchmark. We test our method on the large-scale multi-domain few-shot learning benchmark METADATASET [44]. It consists of 10 widely used datasets with various data distributions from different visual domains, including natural images (ImageNet [6], Birds [49], VGG Flower [30], Fungi [39]), common objects (MSCOCO [25], Traffic Signs [17], Aircraft [29]), hand-written characters (Omniglot [21], Quick Draw [15]) and textures (Describable Textures [5]). To be consistent with previous work [34, 10], we train our model on the official training splits of the 8 datasets (according to [44]) and use the test splits of each dataset to evaluate the ***in-domain accuracy***. In addition, we use the remaining two (Traffic Signs and MSCOCO) as well as 3 external datasets, namely MNIST [22], CIFAR10 [20] and CIFAR100 as the unseen domains to evaluate the ***out-of-domain accuracy***. All 13 datasets are used to report the ***overall accuracy***. Few-shot tasks are generated following [44]. The generated tasks

Table 1. Comparison to the state-of-the-art methods on META-DATASET. Error intervals show the 95% confidence interval, and the numbers in **bold** have intersecting confidence intervals with the most accurate method. Average rank is obtained by ranking methods on each dataset and averaging the ranks. Due to the shuffling issue[2], Meta-Dataset updated the evaluation on TrafficSigns. Therefore, We report the updated accuracy of all methods on TrafficSigns (i.e. 63.0 ± 1.0 for *tri*-M) in the Supplementary.

Dataset	ProtoMAML [44]	BOHB-E [36]	AR-CNAPS [34]	TaskNorm [3]	SimpleCNAPS [2]	SUR-pf [10]	SUR [10]	*tri*-M (Ours)
ImageNet	47.9±1.1	55.4±1.1	52.3±1.0	50.6±1.1	**58.6±1.1**	56.4±1.2	56.3±1.1	**58.6±1.0**
Omniglot	82.9±0.9	77.5±1.1	88.4±0.7	90.7±0.6	91.7±0.6	88.5±0.8	**93.1±0.5**	**92.0±0.6**
Aircraft	74.2±0.8	60.9±0.9	80.5±0.6	83.8±0.6	82.4±0.7	79.5±0.8	**85.4±0.7**	82.8±0.7
Birds	70.0±1.0	73.6±0.8	72.2±0.9	74.6±0.8	**74.9±0.8**	**76.4±0.9**	71.4±1.0	**75.3±0.8**
Textures	67.9±0.8	72.8±0.7	58.3±0.7	62.1±0.7	67.8±0.8	**73.1±0.7**	71.5±0.8	71.2±0.8
QuickDraw	66.6±0.9	61.2±0.9	72.5±0.8	74.8±0.7	77.7±0.7	75.7±0.7	**81.3±0.6**	77.3±0.7
Fungi	42.0±1.1	44.5±1.1	47.4±1.0	48.7±1.0	46.9±1.0	48.2±0.9	**63.1±1.0**	48.5±1.0
VGGFlower	88.5±0.7	**90.6±0.6**	86.0±0.5	**89.6±0.6**	**90.7±0.5**	**90.6±0.5**	82.8 ±0.7	**90.5±0.5**
TrafficSigns	52.3±1.1	57.5±1.0	60.2±0.9	67.0±0.7	73.5±0.7	65.1±0.8	70.4±0.8	**78.0±0.6**
MSCOCO	41.3±1.0	**51.9±1.0**	42.6±1.1	43.4±1.0	46.2±1.1	**52.1±1.0**	**52.4±1.1**	**52.8±1.1**
MNIST	NA	NA	92.7±0.4	92.3±0.4	93.9±0.4	93.2±0.4	94.3±0.4	**96.2±0.3**
CIFAR10	NA	NA	61.5±0.7	69.3±0.8	**74.3±0.7**	66.4±0.8	66.8±0.9	**75.4±0.8**
CIFAR100	NA	NA	50.1±1.0	54.6±1.1	**60.5±1.0**	57.1±1.0	56.6±1.0	**62.0±1.0**
In-Domain Avg	67.5	67.1	69.7	71.9	73.8	73.6	**75.6**	74.5
Out-of-Domain Avg	46.8	54.7	61.5	65.3	69.7	66.8	68.1	**72.9**
Overall Avg	63.4	64.6	66.5	69.3	72.2	70.9	72.7	**73.9**
Average Rank	7.2	5.7	6.1	4.6	3.1	3.6	3.2	**2.1**
Learnable Parameters	10.49M	NA	13.4M	9.39M	8.60M	1.67M	79.45M	7.78M
Forward Pass	1	1	1	1	1	8	8	1

can be of varying number of classes, varying number of shots and class imbalance. For evaluation, 600 tasks on each dataset are sampled and the average accuracy of each dataset, in-domain, out-of-domain and overall are reported.

Implementation Details. For a fair comparison, we follow [34, 2] to employ ResNet18 [16] as the backbone which is pre-trained on the training split of the META-DATASET version of ImageNet. The proposed multi-mode modulator is applied on all except for the first convolutional layers. For multihead attention, $h = 3, d_k = d_v = 32$. λ in Eq. 3 is set to 0.001 according to the validation set. Images of all datasets are resized to 84×84 pixels and no data augmentation is applied during training. We train in an end-to-end fashion for 150,000 episodes with the Adam [19] optimizer, using a batch size of 16 episodes, and a fixed learning rate of 0.002.

5.2. Comparison to state-of-the-art methods

We compare our ***tri*-M** method with recent state-of-the-art few-shot methods and report the results in Table 1. Besides the accuracy metric, we also report the widely-used average rank which is obtained by ranking methods on each dataset and averaging them. In Table 1, the proposed multi-mode modulator achieves the best average rank (+1.0) and overall accuracy (+1.2%), setting a new state of the art on META-DATASET. Specifically, our method is among the most accurate methods on 9 out of 13 datasets, achieves the best out-of-domain accuracy (+3.2%) and second-best in-domain accuracy. The excellent out-of-domain accuracy indicates that our method can effectively generalize to unseen test domains.

Considering the number of learnable parameters, our method is the best one among all models which need to forward only *one* network. SUR-pf [10] has a smaller number of learnable parameters at the cost of forwarding *eight* networks, which is inefficient. Note that SUR [10] has more than $10\times$ learnable parameters[3] and $8\times$ forward compared with our method to achieve a slightly better in-domain accuracy.

5.3. Ablation Study

Effect of the domain-specific and domain-cooperative parameter sets. Table 2 shows the results comparison of only using domain-specific (*Spec*), only using domain-cooperative (*Coop*), and using all (*Spec*+*Coop*) parameter sets. The domain-cooperative parameter set generally performs better than the domain-specific ones. We attribute this to the information exchange across different domains, which potentially leads to increased number of relevant training examples for correlated domains. For example, compared with *Spec*, *Coop* on Omniglot and QuickDraw (both contains simple while-and-black shapes) benefits from each other, with an improvements of 2.3% and 1.1%, relatively.

The fusion model, *Spec*+*Coop* achieves the best performance over 12 out of 13 datasets and obtains the highest average accuracy, indicating that the two parameter sets are mutually complementary and the fusion strategy is effective. *Spec*+*Coop* has 7.78M learnable parameters, which is only 0.13M larger than the single-best model *Coop*.

[2]https://github.com/google-research/meta-dataset/issues/54

[3]Note that we only count the trainable parameters for all methods, excluding the ImageNet pre-trained backbone.

Table 2. Effect of the domain-specific (*Spec*) and domain-cooperative (*Coop*) parameter sets.

Dataset	*Spec*	*Coop*	*Spec+Coop*
ImageNet	55.6	57.5	**58.6**
Omniglot	88.2	90.5	**92.0**
Aircraft	82.1	81.5	**82.8**
Birds	73.3	75.2	**75.3**
Textures	68.2	69.4	**71.2**
QuickDraw	75.1	76.2	**77.3**
Fungi	48.4	48.0	**48.5**
VGGFlower	85.9	90.1	**90.5**
TrafficSigns	74.4	76.5	**78.0**
MSCOCO	53.2	**53.3**	52.8
MNIST	94.8	94.3	**96.2**
CIFAR10	73.0	74.3	**75.4**
CIFAR100	61.9	61.8	**62.0**
In-Domain Avg	72.1	73.6	**74.5**
Out-of-Domain Avg	71.5	72.0	**72.9**
Overall Avg	71.9	73.0	**73.9**
Learnable Parameters	0.22M	7.65M	7.78M

Table 3. Accuracy with various number of modes. $\{4, 4\}$ denotes 4 domain-specific modes and 4 domain-cooperative modes.

Datasets	Number of Modes			
	{4,4}	{8,8}	{8,16}	{16,16}
ImageNet	57.9	**58.6**	57.7	58.0
Omniglot	**92.4**	92.0	91.6	91.4
Aircraft	82.2	82.8	**83.2**	81.1
Birds	75.2	**75.3**	74.7	74.0
Textures	66.5	**71.2**	67.0	68.4
QuickDraw	77.2	77.3	**77.6**	77.0
Fungi	48.3	**48.5**	47.5	48.0
VGGFlower	89.7	**90.5**	89.8	89.5
TrafficSigns	75.7	**78.0**	76.8	73.0
MSCOCO	50.2	**52.8**	52.7	48.3
MNIST	94.7	**96.2**	94.8	94.4
CIFAR10	73.6	**75.4**	74.7	74.4
CIFAR100	60.9	**62.0**	61.5	61.1
In-Domain Avg	73.7	**74.5**	73.6	73.4
Out-of-Domain Avg	71.0	**72.9**	72.1	70.2
Overall Avg	72.7	**73.9**	73.0	72.2
Learnable Parameters	7.71M	7.78M	7.84M	7.90M

Performance of varying number of modes. On META-DATASET, there are 8 training datasets representing 8 different domains. By default, in the proposed multi-mode modulator, we set the number of modes $M = 8$ for both the domain-specific and domain-cooperative parameter sets, aiming to link each mode to a specific dataset. In addition, it is flexible to change the number of modes for each of the parameter sets to decrease or increase the model capacity. To decrease the number of modes for domain-specific set, we manually merge different datasets by their visual similarity to form new domains and calculate a new domain classification loss (Eq. 3). For example, we can form 4 new domains: {ImageNet, Birds, DTD}, {Omniglot, QuickDraw}, {Fungi, VGGFlower}, and {Aircraft}. To increase the number of modes for domain-specific set, we assign 2 modes for each dataset and average their parameters before hard-gating. To change the number of modes for domain-cooperative set, we directly modify the cooperative mode number M.

We experimented with various mode combinations and report the results in Table 3. The best performance is achieved with the combination $\{8, 8\}$, which contains 8 modes for both parameter sets. This is not surprising since we have 8 training datasets in total. When we decrease the modes to $\{4, 4\}$, the overall accuracy drops a little, but Omniglot increases slightly. This is due to the increased number of training examples by merging Omniglot and QuickDraw as a new domain. When we increase the modes to $\{8, 16\}$ or $\{16, 16\}$, the overall accuracy drops. The redundant modes may hinder the datasets to learn the inherent relationships. Note that when we increase the number of modes, the number of learnable parameters only increases slightly. This shows that we can extend our model to much larger number of datasets (*e.g.*, more than 16) without significantly increasing the number of parameters.

Performance of varying number of modulation layer groups. The ResNet18 [16] backbone has 4 layer groups with each group containing 2 building blocks and each block containing 2 convolutional layers. By default, feature modulation is applied after the BN of each convolutional layer (details in the supplementary). In order to study the modulation property with respect to different layers, we gradually add modulation from deep to shallow layer groups. As a special variant, we only add modulation on the second convolutional layer in each of the building blocks across all layer groups, dubbed *half* in Table 4.

From Table 4, we can see that the accuracy increases with the increased number of modulation layer groups from 1 to 4. The overall accuracy increase from 3 groups to 4 groups is 2.2%. This indicates that the modulations to shallower layers are indispensable since these shallow layers extract low-level features, which is more generalizable across datasets. Moreover, only modulating the second layers (*half*) obtains approximate accuracy compared with modulation on all layers (4 groups). However, *half* has 4.04M learnable parameters, only 52% of all 4 groups. This provides us a way to get a slim yet competitive model by applying modulation evenly on all candidate layers.

5.4. Visualization

We first visualize the selection gates of the domain-specific parameter set in Figure 3. Each training dataset focus on one mode, showing the effectiveness of the hard-gating mechanism. For the unseen test datasets, they try to select the modes according to their relevance to training datasets. TrafficSigns, MSCOCO, CIFAR10, and CIFAR100 contain common images, showing high relevance

Table 4. Accuracy with various number of modulation layer groups. *half* denotes that we only modulate the second covolutional layer in each of the basic blocks across all layer groups.

Datasets	Number of Modulation Layer Groups				
	1	2	3	*half*	4
ImageNet	54.2	57.4	57.5	57.7	**58.6**
Omniglot	83.2	88.8	91.1	91.3	**92.0**
Aircraft	74.0	80.7	82.7	82.2	**82.8**
Birds	65.4	71.7	73.8	75.0	**75.3**
Textures	70.5	68.3	68.0	69.9	**71.2**
QuickDraw	70.5	75.3	76.5	77.2	**77.3**
Fungi	42.2	45.6	47.0	**48.5**	**48.5**
VGGFlower	88.6	89.6	89.4	90.0	**90.5**
TrafficSigns	72.1	73.2	71.8	75.0	**78.0**
MSCOCO	50.2	49.3	49.9	52.5	**52.8**
MNIST	91.7	94.4	94.2	94.4	**96.2**
CIFAR10	69.3	71.1	72.2	75.3	**75.4**
CIFAR100	56.1	57.5	58.4	61.5	**62.0**
In-Domain Avg	68.6	72.2	73.3	74.0	**74.5**
Out-of-Domain Avg	67.9	69.1	69.3	71.7	**72.9**
Overall Avg	68.3	71.0	71.7	73.1	**73.9**
Learnable Parameters	4.81M	6.66M	7.49M	4.04M	7.78M

to ImageNet. MNIST contains black-white digit images, showing high relevance to Omniglot. In addition, TrafficSigns and MNIST show relevance to QuickDraw as they all contain simple shapes. Overall, the domain-specific information can be successfully learned with the selection mechanism.

The query scores of the domain-cooperative parameter set are shown in Figure 4. The scores visualization is quite different from the gates visualization. ImageNet attends to all modes with relatively even values, because the backbone is pre-trained on ImageNet. Datasets (CIFAR10, CIFAR100, and MSCOCO) that contain common images and are visually similar to ImageNet ensemble the scores of ImageNet. To the contrary, datasets (Omniglot, QuickDraw, and MNIST) that share less overlaps with ImageNet show a unimodal high value on mode 3. Some others (Aircraft, VGGFlower) are fine-grained datsets, they show trimodal scores. Other implicit correlations are also learned with different mode combinations. Overall, the domain-cooperative information can be automatically learned and obtained with the cooperative query mechanism.

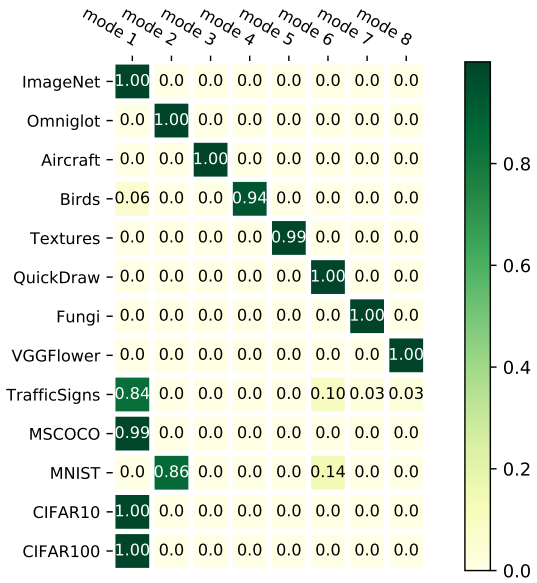

Figure 3. Selection gates of the domain-specific parameter set.

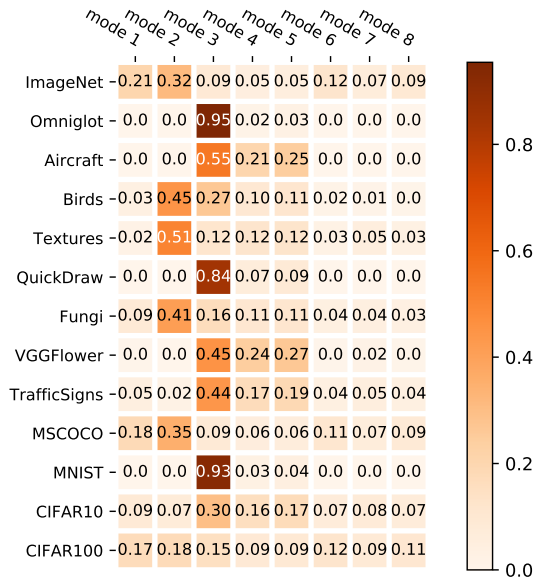

Figure 4. Query scores of the domain-cooperative parameter set.

6. Conclusion

In this paper, we deal with the multi-domain few-shot classification problem. It differs from single-domain few-shot classification in two aspects: 1) it contains multiple diverse datasets for training and extra unseen domains for test; 2) potential correlations exist across multiple domains. We propose a single-network multi-mode modulator to apply layer-wisely to generate multi-domain representation. In the modulator, we further introduce the domain-specific and domain-cooperative parameter sets that work complementarily to extract the intra-domain and inter-domain information to model domain correlations. The state-of-the-art performance is achieved on the challenging META-DATASET, especially for the unseen test domains.

Acknowledgement Yanbin Liu, Linchao Zhu and Yi Yang were supported by ARC DP200100938. Juho Lee was supported by Institute of Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)) and Samsung Electronics Co., Ltd (IO201214-08176-01). Ling Chen was partially supported by ARC DP180100966.

References

[1] Peyman Bateni, Jarred Barber, Jan-Willem van de Meent, and Frank Wood. Improving few-shot visual classification with unlabelled examples. *arXiv preprint arXiv:2006.12245*, 2020.

[2] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14493–14502, 2020.

[3] John Bronskill, Jonathan Gordon, James Requeima, Sebastian Nowozin, and Richard E Turner. Tasknorm: Rethinking batch normalization for meta-learning. *arXiv preprint arXiv:2003.03284*, 2020.

[4] 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*, 2018.

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Yuhang Ding, Xin Yu, and Yi Yang. Modeling the probabilistic distribution of unlabeled data for one-shot medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[8] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *arXiv preprint arXiv:2007.11498*, 2020.

[9] 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*, pages 3723–3731, 2019.

[10] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision (ECCV)*, 2020.

[11] Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE transactions on neural networks and learning systems*, 2020.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135, 2017.

[13] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision*, pages 124–141. Springer, 2020.

[14] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[15] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[16] 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*, pages 770–778, 2016.

[17] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2013.

[18] Zihang Jiang, Bingyi Kang, Kuangqi Zhou, and Jiashi Feng. Few-shot classification via adaptive attention. *arXiv preprint arXiv:2008.02465*, 2020.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

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

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

[22] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.

[23] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[24] Peike Li, Xin Yu, and Yi Yang. Super-resolving cross-domain face miniatures by peeking at one-shot exemplar. *arXiv preprint arXiv:2103.08863*, 2021.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[26] Feng Liu, Guangquan Zhang, and Jie Lu. Heterogeneous domain adaptation: An unsupervised approach. *IEEE transactions on neural networks and learning systems*, 31(12):5588–5602, 2020.

[27] Lu Liu, William Hamilton, Guodong Long, Jing Jiang, and Hugo Larochelle. A universal representation transformer layer for few-shot image classification. *arXiv preprint arXiv:2006.11702*, 2020.

[28] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations*, 2019.

[29] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[30] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[31] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31:721–731, 2018.

[32] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[33] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016.

[34] 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.

[35] 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*, 2018.

[36] Tonmoy Saikia, Thomas Brox, and Cordelia Schmid. Optimized generic feature learning for few-shot classification across domains. *arXiv preprint arXiv:2001.07926*, 2020.

[37] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[38] Jürgen Schmidhuber. A neural network that embeds its own meta-levels. In *IEEE International Conference on Neural Networks*, pages 407–412. IEEE, 1993.

[39] B Schroeder and Y Cui. Fgvcx fungi classification challenge 2018, 2018.

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[41] 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*, pages 1199–1208, 2018.

[42] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[43] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

[44] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2019.

[45] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*, 2019.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[47] 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.

[48] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pages 1–12, 2019.

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

[50] Zhongwen Xu, Linchao Zhu, and Yi Yang. Few-shot object recognition from machine-labeled web images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2017.

[51] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[52] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020.

[53] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–766, 2018.