Domain-Adaptive Few-Shot Learning

An Zhao[1,2*] Mingyu Ding[3*] Zhiwu Lu[1,2†] Tao Xiang[4] Yulei Niu[5] Jiechao Guan[1] Ji-Rong Wen[1]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]The University of Hong Kong [4]University of Surrey, United Kingdom
[5]Nanyang Technological University, Singapore
zhaoan_ruc@163.com mingyuding@hku.hk luzhiwu@ruc.edu.cn

Abstract

Existing few-shot learning (FSL) methods make the implicit assumption that the few target class samples are from the same domain as the source class samples. However, in practice, this assumption is often invalid – the target classes could come from a different domain. This poses an additional challenge of domain adaptation (DA) with few training samples. In this paper, the problem of domain-adaptive few-shot learning (DA-FSL) is tackled, which is expected to have wide use in real-world scenarios and requires solving FSL and DA in a unified framework. To this end, we propose a novel domain-adversarial prototypical network (DAPN) model. It is designed to address a specific challenge in DA-FSL: the DA objective means that the source and target data distributions need to be aligned, typically through a shared domain-adaptive feature embedding space; but the FSL objective dictates that the target domain per class distribution must be different from that of any source domain class, meaning aligning the distributions across domains may harm the FSL performance. How to achieve global domain distribution alignment whilst maintaining source/target per-class discriminativeness thus becomes the key. Our solution is to explicitly enhance the source/target per-class separation before domain-adaptive feature embedding learning, to alleviate the negative effect of domain alignment on FSL. Extensive experiments show that our DAPN outperforms the state-of-the-arts. The code is available at https://github.com/dingmyu/DAPN.

1. Introduction

Unlike many visual recognition methods [17, 34, 60, 7, 1, 6, 8] that require a lot of supervision, few-shot learning (FSL) [9, 28, 23, 27, 16] aims to recognize a set of target (rare/new) classes by learning with sufficient labeled samples from a set of source/seen classes but only with a few labeled samples from the target classes.

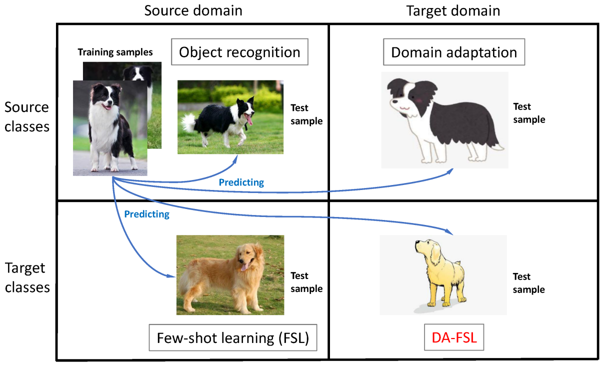

Figure 1. Illustration of the difference among four related visual recognition problems (i.e. objection recognition, FSL, domain adaptation, and domain-adaptive FSL).

FSL [42, 51] is often stated as a transfer learning problem [37] from the source classes to the target ones. The efforts so far are mainly on how to build a classifier with few samples. However, there is an additional challenge that has largely been neglected, that is, the target classes not only are poorly represented by the few training samples but also may come from a different domain from the source classes. For example, the target class samples could be collected by different imaging devices (e.g., mobile phone camera vs. single-lens reflex camera), resulting in different photo styles. In more extreme cases, the source classes could be captured in photos and the target ones in sketch or cartoon images. This means that the visual recognition model trained from the source classes needs to be adapted to both new classes and new domains, with few samples from the target classes. This problem is termed as domain-adaptive few-shot learning (DA-FSL), which is illustrated in Fig. 1.

DA-FSL is a more challenging problem due to the added objective of few-shot domain adaption. Neither convention-

[*] Equal Contribution
[†] Corresponding Author

al FSL [48, 49, 35, 21, 57, 52, 41] nor DA [54, 12, 3, 30, 2, 19, 62, 43, 50, 22] methods can be directly extended to DA-FSL, though they have drawn much attention. As far as we know, addressing both the few-shot DA and few-shot recognition problems jointly has never been attempted before. A straightforward solution seems to be combining an FSL with an existing DA method. In particular, most existing FSL methods [49, 52, 41, 11] rely on feature reuse to the target classes in a feature embedding space learned from the source [39]. It is thus natural to introduce the DA learning objective by aligning the source and target data distributions in that embedding space. Nevertheless, a naïve combination of existing DA and FSL methods fails to offer an effective solution (see Tables 2–4). This is because existing UDA methods assume that the target and source domains have identical label space. Given that they are mainly designed for distribution alignment across domains (recently focusing on per-class alignment [47, 32, 5, 25, 46]), they are intrinsically unsuited for FSL whereby the target classes are completely different from the source ones: either global or per-class distribution alignment would have a detrimental effect on class separation and discriminativeness. How to achieve domain distribution alignment for DA whilst maintaining source/target per-class discriminativeness thus becomes the key for DA-FSL.

To this end, we propose a domain-adversarial prototypical network (DAPN) to solve the DA-FSL problem. Specifically, based on prototypical learning for few training samples [49], we introduce a novel adversarial learning method for few-shot domain adaptation. Note that domain adversarial learning has been popular among existing UDA methods [12, 19, 54, 30] for global (as opposed to per-class) distribution alignment. Since per-class alignment is the ultimate goal for UDA, its successful use in these UDA methods suggests that, global distribution alignment would indirectly lead to per-class alignment. That is an unwanted effect for our DA-FSL problem as the target classes are different from those of the source. Therefore, in addition to the domain confusion objective commonly used by existing UDA methods for learning a domain-adaptive feature embedding space, new losses are introduced before feature embedding to enforce source/target class discriminativeness. The result is that we would have the better of both worlds: the global distributions of the source and target are aligned to reduce the domain gap for DA; in the meantime, the per-class distribution are not aligned and the source and target classes remain well-separable, benefiting the FSL task. With two sets of losses designed for DA and FSL respectively, to remove the need for weight selection for multiple losses, an adaptive re-weighting module is also introduced to further balance the two objectives.

Our main contributions are three-fold: (1) The DA-FSL problem is formally defined and tackled. For the first time, we address both the few-shot DA and few-shot recognition problems jointly in a unified framework. (2) We propose a novel adversarial learning method to learn feature representation, which is not only domain-confused for domain adaptation but also domain-specific for class separation. (3) Extensive experiments show that our proposed model outperforms the state-of-the-art FSL and domain adaptation models (as well as their naïve combinations).

2. Related Work

Few-Shot Learning. FSL is dominated by meta-learning based methods. They can be organized into three groups: (1) The first group adopts model-based learning strategies [48, 35] that fine-tune the model trained from the source classes and then quickly adapt it to the target classes. (2) The second group [21, 57, 49, 52, 41] focuses on distance metric learning for the nearest neighbor (NN) search. Matching Network (MatchingNet) [57] builds different encoders for the support set and the query set. Prototypical Network (ProtoNet) [49] learns a metric space in which object classification can be performed by computing the distance of a test sample to the prototype representation of each target class. [41] makes improvements over ProtoNet towards a scenario where the unlabeled samples are also available within each episode. Relation Network (RelationNet) [52] recognizes the samples of new/target classes by computing relation scores between query images and the few samples of each new class. (3) The third group [40, 11] chooses to utilize novel optimization algorithms instead of gradient descent to fit in the few-shot regime. Although our model belongs to the second group with ProtoNet as a component, it is designed to address both few-shot DA and few-shot recognition problems (included in DA-FSL) jointly in a unified framework, which has not been studied before.

Domain Adaptation. Note that the domain adaptation problem involved in our DA-FSL setting cannot be solved by supervised domain adaptation (SDA) [34, 1]. Although there exists a small set of labelled samples from the target domain used for DA under our DA-FSL setting, the classes from the target domain have no overlap with the classes from the source domain. Recently, unsupervised domain adaptation (UDA) has dominated the studies on DA. The conventional UDA models [10, 15, 36, 13, 55, 56, 64, 29, 31] typically leverage the subspace alignment technique. Many modern UDA methods [54, 12, 3, 30, 2, 19, 62, 43, 50, 22] resort to adversarial learning [14], which minimizes the distance between the source and target features by a discriminator. However, as mentioned early, even if global domain distribution alignment is enforced, it often leads to per-class alignment which reduces the discriminativeness of the learned feature representation for the FSL task. Moreover, since existing UDA methods still assume that the target domain contains the same classes as the source domain, recent methods focus on per-class cross-domain alignment [47, 32, 5, 25, 46] are unsuitable for our DA-FSL problem. Global domain data distribution alignment [54, 24, 19] is thus adopted in our DAPN with a special mechanism introduced to prevent per-class alignment.

Domain Adaptation + Few-Shot Learning. Note that a cross-domain dataset (*mini*ImageNet [40] $\rightarrow$ CUB [58]) is used for FSL in [4, 53]. However, it is only for evaluating the cross-dataset generalization, rather than developing a new cross-domain FSL method. In contrast, this work focuses on much larger domain change (e.g., natural images vs. cartoon-like ones). Importantly, we develop a novel DA-FSL model to address the problem. Note that a new setting called few-shot domain adaptation (FSDA) is proposed [33]. However, the FSDA setting in [33] is very different from ours: both source and target domains share the same set of classes under the FSDA setting, while the source and target classes have no overlap under our DA-FSL setting. [45] proposes a DA-based FSL setting, but again it is very different from our work: in addition to few labeled samples, [45] assumes the access to a large number of unlabeled samples from the target domain. In contrast, we do not make this assumption. Therefore, the problem setting in [45] is much easier than ours, and designed to exploit unlabeled target domain data, the method in [45] cannot be used here.

3. Methodology

3.1. Problem Definition

Under our DA-FSL setting, we are given a large sample set $\mathcal{D}_s$ from a set of source classes $\mathcal{C}_s$ in a source domain, a few-shot sample set $\mathcal{D}_d$ from a set of target classes $\mathcal{C}_d$ in a target domain, and a test set $\mathcal{T}$ from another set of target classes $\mathcal{C}_t$ in the target domain, where $\mathcal{C}_s \cap \mathcal{C}_d = \emptyset$, $\mathcal{C}_t \cap \mathcal{C}_d = \emptyset$, and $\mathcal{C}_s \cap \mathcal{C}_t = \emptyset$. Our focus is then on training a model with $\mathcal{D}_s$ and $\mathcal{D}_d$ and then evaluating its generalization ability on $\mathcal{T}$. Note that there is also a few-shot sample set $\mathcal{D}_t$ (i.e. the support set) from the set of target classes $\mathcal{C}_t$, which could also be used for model training. However, we follow the FSL methods that do not require finetuning [4] and thus ignore $\mathcal{D}_t$ in the training phase. Due to the domain differences, the data distribution $P_s(x)$ for the set of source classes $\mathcal{C}_s$ is different from that (i.e. $P_t(x)$) for the set of target classes $\mathcal{C}_t \cup \mathcal{C}_d$, where x denotes a sample. Formally, we have $\mathcal{D}_s = \{(x_1, y_1), \ldots, (x_N, y_N) \mid x_i \sim P_s(x), y_i \in \mathcal{C}_s\}$ and $\mathcal{D}_d = \{(x_1, y_1), \ldots, (x_K, y_K) \mid x_i \sim P_t(x), y_i \in \mathcal{C}_d\}$, where y_i denotes the class label of sample x_i. The goal of our DA-FSL is to exploit $\mathcal{D}_s$ and $\mathcal{D}_d$ for training a classifier that can generalize well to $\mathcal{T}$. The proposed DAPN model is illustrated in Fig. 2.

3.2. Few-Shot Learning Module

3.2.1 Episode Training

To simulate the few-shot test process in the training phase, a small amount of data from both $\mathcal{D}_s$ and $\mathcal{D}_d$ are sampled to form episodic training sets. Specifically, we first build training episodes from the large sample set $\mathcal{D}_s$. To form a training episode e_s, we randomly choose N_{sc} classes from $\mathcal{D}_s$ and then build two sets of samples from the N_{sc} classes: the support set S_s consists of $k \times N_{sc}$ samples (k samples per class), and the query set Q_s is composed of samples from the same N_{sc} classes. For an N_{meta}-way k-shot problem, we train our model with an N_{sc}-way k-shot training episode, where $N_{sc} > N_{meta}$, as in [57, 49]. In this work, for 5-way classification and 5-shot learning in the test phase, each training episode is generated with $N_{sc} = 20$ and $k = 5$. In addition to the training episodes from $\mathcal{D}_s$, we also build training episodes from the few-shot sample set $\mathcal{D}_d$. Since *the samples in* $\mathcal{D}_d$ *are scarce* and even cannot form a single training episode, we perform the standard data augmentation method (i.e. horizontal flips and 5 random crops widely used for training existing CNN models) on $\mathcal{D}_d$, and obtain an augmented sample set $\hat{\mathcal{D}}_d$. To form a training episode e_d, we then randomly choose N_{dc} classes from $\hat{\mathcal{D}}_d$ and build two sets of samples from the N_{dc} classes: the support set S_d contains $k \times N_{dc}$ samples with k samples per class, and the query set Q_d is sampled from remainder of the same N_{dc} classes. In this work, we set $N_{dc} = N_{meta}$.

3.2.2 Prototypical Learning

We adopt the idea of prototype learning [49] in our few-shot learning module. We learn a *prototype* of each class in the support set S_s and classifies each sample in the query set Q_s based on the distances between each sample and different prototypes (i.e. the nearest neighbor classifier is used). Specifically, the M-dimensional prototypes are computed through an embedding function $f_\varphi : \mathcal{R}^d \rightarrow \mathcal{R}^M$ with learnable parameters φ. With the embedding function f_φ, the samples are projected from the d-dimensional visual space into an M-dimensional feature space where the samples from the same class are close to each other and the samples from different classes are far away.

Formally, the prototype p_c^s of class c in the support set S_s is defined as the mean vector of the embedded support samples belonging to this class:

$$p_c^s = \frac{1}{|S_c|} \sum_{(x_i, y_i) \in S_c} f_\varphi(x_i), \tag{1}$$

where $S_c = \{(x_i, y_i) : (x_i, y_i) \in S_s, y_i = c\}$ denotes the set of support samples from class c.

Our prototypical network then produces the class distribution of a query sample x based on the softmax output

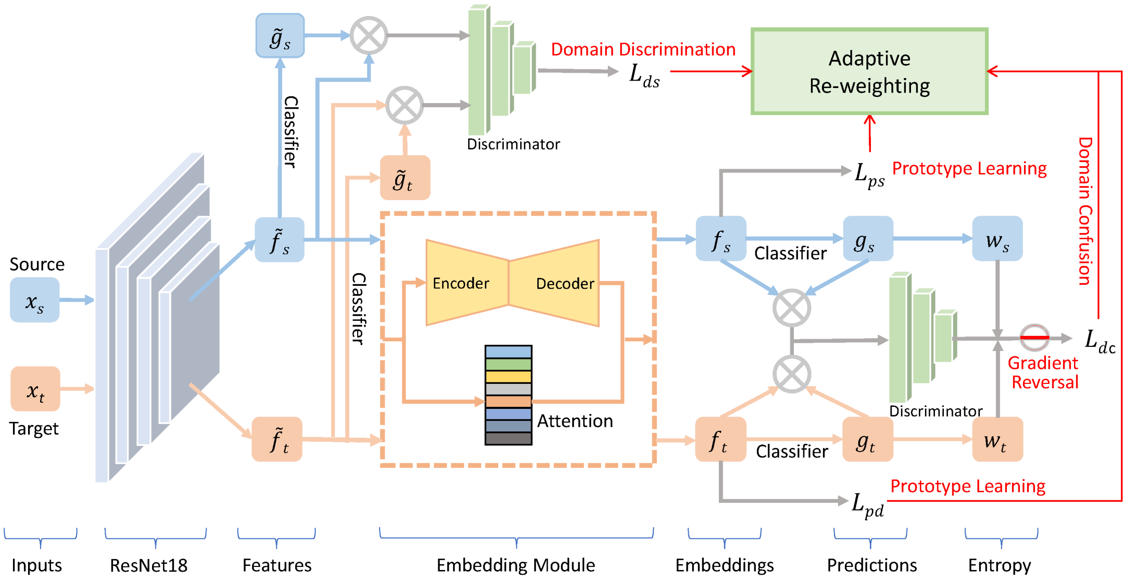

Figure 2. Overview of the proposed DAPN model for DA-FSL. Prototype learning and adversarial-based domain adaptation modules are integrated into a unified framework. We utilize feature embedding in prototype learning, and impose domain discrimination and domain confusion in the spaces before and after embedding, respectively.

w.r.t. the distance between the sample embedding $f_\varphi(x)$ and the class prototype p^s_c as follows:

$$p_\varphi(y=c|x) = \frac{\exp(-\text{dist}(f_\varphi(x), p^s_c))}{\sum_{c'}\exp(-\text{dist}(f_\varphi(x), p^s_{c'}))}, \quad (2)$$

where $\text{dist}(\cdot,\cdot)$ denotes the Euclidean distance in the $\mathcal{R}^M$ space. With the above class distribution, the loss function over each episode e_s is defined based on the negative log-probability of query sample x w.r.t. its true class label c:

$$L_{ps} = \mathbb{E}_{S_s,Q_s}[-\sum_{(x,y)\in Q_s}\log p_\varphi(y=c|x)]. \quad (3)$$

Similarly, the loss function over each episode e_d can be formulated based on the negative log-probability of query sample x w.r.t. its true class label c:

$$L_{pd} = \mathbb{E}_{S_d,Q_d}[-\sum_{(x,y)\in Q_d}\log p_\varphi(y=c|x)]. \quad (4)$$

The above two losses for prototype learning are employed in our proposed DAPN model on the feature output of a domain-adaptive embedding module (see Fig. 2), which is described next.

3.3. Domain Adversarial Adaptation Module

As mentioned, the main objective of domain adaptive module is to learn a feature embedding space where the global distribution of the source and target domains are aligned, while the domain-specific discriminative information is still kept. To this end, we choose to enforce domain discriminativeness and domain alignment learning objectives before and after an embedding module. The task of balancing these two objectives are then handled by an adaptive loss re-weighting module to be described in Sec. 3.4.

3.3.1 Domain Adaptive Embedding

As shown in Fig. 2, the input to the embedding module is the output of a feature extraction CNN (ResNet18 in this work), which represents each sample (image) x as a 512-dimensional feature vector: $\tilde{\text{f}} = \tilde{F}(x)$. The embedding module consists of an autoencoder and an attention sub-module. Concretely, the autoencoder takes $\tilde{\text{f}}$ as input and output an embedding vector $\bar{\text{f}} = \bar{F}(x)$. Moreover, to enforce $\bar{\text{f}}$ to be as domain-confused as possible, we impose an attention sub-module composed of a fully-connected (FC) layer on it: the attention score $\text{sigmoid}(\text{FC}(\bar{\text{f}}))$ is used to remove any domain-specific information (where $\text{FC}(\cdot)$ denotes the output of the FC layer). Combining the autoencoder and attention sub-module together, we have the final output of the embedding module as $\text{f} = F(x)$.

3.3.2 Domain Adaptive Loss

Although both the autoencoder and attention sub-module can implicitly align the two domains, further alignment is needed by introducing domain adaptive losses. Motivated by the superior performance of Conditional Domain Adversarial Network (CDAN) [30] on the domain adaptation task, we define a domain adversarial loss function E on the domain discriminator D across the source distribution $P_s(x)$ and target distribution $P_t(x)$, as well as on the feature representation $\text{f} = F(x)$ after the feature embedding module and the classifier prediction $\mathrm{g} = G(x)$:

$$\begin{aligned}\min_{D}\max_{F,G} E = &- \mathbb{E}_{x_i^s \sim P_s(x)} \log[D(\mathrm{f}_i^s, \mathrm{g}_i^s)] \\ &- \mathbb{E}_{x_j^t \sim P_t(x)} \log[1 - D(\mathrm{f_j}^t, \mathrm{g}_j^t)].\end{aligned} \tag{5}$$

Let $\mathrm{h} = (\mathrm{f}, \mathrm{g})$ be the joint variable of feature representation f and classifier prediction g. Concretely, the multilinear map $T_{\otimes}(\mathrm{h}) = \mathrm{f} \otimes \mathrm{g}$ is chosen to condition D on g, which is defined as the outer product of multiple random vectors. However, multilinear map faces dimension explosion. Let d_{f} and d_{g} denote the dimensions of vectors f and g, respectively. The multilinear map has a dimension of $d_{\mathrm{f}} \times d_{\mathrm{g}}$, which is often too high dimensional to be embedded into deep learning models. To address this dimension explosion problem, the inner-product $T_{\otimes}(\mathrm{f}, \mathrm{g})$ can be approximated by the dot-product $T_{\odot}(\mathrm{f}, \mathrm{g}) = \frac{1}{\sqrt{d}}(R_{\mathrm{f}}\mathrm{f}) \odot (R_{\mathrm{g}}\mathrm{g})$, where $\odot$ is the element-wise product, $R_{\mathrm{f}} \in \mathcal{R}^{d \times d_f}$ and $R_{\mathrm{g}} \in \mathcal{R}^{d \times d_g}$ are two random matrices sampled only once and fixed in the training phase, and $d \ll d_{\mathrm{f}} \times d_{\mathrm{g}}$. Note that each element in R_{f} or R_{g} follows a symmetric distribution with invariance such as the uniform distribution and Gaussian distribution. Finally, we adopt the following conditioning strategy:

$$T(h) = \begin{cases} T_{\otimes}(\mathrm{f}, \mathrm{g}) & \text{if } d_{\mathrm{f}} \times d_{\mathrm{g}} \leq d_{feat} \\ T_{\odot}(\mathrm{f}, \mathrm{g}) & \text{otherwise,} \end{cases} \tag{6}$$

where d_{feat} denotes the dimension of the output of the fully-connected layer. For domain adaptation, we solve an optimization problem derived from Eq. (5):

$$\begin{aligned}\min_{D}\max_{T} E = &- \mathbb{E}_{x_i^s \sim P_s(x)} \log[D(T(\mathrm{h}_i^s))] \\ &- \mathbb{E}_{x_j^t \sim P_t(x)} \log[1 - D(T(\mathrm{h}_j^t))],\end{aligned} \tag{7}$$

where the subproblem of $\max_T E$ is solved by adding a gradient adversarial layer (see Fig. 2) as in [12], and the subproblem of $\min_D E$ is solved with the standard back propagation algorithm.

Note that some samples are easy-to-transfer, while others are hard-to-transfer. If the loss function imposes equal importance for different samples, it could weaken the effectiveness of the learned model. We thus modify the original CDAN [30] formulation by adopting the entropy criterion $H(\mathrm{g}) = -\sum_{c=1}^{C} \mathrm{g}_c \log \mathrm{g}_c$, where C is the number of classes and g_c is the probability of the sample belong to class c. We re-weight training samples by an entropy-aware weight $w(H(\mathrm{g})) = 1 + e^{-H(\mathrm{g})}$ to make easy-to-transfer examples priority to hard ones. The loss for learning domain-confused feature representation is formulated as:

$$\begin{aligned}L_{dc} = &- \mathbb{E}_{x_i^s \sim P_s(x)} w(H(\mathrm{g}_i^s)) \log[D(T(\mathrm{h}_i^s))] \\ &- \mathbb{E}_{x_j^t \sim P_t(x)} w(H(\mathrm{g}_j^t)) \log[1 - D(T(\mathrm{h}_j^t))],\end{aligned} \tag{8}$$

which is illustrated after the embedding module in Fig. 2.

3.3.3 Domain Discriminative Loss

Note that the domain adaptive/confusion loss in Eq. (8) is useful for bridging the domain gap between source and target, but it also has the unwanted side-effect of over-alignment at the per-class level which will harm the FSL performance. To alleviate this problem, we introduce a domain discrimination loss so that the per-class distributions within each domain are different from each other. Note that there is already a domain discriminator for domain alignment after embedding via gradient reversal (see Fig. 2), so it makes little sense to add another on the same embedding space. Instead, our domain discriminative loss is added to the output of the feature extraction CNN. In this way, the features before and after the embedding layer with self-attention are distinguished and confused, respectively. Compared to the vanilla DA model, the powerful ability for class-level feature extraction of the backbone network is enhanced in our proposed model.

Concretely, we first define a conventional classification loss function $\tilde{E}$ on the domain discriminator $\tilde{D}$ across the source distribution $P_s(x)$ and target distribution $P_t(x)$, as well as on the feature representation $\tilde{\mathrm{f}} = \tilde{F}(x)$ before feature embedding and the classifier prediction $\tilde{\mathrm{g}} = \tilde{G}(x)$:

$$\begin{aligned}\min_{\tilde{D},\tilde{F},\tilde{G}} \tilde{E} = &- \mathbb{E}_{x_i^s \sim P_s(x)} \log[\tilde{D}(\tilde{\mathrm{f}}_i^s, \tilde{\mathrm{g}}_i^s)] \\ &- \mathbb{E}_{x_j^t \sim P_t(x)} \log[1 - \tilde{D}(\tilde{\mathrm{f}}_j^t, \tilde{\mathrm{g}}_j^t)].\end{aligned} \tag{9}$$

Let $\tilde{\mathrm{h}} = (\tilde{\mathrm{f}}, \tilde{\mathrm{g}})$. The loss for learning domain-specific feature representation is:

$$\begin{aligned}L_{ds} = &- \mathbb{E}_{x_i^s \sim P_s(x)} \log[\tilde{D}(T(\tilde{\mathrm{h}}_i^s))] \\ &- \mathbb{E}_{x_j^t \sim P_t(x)} \log[1 - \tilde{D}(T(\tilde{\mathrm{h}}_j^t))],\end{aligned} \tag{10}$$

3.4. Adaptive Re-weighting Module

Our DAPN model is trained with multiple objectives mentioned above (i.e. Eqs. (3) (4) (8) (10)), which can be viewed as multi-task learning. Among the losses, the FSL losses in Eqs. (3) (4) and the domain discriminative loss in (10) are pulling in different directions as the domain adaptive loss in (8). This makes it more crucial to balance among them, especially since in different episodes, different recognition tasks are sampled which pose different level of demand for these competing learning objectives. A naïve weighted sum of losses thus does not suffice. More sophisticated adaptive loss re-weighting mechanism is required.

As reported in [20], there exists task-dependent uncertainty in multi-task learning, which stays constant for all input data and varies between different tasks. Therefore, we adopt an adaptive multi-task loss function based on maximizing the Gaussian likelihood with task-dependent uncertainty, in order to determine the weights of the objectives automatically. Let the output of a neural network model with weights $\mathbf{W}$ on input x be denoted as $\mathbf{f}^{\mathbf{W}}(x)$ (with $f_c^{\mathbf{W}}(x)$ be the c-th element of $\mathbf{f}^{\mathbf{W}}(x)$) and the discrete output of the model be denoted as y. We utilize the classification likelihood to squash a scaled version of the model's output with a softmax function as follows:

$$p(\mathrm{y}|\mathbf{f}^{\mathbf{W}}(x)) = \text{softmax}(\mathbf{f}^{\mathbf{W}}(x)). \tag{11}$$

Specifically, with a positive scalar σ, the log likelihood for this output can be formulated as:

$$\log p(\mathrm{y}|\mathrm{f}^{\mathrm{W}}(x), \sigma) = \frac{1}{\sigma^2} f_c^{\mathrm{W}}(x) - \log \sum_{c'} \exp(\frac{1}{\sigma^2} f_{c'}^{\mathrm{W}}(x)). \tag{12}$$

In this work, our DAPN has four discrete outputs $\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_4$, modeled with multiple softmax likelihoods, respectively. The joint loss $L(\mathbf{W}, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$ is:

$$\begin{aligned}
& L(\mathrm{W}, \sigma_1, \sigma_2, \sigma_3, \sigma_4) \\
& = \text{softmax}(\mathrm{y}_1{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_1) \cdot \text{softmax}(\mathrm{y}_2{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_2) \\
& \quad \cdot \text{softmax}(\mathrm{y}_3{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_3) \cdot \text{softmax}(\mathrm{y}_4{=}c; \mathrm{f}^{\mathrm{W}}(x), \sigma_4) \\
& = -\log p(\mathrm{y}_1|\mathrm{f}^{\mathrm{W}}(x), \sigma_1) - \log p(\mathrm{y}_2|\mathrm{f}^{\mathrm{W}}(x), \sigma_2) \\
& \quad - \log p(\mathrm{y}_3|\mathrm{f}^{\mathrm{W}}(x), \sigma_3) - \log p(\mathrm{y}_4|\mathrm{f}^{\mathrm{W}}(x), \sigma_4) \\
& \approx \frac{1}{\sigma_1^2} L_1(\mathrm{W}) + \frac{1}{\sigma_2^2} L_2(\mathrm{W}) + \frac{1}{\sigma_3^2} L_3(\mathrm{W}) + \frac{1}{\sigma_4^2} L_4(\mathrm{W}) \\
& \quad + \log \sigma_1 + \log \sigma_2 + \log \sigma_3 + \log \sigma_4.
\end{aligned}$$

In this paper, the adaptive weights among L_1, L_2, L_3 and L_4 are directly defined as: $w_j = \log \sigma_j^2$ $(j = 1, 2, 3, 4)$. Let $L_1 = L_{ps}$ (see Eq. (3)), $L_2 = L_{pd}$ (see Eq. (4)), $L_3 = L_{dc}$ (see Eq. (8)) and $L_4 = L_{ds}$ (see Eq. (10)). The overall loss of our model is thus formulated as follows:

$$\begin{aligned}
L = {} & w_1/2 + \exp(-w_1) L_s + w_2/2 + \exp(-w_2) L_d \\
& + w_3/2 + \exp(-w_3) L_{dc} + w_4/2 + \exp(-w_4) L_{ds}.
\end{aligned} \tag{13}$$

4. Experiments

4.1. Datasets and Settings

Datasets. (1) ***mini*ImageNet** [40]: This dataset is a subset of ILSVRC-12 [44]. It consists of from 100 classes, with 600 images per class. We follow the widely-used class split as in [40] and adapt it to our domain-adaptive FSL setting: 64 classes for $\mathcal{C}_s$, 16 for $\mathcal{C}_d$, and 20 for $\mathcal{C}_t$. Further, we utilize the style transfer algorithm [63] to transfer the samples from $\mathcal{C}_d$ and $\mathcal{C}_t$ into a new domain. In this work, the samples of the source domain are natural pictures while the samples of the new/target domain are pencil paintings. (2) ***tiered*ImageNet** [41]: This dataset is a larger subset of ILSVRC-12. We use 351 classes for $\mathcal{C}_s$, 97 classes for $\mathcal{C}_d$, and 160 classes for $\mathcal{C}_t$. The same style transfer is performed on the $\mathcal{C}_d$ and $\mathcal{C}_t$ splits of *tiered*ImageNet to form a new domain. (3) **DomainNet** [38]: To generate a new dataset for domain-adaptive FSL, we exploit an existing multi-source domain adaptation dataset, which is the largest UDA dataset until now. There are 275 classes for $\mathcal{C}_s$, 55 classes for $\mathcal{C}_d$, and 70 classes for $\mathcal{C}_t$. In this work, we use the real split in DomainNet as the source domain and the sketch split as the target domain. (4) ***mini*ImageNet → CUB** [58]: The CUB-200-2011 dataset contains 200 classes and 11,788 natural images of birds. Under our DA-FSL setting, we use 100 classes from *mini*ImageNet for $\mathcal{C}_s$, and 50/50 classes from CUB for $\mathcal{C}_d$ and $\mathcal{C}_t$. We use the same class split as in [4]. Overall, the cross-domain settings of the four datasets are given in Table 1, and all images are resized to 84 × 84.

Datasets	$\mathcal{C}_s$ (source)	$\mathcal{C}_d$ (target)	$\mathcal{C}_t$ (target)
*mini*ImageNet [54]	64	16	20
*tiered*ImageNet [19]	351	97	160
DomainNet [62]	275	55	70
mini → CUB [59]	100 (*mini*)	50 (CUB)	50 (CUB)

Table 1. Cross-domain settings of four datasets. The class set $\mathcal{C}_s$ from the source domain and the class set $\mathcal{C}_d$ from the target domain are used for episode training and domain adaptation, and the class set $\mathcal{C}_t$ is for DA-FSL testing. Note that each class in $\mathcal{C}_d$ contains only k samples (see Sec. 3.2 for details).

Evaluations. We make the evaluation on the test set under the 5-way 1-shot and 5-way 5-shot settings, as in previous works. The top-1 accuracy is computed for each test episode, and the average top-1 accuracy is reported over 2,000 test episodes (with 95% confidence intervals).

Baselines. (1) **FSL Baselines**: Representative FSL baselines include relation network [52], MatchingNet [57], S-GM [61], ProtoNet [49], MetaOptNet [26] and Baseline++ [4]. We report the test results under the 5-way 1-shot and 5-way 5-shot settings. (2) **UDA Baselines**: Representative UDA baselines include CDAN [30], ADDA [54], AFN [62], M-ADDA [24], and CyCADA [19]. For testing under the 5-way 1-shot and 5-way 5-shot settings, we first train the CNN backbone with these UDA methods, and then extract the features of test/target samples so that a naïve nearest neighbor classifier can be used to recognize the test/target classes. (3) **UDA+FSL Baselines**: Representative baselines for directly combining UDA and FS-L include CDAN+ProtoNet and CDAN+MetaOptNet (both trained end-to-end). We select the UDA+FSL baselines based on two criteria: 1) UDA baselines are latest/state-of-the-art (e.g. CDAN [30] is state-of-the-art); 2) FSL baselines are representative/state-of-the-art (e.g. ProtoNet [49] is representative and MetaOptNet [26] is state-of-the-art).

Implementation Details. Our model is implemented in PyTorch. The ResNet18 model [18] is used as the backbone for all compared methods. We pretrain the backbone from scratch using the training set and then finetune it to solve the DA-FSL problem. In this work, each training episode is generated with $N_{sc} = 20$ and $k = 1/5$, and the query set for each category contains 15 images. The training process

Model	5-way 1-shot	5-way 5-shot
ADDA [54]	22.83 ± 0.26	29.13 ± 0.43
CyCADA [19]	22.65 ± 0.28	29.36 ± 0.33
AFN [62]	23.83 ± 0.22	32.56 ± 0.30
CDAN [30]	23.82 ± 0.24	31.77 ± 0.28
M-ADDA [24]	23.54 ± 0.29	30.30 ± 0.23
RelationNet [52]	23.87 ± 0.82	33.29 ± 0.96
MatchingNet [57]	23.35 ± 0.64	32.42 ± 0.55
SGM [61]	23.49 ± 0.29	32.67 ± 0.32
ProtoNet [49]	23.23 ± 0.32	32.92 ± 0.41
MetaOptNet [26]	24.53 ± 0.20	33.23 ± 0.63
Baseline++ [4]	24.06 ± 0.46	32.74 ± 0.81
CDAN+ProtoNet	25.36 ± 0.21	35.51 ± 0.25
CDAN+MetaOptNet	25.78 ± 0.23	35.87 ± 0.25
DAPN (ours)	**27.25** ± 0.25	**37.45** ± 0.25

Table 2. Comparative accuracies (%, top-1) with 95% confidence intervals on the test split of ***mini*ImageNet**.

Model	5-way 1-shot	5-way 5-shot
ADDA [54]	25.31 ± 0.31	30.22 ± 0.44
CyCADA [19]	25.28 ± 0.33	32.14 ± 0.33
AFN [62]	25.74 ± 0.24	33.06 ± 0.39
CDAN [30]	25.82 ± 0.30	34.11 ± 0.31
M-ADDA [24]	25.92 ± 0.32	33.56 ± 0.33
RelationNet [52]	24.12 ± 0.84	33.15 ± 0.94
MatchingNet [57]	25.53 ± 0.46	32.59 ± 0.46
SGM [61]	24.03 ± 0.26	33.42 ± 0.31
ProtoNet [49]	23.54 ± 0.33	33.38 ± 0.29
MetaOptNet [26]	25.06 ± 0.33	34.36 ± 0.25
Baseline++ [4]	24.65 ± 0.74	34.29 ± 1.09
CDAN+ProtoNet	26.52 ± 0.23	37.43 ± 0.29
CDAN+MetaOptNet	26.87 ± 0.41	37.79 ± 0.32
DAPN (ours)	**28.47** ± 0.25	**39.90** ± 0.29

Table 3. Comparative accuracies (%, top-1) with 95% confidence intervals on the test split of ***tiered*ImageNet**.

is optimized by stochastic gradient descent (SGD) with a momentum of 0.9 and a weight decay of 0.01 for 100000 iterations. The learning rate is initially set to $\eta_0 = 0.001$, and then adjusted (as in [30]) by $\eta_p = \eta_0(1+\alpha p)^{-\beta}$, where $\alpha = 10$, $\beta = 0.75$, and p is the training progress ranging from 0 to 1. Since the whole framework is trained end-to-end with an adaptive re-weighting module, there are no other free hyper-parameters to tune.

4.2. Main Results

The comparative results under our DA-FSL setting on four datasets are shown in Tables 2, 3, 4 and 5, respectively. We can observe that: (1) On all datasets, our DAPN significantly outperforms the state-of-the-art FSL and UDA methods, because of its ability to tackle both problems. (2) Our DAPN model also clearly performs better than the two UDA+FSL baselines, showing that the naïve combination of UDA and FSL is not as effective as our specifically designed DAPN model for DA-FSL. (3) Interestingly, when combined with a naïve nearest neighbor classifier (for FSL), the performance of existing UDA methods is as good as that of any existing FSL methods. This suggests that solving the domain adaptation problem is the key to our DA-FSL setting. (4) As shown in Table 5, we extend the *mini* → CUB setting in [4] to show the effectiveness of our approach for the DA-FSL setting. By additionally using k-shot data $\mathcal{D}_d$ as the training data, the traditional FSL methods only obtain small boost and achieve comparable performance w.r.t. the DA methods. However, our method outperforms all FSL and DA methods and their simple combinations. (5) Given the same 5-way 5-shot (or 5-way 1-shot) evaluation setting, the test results on the first two datasets are clearly worse than those on the DomainNet and *mini* → CUB datasets. This indicates that the domain gap (induced by style trans-

Model	5-way 1-shot	5-way 5-shot
ADDA [54]	31.14 ± 0.36	45.86 ± 0.48
CyCADA [19]	32.27 ± 0.34	48.11 ± 0.52
AFN [62]	32.78 ± 0.31	50.22 ± 0.49
CDAN [30]	33.55 ± 0.35	51.56 ± 0.34
M-ADDA [24]	31.71 ± 0.35	47.23 ± 0.39
RelationNet [52]	31.98 ± 0.72	51.12 ± 0.58
MatchingNet [57]	32.10 ± 0.73	51.07 ± 0.74
SGM [61]	33.29 ± 0.27	51.42 ± 0.24
ProtoNet [49]	33.66 ± 0.36	51.72 ± 0.34
MetaOptNet [26]	34.50 ± 0.36	51.76 ± 0.52
Baseline++ [4]	34.34 ± 0.77	51.73 ± 0.70
CDAN+ProtoNet	35.10 ± 0.42	52.10 ± 0.42
CDAN+MetaOptNet	35.46 ± 0.36	52.72 ± 0.41
DAPN (ours)	**36.96** ± 0.35	**54.32** ± 0.36

Table 4. Comparative accuracies (%, top-1) with 95% confidence intervals on the test split of **DomainNet**.

Model	$\mathcal{D}_d$	5-way 1-shot	5-way 5-shot
RelationNet [52]		42.91 ± 0.78	57.71 ± 0.73
MatchingNet [57]		45.59 ± 0.81	53.07 ± 0.74
ProtoNet [49]		45.31 ± 0.78	62.02 ± 0.70
Baseline++ [4]		43.04 ± 0.60	62.04 ± 0.76
RelationNet [52]	✓	44.86 ± 0.69	62.34 ± 0.64
MatchingNet [57]	✓	46.03 ± 0.67	57.92 ± 0.70
ProtoNet [49]	✓	45.31 ± 0.74	63.32 ± 0.71
Baseline++ [4]	✓	45.10 ± 0.68	65.86 ± 0.70
ADDA [54]	✓	45.21 ± 0.33	66.03 ± 0.44
CyCADA [19]	✓	45.98 ± 0.35	66.28 ± 0.43
CDAN [30]	✓	45.65 ± 0.52	65.77 ± 0.55
M-ADDA [24]	✓	44.01 ± 0.45	65.17 ± 0.41
CDAN+ProtoNet	✓	46.23 ± 0.49	67.29 ± 0.53
DAPN (ours)	✓	**48.08** ± 0.50	**68.47** ± 0.44

Table 5. Comparative accuracies (%, top-1) with 95% confidence intervals on the test split of ***mini* → CUB**. ✓ denotes whether to use k-shot data $\mathcal{D}_d$ from the target domain for training.

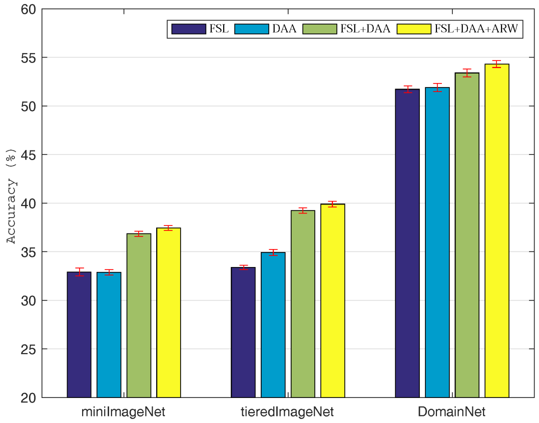

Figure 3. Ablation study results for our full model under the domain-adaptive FSL setting (5-way 5-shot) on the first three datasets. The error-bars show the 95% confidence intervals.

fer) and the category gap (induced by FSL) of the first two datasets are even bigger than those of the widely-used realistic datasets (i.e. DomainNet and *mini* → CUB), which justifies the necessity of using the two synthesized datasets for the novel DA-FSL setting.

4.3. Further Evaluations

Ablation Study on Our Full Model. To demonstrate the contribution of each module of our full model, we make the comparison to its three simplified versions: (1) FSL – only the few-shot learning (FSL) module (described in Sect. 3.2) is used; (2) DAA – the domain adversarial adaptation (DAA) module (described in Sect. 3.3) is combined with a naive nearest neighbor classifier; (3) FSL+DAA – the FSL and DAA modules are combined for domain-adaptive FSL without using adaptive re-weighting. Since our full model combines the two main modules using adaptive re-weighting (ARW), it can be denoted as Full or FSL+DAA+ARW. The ablation study is performed under the 5-way 5-shot domain-adaptive FSL setting. The obtained ablative results are presented in Figure 3. It can be seen that: (1) The performance continuously increases when more modules are used for domain-adaptive FSL, showing the contribution of each module. (2) The improvements achieved by DAA over FSL suggest that the domain adaptation module is important for domain-adaptive FSL and it can perform well even with the naive nearest neighbor classifier. (3) The ARW module clearly yields performance improvements, validating its effectiveness in determining the weights of multiple losses.

Ablation Study on Our DAA Module. We further conduct ablation study to show the contribution of each component of our DAA module. Five methods are compared: (1) FSL – FSL using the two losses L_{ps} defined in Eq. (3) and L_{pd} defined in Eq. (4); (2) FSL+DC – domain-adaptive FSL using the three losses L_{ps}, L_{pd}, and L_{dc} defined in Eq. (8); (3) FSL+DC+DC – based on (2), another domain confusion loss is added to the features before the embedding module. (4) FSL+DC+CLS – based on (2), a classification loss is added to the domain-invariant embedding for class separation. (5) FSL+DC+DS – our domain-adaptive FSL using the four losses L_{ps}, L_{pd}, L_{dc}, and L_{ds} defined in Eq. (10). For a fair comparison, adaptive re-weighting is used for all five methods. The ablative results are shown in Fig. 4. We have the following observations: (1) The significant improvements achieved by FSL+DC over FSL show that domain fusion after the embedding module is essential for our domain-adaptive FSL setting. (2) Imposing domain confusion on CNN features leads to over-alignment at the per-class level that harms the FSL performance. (3) FSL+DC+DS consistently outperforms all alternatives, validating the effectiveness of domain discrimination before the embedding module.

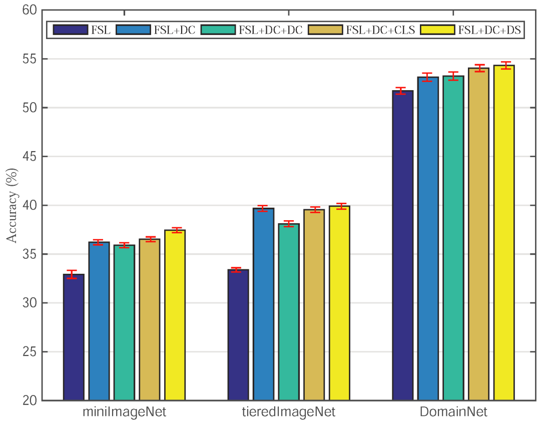

Figure 4. Ablation study results for our DAA module under the domain-adaptive FSL setting (5-way 5-shot) on the first three datasets. The error-bars show the 95% confidence intervals.

5. Conclusion

In this work, we have investigated a new FSL setting called DA-FSL. This challenging FSL setting is closer to some realistic problems where the category gap and domain gap exist at the same time. To simultaneously learn a classifier for new classes with a few shots and bridge the domain gap, we proposed a novel DAPN model by integrating prototypical metric learning and domain adaptation within a unified framework. The domain discriminative and domain confusion learning objectives are introduced before and after a domain-adaptive embedding module, which are further balanced with an adaptive re-weighting module. Extensive experiments showed that our DAPN model outperforms the state-of-the-art FSL and domain adaptation models.

Acknowledgement Zhiwu Lu is partially supported by National Natural Science Foundation of China (61976220 and 61832017), and Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098).

References

[1] Mohammed Abdelwahab and Carlos Busso. Supervised domain adaptation for emotion recognition from speech. In *ICASSP*, pages 5058–5062, 2015.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, pages 3722–3731, 2017.

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NeurIPS*, pages 343–351, 2016.

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[5] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, 2019.

[6] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR*, pages 11672–11681, 2020.

[7] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *AAAI*, pages 10713–10720, 2020.

[8] Mingyu Ding, An Zhao, Zhiwu Lu, Tao Xiang, and Ji-Rong Wen. Face-focused cross-stream network for deception detection in videos. In *CVPR*, pages 7802–7811, 2019.

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003.

[10] B. Fernando, A. H. M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, pages 2960–2967, 2013.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[13] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[15] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006, 2011.

[16] Jiechao Guan, Zhiwu Lu, Tao Xiang, Aoxue Li, An Zhao, and Ji-Rong Wen. Zero and few shot learning with semantic feature synthesis and competitive learning. *TPAMI*, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[19] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle consistent adversarial domain adaptation. In *ICML*, pages 1989–1998, 2018.

[20] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.

[21] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[22] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pages 2452–2460, 2015.

[23] Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. One-shot learning by inverting a compositional causal process. In *NeurIPS*, pages 2526–2534, 2013.

[24] Issam Laradji and Reza Babanezhad. M-ADDA: Unsupervised domain adaptation with deep metric learning. *arXiv preprint arXiv:1807.02552*, 2018.

[25] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019.

[26] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[27] Aoxue Li, Tiange Luo, Zhiwu Lu, Tao Xiang, and Liwei Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *CVPR*, pages 7212–7220, 2019.

[28] Fei-Fei Li, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.

[29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[30] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1640–1650, 2018.

[31] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[32] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2019.

[33] Saeid Motiian, Quinn Jones, Seyed Mehdi Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6673–6683, 2017.

[34] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pages 5715–5725, 2017.

[35] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, pages 2554–2563, 2017.

[36] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *CVPR*, pages 692–699, 2013.

[37] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[38] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. *arXiv preprint arXiv:1812.01754*, 2018.

[39] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *ICLR*, 2020.

[40] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[41] 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 *ICLR*, 2018.

[42] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *NeurIPS*, pages 46–54, 2013.

[43] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *TPAMI*, 41(4):801–814, 2019.

[44] 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. *IJCV*, 115(3):211–252, 2015.

[45] Doyen Sahoo, Hung Le, Chenghao Liu, and Steven C. H. Hoi. Meta-learning with domain adaptation for few-shot learning under domain shift, 2019.

[46] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.

[47] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[48] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

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

[50] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *ECCV*, pages 443–450, 2016.

[51] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, pages 403–412, 2019.

[52] 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 *CVPR*, pages 1199–1208, 2018.

[53] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *ICLR*, 2020.

[54] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017.

[55] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[56] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.

[57] 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.

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

[59] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[60] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020.

[61] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-Shot Learning from Imaginary Data. In *CVPR*, pages 7278–7286, 2018.

[62] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Unsupervised domain adaptation: An adaptive feature norm approach. *arXiv preprint arXiv:1811.07456*, 2018.

[63] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.

[64] Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, pages 1859–1867, 2017.