IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 1, 2020, accepted September 10, 2020, date of publication September 14, 2020,
date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023913

Discriminativeness-Preserved Domain Adaptation for Few-Shot Learning

GUANGZHEN LIU[1] AND ZHIWU LU[1,2], (Member, IEEE)
[1]School of Information, Renmin University of China, Beijing 100872, China
[2]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China

Corresponding author: Zhiwu Lu (zhiwu.lu@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976220 and Grant 61832017.

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, this assumption is often invalid in practice – the target classes could come from a different domain. This poses an additional domain adaptation (DA) challenge with few training samples. In this article, the problem of cross-domain few-shot learning (CD-FSL) is approached, which requires solving FSL and DA in a unified framework. To this end, we propose a novel discriminativeness-preserved domain adaptive prototypical network (DPDAPN) model. It is designed to address a specific challenge in CD-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 while 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 in DPDAPN to alleviate the negative effect of domain alignment on FSL. Extensive experiments show that our DPDAPN outperforms the state-of-the-art FSL and DA models, as well as their naive combinations.

INDEX TERMS Discriminativeness, domain adaptation, few-shot learning, prototypical network.

I. INTRODUCTION

In the past few years, few-shot learning (FSL) [1]–[5] has attracted growing attention. This is because to scale a visual recognition model to thousands of (or even more) categories, the problem of lacking labeled data must be overcome. In particular, most visual recognition models are based on deep convolutional neural networks (CNNs). Training them typically requires hundreds of (or more) samples per class to be collected and annotated. This is often infeasible or even impossible for some rare categories. The goal of FSL is thus to recognize a set of target classes by learning with sufficient labeled samples from source classes but only with a few labeled samples from the target classes.

FSL [6], [7] is often formulated as a transfer learning problem [8] from the source classes to the target classes. 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 can come from a different domain from that of the source classes. For example, the target class samples could be collected by a different imaging device (e.g., mobile phone camera vs. single-lens reflex camera), resulting in different photo styles. In a more extreme case, the source classes could be captured in photos and the target classes 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. Such a problem setting is thus termed cross-domain few-shot learning (CD-FSL).

CD-FSL is a more challenging problem due to the added objective of few-shot domain adaptation (DA). As far as we know, jointly addressing both the few-shot DA and few-shot recognition problems has never been attempted. However, DA on its own, particularly unsupervised DA (UDA), has been studied intensively [9]–[19]. A straightforward solution seems to be combining FSL with an existing DA method.

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

TABLE 1. Comparative accuracies (%, top-1) with 95% confidence intervals under the CD-FSL setting (5-way 1-shot) on the three datasets.

Model	*mini*ImageNet	*tiered*ImageNet	DomainNet
ADDA [9]	23.05 ± 0.17	25.82 ± 0.41	31.79 ± 0.39
CyCADA [14]	23.64 ± 0.41	26.56 ± 0.32	33.41 ± 0.33
AFN [15]	24.44 ± 0.17	26.62 ± 0.31	33.73 ± 0.38
CDAN [12]	25.63 ± 0.27	27.79 ± 0.41	36.78 ± 0.24
M-ADDA [47]	25.71 ± 0.22	28.57 ± 0.35	36.63 ± 0.30
MDD [55]	26.74 ± 0.22	29.16 ± 0.43	37.47 ± 0.45
FSDA [53]	27.84 ± 0.52	29.61 ± 0.27	37.62 ± 0.39
RelationNet [21]	23.91 ± 0.47	24.07 ± 0.73	32.84 ± 0.58
MatchingNet [33]	23.69 ± 0.37	24.61 ± 0.45	33.73 ± 0.63
PPA [60]	23.99 ± 0.39	24.99 ± 0.21	34.01 ± 0.59
SGM [61]	24.44 ± 0.31	25.09 ± 0.23	34.53 ± 0.29
ProtoNet [20]	24.68 ± 0.36	25.01 ± 0.47	35.76 ± 0.23
MetaOptNet [62]	26.06 ± 0.29	25.37 ± 0.22	36.65 ± 0.20
Baseline++ [52]	25.68 ± 0.57	26.30 ± 0.89	36.88 ± 0.46
CDAN+ProtoNet	26.16 ± 0.24	28.06 ± 0.15	37.16 ± 0.30
CDAN+MetaOptNett	26.59 ± 0.24	28.66 ± 0.36	38.16 ± 0.23
MDD+ProtoNet	27.00 ± 0.20	29.58 ± 0.37	38.86 ± 0.34
MDD+MetaOptNet	27.12 ± 0.65	29.88 ± 0.84	39.36 ± 0.42
DPDAPN (ours)	**30.23** ± 1.01	**30.70** ± 0.85	**44.17** ± 0.58

TABLE 2. Comparative accuracies (%, top-1) with 95% confidence intervals under the CD-FSL setting (5-way 5-shot) on the three datasets.

Model	*mini*ImageNet	*tiered*ImageNet	DomainNet
ADDA [9]	30.25 ± 0.46	31.65 ± 0.53	47.70 ± 0.65
CyCADA [14]	31.31 ± 0.39	33.31 ± 0.19	50.49 ± 0.27
AFN [15]	33.03 ± 0.41	35.37 ± 0.44	52.70 ± 0.33
CDAN [12]	33.47 ± 0.32	35.94 ± 0.17	53.33 ± 0.37
M-ADDA [47]	34.68 ± 0.29	36.07 ± 0.27	54.18 ± 0.30
MDD [55]	34.60 ± 0.15	36.55 ± 0.26	54.43 ± 0.27
FSDA [53]	34.84 ± 0.43	37.66 ± 0.62	55.27 ± 0.47
RelationNet [21]	34.23 ± 0.70	33.85 ± 0.59	52.19 ± 0.41
MatchingNet [33]	33.95 ± 0.68	34.30 ± 0.35	53.01 ± 0.34
PPA [60]	35.37 ± 0.46	36.05 ± 0.33	53.22 ± 0.51
SGM [61]	34.83 ± 0.23	35.53 ± 0.27	53.50 ± 0.14
ProtoNet [20]	34.29 ± 0.34	36.46 ± 0.37	53.92 ± 0.49
MetaOptNet [62]	35.43 ± 0.13	36.99 ± 0.25	54.02 ± 0.68
Baseline++ [52]	35.70 ± 0.74	37.29 ± 0.89	54.79 ± 0.51
CDAN+ProtoNet	37.87 ± 0.25	39.16 ± 0.41	55.49 ± 0.60
CDAN+MetaOptNet	38.37 ± 0.21	40.42 ± 0.44	56.40 ± 0.36
MDD+ProtoNet	38.47 ± 0.27	39.91 ± 0.31	58.53 ± 0.19
MDD+MetaOptNet	40.43 ± 0.32	42.27 ± 0.19	59.03 ± 0.21
DPDAPN (ours)	**43.53** ± 0.89	**43.58** ± 0.82	**66.86** ± 0.53

In particular, most existing FSL methods [20]–[23] rely on feature reuse to the target classes in a feature embedding space learned from the source [24]. 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 1&2) because the 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 [25]–[29]), they are intrinsically unsuited for FSL whereby the target classes are completely different from the source classes; either global or per-class distribution alignment would have a detrimental effect on class separation and model discriminativeness. How to achieve domain distribution alignment for DA while maintaining source/target per-class discriminativeness thus becomes the key to CD-FSL.

To this end, we propose a discriminativeness-preserved domain adaptive prototypical network (DPDAPN) to solve the CD-FSL problem. Specifically, in addition to the prototypical network [20] (designed for FSL), we introduce a novel adversarial learning method for few-shot domain adaptation. Note that domain adversarial learning has been popular among existing UDA methods [9], [10], [12], [14] 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. This is an unwanted effect for our CD-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 the source/target class discriminativeness. To define the new losses, an autoencoder-based feature embedding layer is added after the feature extractor (see Figure 1). The end 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 (i.e., domain adaptive); in the meantime, the per-class distribution is not aligned, and the source and target classes remain well separable (i.e. discriminativeness-preserved), benefiting the FSL task. With two sets of losses designed for DA and FSL, to remove the need for weight selection for multiple losses, an adaptive reweighting module is also introduced to further balance the two objectives.

Our main contributions are: (1) The CD-FSL problem is formally defined and addressed. Importantly, this work is the first to jointly address both the few-shot DA and few-shot recognition problems in a unified framework. (2) We propose a novel domain adversarial learning method to learn the feature representation that is not only domain-confused for domain adaptation but also domain-specific for class separation (different losses are also balanced by adaptive reweighting). (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).

II. RELATED WORK

A. FEW-SHOT LEARNING

FSL has been dominated by meta-learning-based methods. They can be organized into three groups: (1) The first group adopts model-based learning strategies [30], [31] that finetune the model trained from the source classes and then quickly adapt it to the target classes. (2) The second group [20]–[22], [32], [33] focuses on distance metric learning for the nearest neighbor (NN) search. The matching network (MatchingNet) [33] builds different encoders for the support set and the query set. The prototypical network (ProtoNet) [20] 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. [22] improved ProtoNet in scenarios where the unlabeled samples are also available within each episode. The relation network (RelationNet) [21] 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 [23], [34] utilizes novel optimization algorithms instead of gradient descent to fit in the few-shot regime. [34] formulated an LSTM-based metalearner model to learn an exact optimization algorithm used to train another neural network classifier in the few-shot regime. Reference [23] proposed a model-agnostic metalearning (MAML) learner, whose weights are updated using the gradient, rather than a learned update rule. Although our DPDAPN 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 CD-FSL) jointly in a unified framework, which has not been studied before.

B. DOMAIN ADAPTATION

Note that the domain adaptation problem involved in our CD-FSL setting cannot be solved by supervised domain adaptation (SDA) [35], [36]. Although there exists a small set of labeled samples from the target domain used for DA under our CD-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 [37]–[45] typically leverage the subspace alignment technique. Many modern UDA methods [9]–[19] resort to adversarial learning [46], which minimizes the distance between the source and target features by a discriminator. However, as mentioned, 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, the more recent methods that focus on per-class cross-domain alignment [25]–[29] are unsuitable for our CD-FSL problem. Thus, global domain data distribution alignment [9], [14], [47] is adopted in our DPDAPN with a special mechanism introduced to prevent per-class alignment. In addition, although the target domain differs significantly from the source domain in both heterogeneous domain adaptation (HDA) [48]–[50] and our CD-FSL, they have a distinct difference: the source and target domains share the same set of classes in HDA, but have two disjoint sets of classes in CD-FSL.

C. DOMAIN ADAPTATION + FEW-SHOT LEARNING

A cross-domain dataset (*mini*ImageNet [34] → CUB [51]) is used for FSL in [52]. 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 changes (e.g., natural images vs. cartoon-like images). Importantly, we develop a novel CD-FSL model to address the problem. Note that a new setting called few-shot domain adaptation (FSDA) is proposed [53]. However, the FSDA setting in [53] is very different from ours in that 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 CD-FSL setting. [54] also proposed a DA-based FSL setting, but again it is very different from our work; in addition to a few labeled samples, [54] assumed 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 [54] is much easier than ours, and designed to exploit unlabeled target domain data, the method in [54] cannot be used here.

III. METHODOLOGY

A. PROBLEM DEFINITION

Under our CD-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 can also be used for model training. However, we follow the FSL methods that do not require finetuning [52] 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 CD-FSL is to exploit $\mathcal{D}_s$ and $\mathcal{D}_d$ to train a classifier that can generalize well to the test set $\mathcal{T}$.

The proposed DPDAPN model is illustrated in Figure 1. Various modules in the network are designed for few-shot learning, domain adaptation, and adaptive reweighting to balance the two main objectives. They are introduced in detail in the next three subsections.

B. FEW-SHOT LEARNING MODULE

1) EPISODE TRAINING

To simulate the few-shot test process in the training phase, a small quantity 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 [20], [33]. For example, if we perform 5-way classification and 5-shot learning in the test phase, each training episode can be 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 $\mathcal{D}_d$.

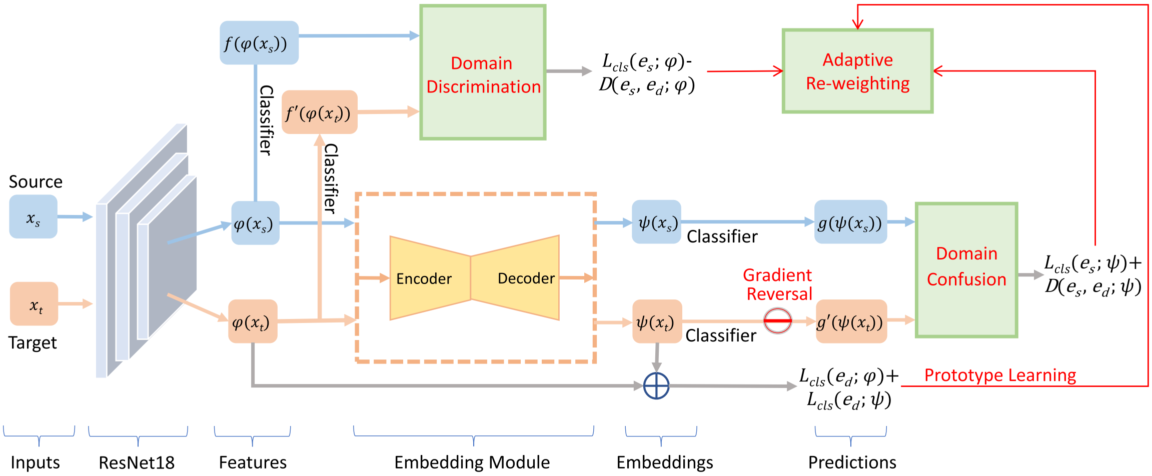

FIGURE 1. Overview of the proposed DPDAPN model for CD-FSL. Both source/target domain confusion and domain discrimination are explicitly included in our DPDAPN model. For simplicity, the weights of multiple losses are not shown here (they can be determined by adaptive reweighting).

Since *the samples in* $\mathcal{D}_d$ *are scarce* and 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 $\widehat{\mathcal{D}}_d$. To form a training episode e_d, we then randomly choose N_{dc} classes from $\widehat{\mathcal{D}}_d$ 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 the remainder of the same N_{dc} classes. In this work, we simply set $N_{dc} = N_{meta}$.

2) PROTOTYPICAL NETWORK

The prototypical network [20] is selected as the main FSL component in our model because it is simple yet remains very competitive [52]. It learns 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 $\psi(x)$. Moreover, with ψ, the samples are projected 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:

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

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.

The prototypical network then produces the class distribution of a query sample x based on the softmax output w.r.t. the distance between the sample embedding $\psi(x)$ and the class prototype p_c^s as follows:

$$p(y = c|x) = \frac{\exp(-\text{dist}(\psi(x), p_c^s))}{\sum_{c'} \exp(-\text{dist}(\psi(x), p_{c'}^s))}, \tag{2}$$

where dist(·, ·) 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:

$$L_{cls}(e_s; \psi) = \mathbb{E}_{(x,y)\in Q_s}[-\log p(y = c|x)]. \tag{3}$$

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:

$$L_{cls}(e_d; \psi) = \mathbb{E}_{(x,y)\in Q_d}[-\log p(y = c|x)]. \tag{4}$$

C. DOMAIN ADVERSARIAL ADAPTATION MODULE

As mentioned before, the main objective of the 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 retained. 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 is then handled by an adaptive loss reweighting module, which is described in Sec. III-D.

1) FEATURE EMBEDDING

As shown in Figure 1, the input to the feature embedding module is the output of a feature extraction CNN (i.e., ResNet18 in this work), which represents each sample (image) x as a 512-dimensional feature vector $\varphi(x)$. In this work, the feature embedding module is simply an autoencoder. Specifically, its encoder has two fully connected (FC) layers: {FC layer (512, 256)}, {FC layer (256, 128)}, and its decoder also has two FC layers: {FC layer (128, 256)}, {FC layer (256, 512)}. The autoencoder takes $\varphi(x)$ as input and outputs an embedding vector $\psi(x)$. The final output of the feature embedding module is $\psi(x)$ (i.e., ψ includes both the feature extractor and autoencoder).

2) DOMAIN ADAPTIVE LOSS

After the autoencoder module, domain alignment is needed by introducing domain adaptive losses. In this work, motivated by the superior performance of domain adaptation with margin disparity discrepancy (MDD) [55], we define an MDD-based domain adversarial loss function across the source domain and the target domain.

Formally, during each training iteration, we are given a pair of training episodes: $e_s = \{S_s, Q_s\}$ from the source domain and $e_d = \{S_d, Q_d\}$ from the target domain. Let $g : \mathbb{R}^M \rightarrow \mathbb{R}^{N_{sc}}$ denote the scoring function constructed from S_s within the source episode e_s. In this work, g is determined by the prototypical network (see Eq. (2)): $g(\psi(x); c) = -\text{dist}(\psi(x), p_c^s)$, where $g(\psi(x); c)$ is the c-th element of $g(\psi(x))$. In addition to the scoring function g, we also introduce an auxiliary scoring function $g' : \mathbb{R}^M \rightarrow \mathbb{R}^{N_{sc}}$ sharing the same hypothesis space with g. Since g is used to score each sample in Q_s on the N_{sc} source classes, g' is designed as a metric-learning network that computes the similarity scores of query-prototype pairs. We set g' to be a multilayer perceptron (MLP) module (see its detailed architecture in Sec. IV-A) stacked after the absolute difference between a query sample and a source class prototype (i.e., the mean representation of support samples from this source class). Our domain adversarial learning objective for learning domain-confused feature representation is formulated as follows:

$$\min_{\psi,\, g} \max_{g'} \quad L_{cls}(e_s; \psi) + \lambda_{dc} D(e_s, e_d; \psi), \tag{5}$$

where λ_{dc} is the trade-off coefficient between the few-shot classification loss $L_{cls}(e_s; \psi)$ and the DA loss $D(e_s, e_d; \psi)$. In this work, the DA loss is defined by the recent margin disparity discrepancy (MDD) [55]. We then have:

$$\begin{aligned} L_{cls}(e_s; \psi) &= \mathbb{E}_{(x_s, y_s) \in Q_s} L(y_s, g(\psi(x_s))), \qquad (6) \\ D(e_s, e_d; \psi) &= \text{disp}_{e_d}(g, g') - \gamma \text{disp}_{e_s}(g, g') \\ &= \mathbb{E}_{(x_t, y_t) \in Q_d} L'(g(\psi(x_t)), g'(\psi(x_t))) \\ &\quad - \gamma \mathbb{E}_{(x_s, y_s) \in Q_s} L(g(\psi(x_s)), g'(\psi(x_s))), \qquad (7) \end{aligned}$$

where γ is a positive hyperparameter, and $\text{disp}_{e_s}(g, g')$ and $\text{disp}_{e_d}(g, g')$ are the two margin disparities of the source and target episodes, respectively. We train g' to maximize the distribution discrepancy between the two episodes and train ψ, g to minimize the maximum MDD, according to Eq. (5). In this minimax manner, the domain gap between the two episodes caused by their disjoint sets of classes is reduced.

With the softmax function $\sigma_j(\mathbf{v}) \triangleq \frac{\exp(v_j)}{\sum_{j'=1}^{N_{sc}} \exp(v_{j'})}$ $(\mathbf{v} \in \mathbb{R}^{N_{sc}}, j = 1, \cdots, N_{sc})$, the loss $L(\cdot, \cdot)$ used in Eqs. (6)–(7) is defined as the cross-entropy loss:

$$L(y_s, g(\psi(x_s))) = -\log[\sigma_{y_s}(g(\psi(x_s)))], \tag{8}$$

$$L(g(\psi(x_s)), g'(\psi(x_s))) = -\sum_{j=1}^{N_{sc}} \sigma_j(g(\psi(x_s))) \log[\sigma_j(g'(\psi(x_s)))]. \tag{9}$$

Similarly, the loss $L'(\cdot, \cdot)$ used in Eq. (7) is defined as a modified cross-entropy loss:

$$L'(g(\psi(x_t)), g'(\psi(x_t))) = \sum_{j=1}^{N_{sc}} \sigma_j(g(\psi(x_t))) \log[1 - \sigma_j(g'(\psi(x_t)))], \tag{10}$$

which was introduced in [46] to ease the burden of vanishing or exploding gradients for adversarial learning.

Note that in Eq. (10), although x_t from Q_d does not belong to any class in the source episode e_s, the similarity scores after softmax $\sigma_j(g(\psi(x_t)))$ and $\sigma_j(g'(\psi(x_t)))$ $(j = 1, \cdots, N_{sc})$ can be considered to come from distributions in an N_{sc}-dimensional space. This is also the reason why we use the binary cross-entropy loss in both Eq. (9) and Eq. (10). Moreover, since g is determined by the meta-learning-based FSL method and it may contain no learnable parameters (e.g., prototypical networks [20] used the negative Euclidean distance as the score), we cut off the gradients over g in Eq. (7) and directly train ψ to minimize this discrepancy loss through a gradient reversal layer (GRL) [10].

3) DOMAIN DISCRIMINATIVE LOSS

Note that the domain adaptive/confusion loss in Eq. (5) is useful for bridging the domain gap between the source and target, but it also has the unwanted side-effect of overalignment 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 Figure 1), so it makes little sense to add another on the same embedding space. Instead, our domain discriminative loss is added to the output (i.e., $\varphi(x) \in \mathbb{R}^M$) of the feature extraction CNN. The main idea is to define the domain discriminative loss with the negative margin disparity discrepancy (MDD) [55]. By minimizing this loss, our model can enhance the source/target per-class separation before domain adaptive feature embedding learning and alleviate the negative effect of domain alignment on FSL.

Specifically, during each training iteration, we still have a pair of training episodes: $e_s = \{S_s, Q_s\}$ from the source domain and $e_d = \{S_d, Q_d\}$ from the target domain. Let $f : \mathbb{R}^M \rightarrow \mathbb{R}^{N_{sc}}$ denote the scoring function constructed from S_s within the source episode e_s. In this work, f is determined by the prototypical network (similar to Eq. (2)): $f(\varphi(x); c) = -\text{dist}(\varphi(x), p_c^s)$, where $f(\varphi(x); c)$ is the c-th element of $f(\varphi(x))$. In addition to the scoring function f, we also introduce an auxiliary scoring function $f' : \mathbb{R}^M \rightarrow \mathbb{R}^{N_{sc}}$, which shares the same hypothesis space with f. Since f is used to score each sample in Q_s on the N_{sc} source classes, f' is designed as a metric-learning network that computes the similarity scores of query-prototype pairs. We set f' to be a multilayer perceptron (MLP) module (see its detailed architecture in Sec. IV-A) stacked after the absolute difference between a query sample and a source class prototype (i.e., the mean representation of its support samples). Our domain discriminative learning objective for learning domain-specific feature representation is given by:

$$\min_{\varphi, f, f'} \quad L_{cls}(e_s; \varphi) - \lambda_{ds} D(e_s, e_d; \varphi), \tag{11}$$

where λ_{ds} is the trade-off coefficient between the few-shot classification loss $L_{cls}(e_s; \varphi)$ and the domain discriminative loss $-D(e_s, e_d; \varphi)$. Note that $L_{cls}(e_s; \varphi)$ and $D(e_s, e_d; \varphi)$ can be similarly computed according to Eq. (6) and Eq. (7), respectively. The only difference is that the embedding function ψ in Eq. (6) and Eq. (7) is replaced by φ.

D. ADAPTIVE REWEIGHTING MODULE

Our DPDAPN model is trained with multiple objectives mentioned above (i.e., Eqs. (4) (5) (11)), which can be viewed as multitask learning. Among the losses, the FSL loss in Eq. (4) and the domain discriminative loss in (11) are pulling in different directions than the domain adaptive loss in (5). This makes it more crucial to balance among them, especially since in different episodes, different recognition tasks are sampled, which pose different levels of demand for these competing learning objectives. A naïve weighted sum of losses thus does not suffice. A more sophisticated adaptive loss reweighting mechanism is required.

As reported in [56], there exists task-dependent uncertainty in multitask learning, which stays constant for all input data and varies between different tasks. Therefore, we adopt an adaptive multitask loss function based on maximizing the Gaussian likelihood with task-dependent uncertainty 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 $f_c^{\mathbf{W}}(x)$ being 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:

$$p(\mathrm{y}|\mathbf{f}^{\mathbf{W}}(x)) = \mathrm{softmax}(\mathbf{f}^{\mathbf{W}}(x)). \tag{12}$$

Specifically, with a positive scalar σ, the log likelihood for this output is:

$$\log p(\mathrm{y} = c|\mathrm{f}^{\mathrm{W}}(x), \sigma) = \frac{1}{\sigma^2} f_c^{\mathrm{W}}(x) - \log \sum_{c'} \exp(\frac{1}{\sigma^2} f_{c'}^{\mathrm{W}}(x)). \tag{13}$$

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

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

The adaptive weights among L_1, L_2, L_3 and L_4 are directly defined as: $w_j = \log \sigma_j^2$ $(j = 1, 2, 3, 4)$. Moreover, by combining Eqs. (4) (5) (11) for model training, we have: $L_1 = L_{cls}(e_s; \psi) + L_{cls}(e_s; \varphi)$, $L_2 = L_{cls}(e_d; \psi) + L_{cls}(e_d; \varphi)$, $L_3 = D(e_s, e_d; \psi)$, and $L_4 = -D(e_s, e_d; \varphi)$. The overall loss of our DPDAPN model is thus formulated as follows:

$$\begin{aligned}
L = {} & w_1/2 + \exp(-w_1)L_1 + w_2/2 + \exp(-w_2)L_2 \\
& + w_3/2 + \exp(-w_3)L_3 + w_4/2 + \exp(-w_4)L_4.
\end{aligned} \tag{14}$$

IV. EXPERIMENTS

A. DATASETS AND SETTINGS

1) DATASETS

Three datasets are used for performance evaluation: (1) ***mini*ImageNet** [34]: This dataset is a subset of ILSVRC-12 [57]. It consists of 100 classes, and all images are of the size 84 × 84. We follow the widely used class split as in [34] and adapt it to our CD-FSL setting: 64 classes for $\mathcal{C}_s$ (with 600 images per class), 16 classes for $\mathcal{C}_d$ (with only k images per class), and 20 classes for $\mathcal{C}_t$ (with only k labeled images per class to form the support set, and the other to form the test set). In this work, we set $k = 1$ or 5. Furthermore, we utilize the style transfer algorithm [58] to transfer the samples from $\mathcal{C}_d$ and $\mathcal{C}_t$ into a new domain. Specifically, the samples of the source domain are natural pictures, while the samples of the new/target domain are pencil paintings. (2) ***tiered*ImageNet** [22]: This dataset is also a subset of ILSVRC-12, but it is larger than *mini*ImageNet. We use 351 classes for $\mathcal{C}_s$ (with an average of 1,278 images per class), 97 classes for $\mathcal{C}_d$ (with only k images per class), and 160 classes for $\mathcal{C}_t$. All images are also of the size 84 × 84. 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** [59]: To generate a new realistic dataset for CD-FSL, we exploit an existing multisource domain adaptation dataset, which is the largest domain adaptation dataset. There are 275 classes for $\mathcal{C}_s$ (with an average of 516 images per class), 55 classes for $\mathcal{C}_d$ (with only k images per class), and 70 classes for $\mathcal{C}_t$. In this work, we take the real photo domain in DomainNet as the source domain and the sketch domain as the target domain. Each image is scaled to the size 84 × 84. For each dataset, examples from the target domain are shown in Figure 2.

2) EVALUATIONS

We make evaluation over the test set under the 5-way 1/5-shot settings, as in previous works. The top-1 accuracy

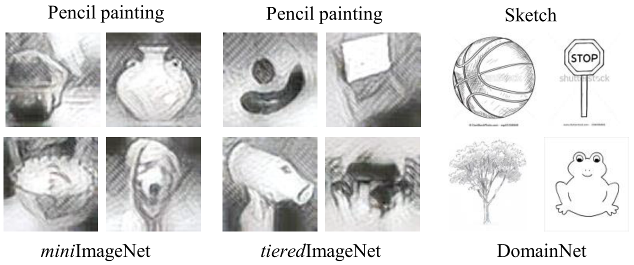

FIGURE 2. Examples from the target domain for the three datasets. In each dataset, the source domain contains real/natural images.

is computed for each test episode, and the average top-1 accuracy is reported over 2,000 test episodes (with 95% confidence intervals).

3) BASELINES

Four groups of baselines are selected: (1) **FSL Baselines**: Representative FSL baselines include RelationNet [21], MatchingNet [33], PPA [60], SGM [61], ProtoNet [20], MetaOptNet [62] and Baseline++ [52]. We report the test results under 5-way 1/5-shot. (2) **UDA Baselines**: Representative UDA baselines based on global domain-level alignment rather than local class-level alignment are chosen. These include CDAN [12], ADDA [9], AFN [15], M-ADDA [47], CyCADA [14], and MDD [55]. For testing under 5-way 1/5-shot, we first train the CNN backbone with these UDA methods and then extract the features of the test/target samples so that a naïve nearest neighbor classifier can be used to recognize the test/target classes. (3) **FSDA Baseline**: FSDA [53] is applied to our CD-FSL setting. For testing under 5-way 1/5-shot, we first train the CNN backbone with FSDA and then extract the features of the test/target samples so that a naïve nearest-neighbor classifier can be used to recognize the test/target classes. (4) **UDA+FSL Baselines**: Representative baselines for directly combining UDA and FSL include CDAN+ProtoNet, CDAN+MetaOptNet, MDD+ProtoNet, and MDD+MetaOptNet, which are all trained end-to-end. We select the UDA+FSL baselines based on two criteria: 1) UDA baselines are representative/state-of-the-art (e.g., CDAN [12] is representative and MDD [55] is state-of-the-art); 2) FSL baselines are representative/state-of-the-art (e.g., ProtoNet [20] is representative and MetaOptNet [62] is state-of-the-art).

4) IMPLEMENTATION DETAILS

Our DPDAPN model is implemented in PyTorch. The ResNet18 model [63] is used as the backbone (which is also used for all compared methods). We pretrain the backbone from scratch using the training set and then finetune it to solve the CD-FSL problem. The auxiliary scoring function g' used in Eq. (5) (or f' used in Eq. (11)) is formed by 4 fully connected (FC) layers: {FC layer (512, 1024), batch normalization, ReLU, dropout(0.5)}, {FC layer (1024, 1024), ReLU, dropout(0.5)}, {FC layer (1024, 64), ReLU}, {FC layer (64, 1)}. In this work, the end-to-end training process is implemented by using backpropagation and stochastic gradient descent. The learning rate is initially set to $\eta_0 = 0.001$ and then is adjusted (as in [12]) 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. A momentum of 0.9 and a weight decay of 0.01 are also selected for training. The code and datasets will be released soon.

B. MAIN RESULTS

The comparative results under our CD-FSL setting on the three datasets are shown in Tables 1 and 2. We make the following observations: (1) On all datasets, our DPDAPN significantly outperforms the state-of-the-art FSL and UDA methods because of its ability to address both problems. (2) Our DPDAPN model also clearly performs better than the four UDA+FSL baselines, showing that the naïve combination of UDA and FSL is not as effective as our specifically designed DPDAPN model for CD-FSL. (3) When combined with a naïve nearest-neighbor classifier (for FSL), the performance of the existing UDA methods is as good as that of any existing FSL methods. This result suggests that solving the domain adaptation problem is the key to our CD-FSL setting. (4) Our DPDAPN model significantly outperforms FSDA [53] in most cases, demonstrating the importance of jointly addressing both the few-shot DA and few-shot recognition problems in a unified framework. (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 DomainNet. This finding indicates that the domain gap (induced by style transfer) and the category gap (induced by FSL) of the first two datasets are even larger than those of the widely used realistic dataset – DomainNet. This result justifies the inclusion of these two synthesized datasets for performance evaluation under the CD-FSL setting.

C. FURTHER EVALUATIONS

1) ABLATION STUDY ON OUR FULL MODEL

To demonstrate the contribution of each module of our full DPDAPN model, we compare it with its three simplified versions: (1) FSL – only the few-shot learning (FSL) module (described in Section III-B) is used, (2) DAA – the domain adversarial adaptation (DAA) module (described in Section III-C) is combined with a naïve nearest neighbor classifier, and (3) FSL+DAA – the FSL and DAA modules are combined for CD-FSL without using adaptive reweighting. Since our full model combines the two main modules using adaptive reweighting (ARW), it can be denoted as Full or FSL+DAA+ARW. The ablation study is performed under the 5-way 5-shot CD-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 to solve the CD-FSL problem, demonstrating the contribution of each module. (2) The improvements achieved by DAA over the classical FSL suggest that the domain adaptation module is important for the CD-FSL setting and can

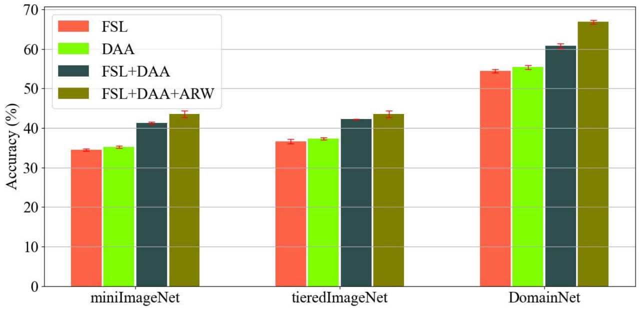

FIGURE 3. Ablation study results for our full model under the CD-FSL setting (5-way 5-shot) on the three datasets. The error bars show the 95% confidence intervals.

perform well even with the naïve nearest neighbor classifier. (3) The ARW module clearly yields performance improvements, validating its effectiveness in determining the weights of multiple losses.

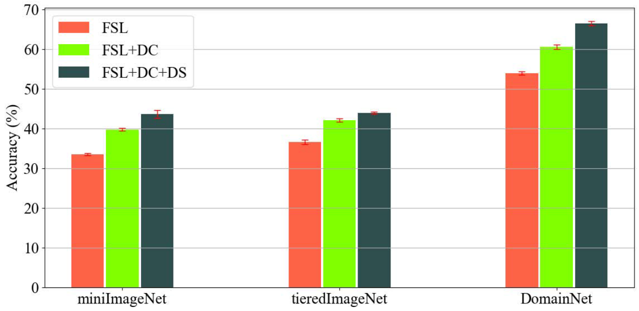

FIGURE 4. Ablation study results for our DAA module under the CD-FSL setting (5-way 5-shot) on the three datasets. The error bars show the 95% confidence intervals.

2) ABLATION STUDY ON OUR DAA MODULE

We further conduct an ablation study to show the contribution of each component of our DAA module. Three methods are compared: (1) FSL – FSL using only the two losses L_1 and L_2 from Eq. (14); (2) FSL+DC – CD-FSL using the three losses L_1, L_2, and L_3 from Eq. (14); and (3) FSL+DC+DS – CA-FSL using the four losses L_1, L_2, L_3, and L_4 from Eq. (14). For a fair comparison, adaptive reweighting is used for all three methods. The ablative results on the three datasets are shown in Figure 4. We make two observations: (1) The significant improvements achieved by FSL+DC over FSL show that domain confusion after the embedding module is extremely important for our CD-FSL setting. (2) FSL+DC+DS consistently outperforms FSL+DC, validating the effectiveness of domain discrimination before the embedding module.

V. CONCLUSION

In this work, we investigated the challenging CD-FSL setting. To simultaneously learn a classifier for new classes with a few shots and bridge the domain gap, we proposed a novel DPDAPN model by integrating prototypical metric learning and domain adaptation within a unified framework. The domain discriminative and domain confusion learning objectives were introduced before and after a domain adaptive embedding module and were further balanced with an adaptive reweighting module. Extensive experiments showed that our DPDAPN model outperforms the state-of-the-art FSL and domain adaptation models.

REFERENCES

[1] L. Fe-Fei, Fergus, and Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1134–1141.

[2] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[3] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2526–2534.

[4] J. Guan, Z. Lu, T. Xiang, and J.-R. Wen, "Few-shot learning as domain adaptation: Algorithm and analysis," in *Proc. ICML*, 2020, pp. 1–8.

[5] N. Fei, Z. Lu, Y. Gao, J. Tian, T. Xiang, and J.-R. Wen, "Meta-learning across meta-tasks for few-shot learning," in *Proc. ECCV*, 2020, pp. 1–12.

[6] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 46–54.

[7] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.

[10] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[11] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 343–351.

[12] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1640–1650.

[13] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3722–3731.

[14] A. Mathur, A. Isopoussu, F. Kawsar, N. B. Berthouze, and N. D. Lane, "FlexAdapt: Flexible cycle-consistent adversarial domain adaptation," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019, pp. 1989–1998.

[15] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," 2018, *arXiv:1811.07456*. [Online]. Available: http://arxiv.org/abs/1811.07456

[16] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, Apr. 2019.

[17] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. ECCV*, 2016, pp. 443–450.

[18] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2452–2460.

[19] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 39, 2020, doi: 10.1109/TPAMI.2020.2991050.

[20] J. Wang and Y. Zhai, "Prototypical siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 4077–4087.

[21] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[22] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *Proc. ICLR*, 2018, pp. 1–15.

[23] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.

[24] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," in *Proc. ICLR*, 2020, pp. 1–21.

[25] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.

[26] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2507–2516.

[27] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9944–9953.

[28] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10285–10295.

[29] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. ICCV*, 2019, pp. 8050–8058.

[30] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," 2016, *arXiv:1605.06065*. [Online]. Available: http://arxiv.org/abs/1605.06065

[31] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. ICML*, 2017, pp. 2554–2563.

[32] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, vol. 2, 2015, pp. 1–5.

[33] O. Vinyals, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.

[34] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. ICLR*, 2017, pp. 1–11.

[35] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5715–5725.

[36] M. Abdelwahab and C. Busso, "Supervised domain adaptation for emotion recognition from speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5058–5062.

[37] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.

[38] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.

[39] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 692–699.

[40] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[41] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474

[42] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5018–5027.

[43] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1859–1867.

[44] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," 2015, *arXiv:1502.02791*. [Online]. Available: http://arxiv.org/abs/1502.02791

[45] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[47] I. Laradji and R. Babanezhad, "M-ADDA: Unsupervised domain adaptation with deep metric learning," 2018, *arXiv:1807.02552*. [Online]. Available: http://arxiv.org/abs/1807.02552

[48] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[49] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2144–2155, Jun. 2019.

[50] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6103–6115, Dec. 2019.

[51] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[52] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. ICLR*, 2019, pp. 1–17.

[53] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 6673–6683.

[54] D. Sahoo, H. Le, C. Liu, and S. C. H. Hoi, "Meta-learning with domain adaptation for few-shot learning under domain shift," in *Proc. ICLR Conf. Blind Submission*, 2019. [Online]. Available: https://openreview.net/pdf?id=ByGOuo0cYm

[55] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. ICML*, 2019, pp. 7404–7413.

[56] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[57] O. Russakovsky, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[58] H. Zhang and K. Dana, "Multi-style generative network for real-time transfer," 2017, *arXiv:1703.06953*. [Online]. Available: http://arxiv.org/abs/1703.06953

[59] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," 2018, *arXiv:1812.01754*. [Online]. Available: http://arxiv.org/abs/1812.01754

[60] S. Qiao, C. Liu, W. Shen, and A. Yuille, "Few-shot image recognition by predicting parameters from activations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7229–7238.

[61] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7278–7286.

[62] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

GUANGZHEN LIU received the M.Eng. degree in computer science from the Renmin University of China, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree in computer science. His research interests include machine learning and computer vision.

ZHIWU LU (Member, IEEE) received the M.S. degree in applied mathematics from Peking University, in 2005, and the Ph.D. degree in computer science from the City University of Hong Kong, in 2011. He is currently a Full Professor with the School of Information, Renmin University of China. He has published over 70 papers in international journals and conference proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), the *International Journal of Computer Vision (IJCV)*, the IEEE Transactions on Image Processing (TIP), ICML, NeurIPS, CVPR, ICCV, and ECCV. His research interests include machine learning, pattern recognition, and computer vision.