CROSS-DOMAIN CROSS-SET FEW-SHOT LEARNING VIA LEARNING COMPACT AND ALIGNED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot learning (FSL) aims to recognize novel query examples with a small support set through leveraging prior knowledge learned from a large-scale training set. In this paper, we extend this task to a more practical setting where the domain shift exists between the support set and query examples, and additional unlabeled data in the target domain can be adopted in the meta-training stage. Such new setting, termed cross-domain cross-set FSL (CDSC-FSL), requires the learning system not only to adapt to new classes with few examples but also to be consistent between different domains. To address this paradigm, we propose a novel approach, namely *stab*PA, to learn prototypical compact and cross-domain aligned representations, so that domain shift and few-shot adaptation can be addressed simultaneously. We evaluate our approach on two new CDCS-FSL benchmarks adapted from the DomainNet and Office-Home datasets, respectively. Remarkably, our approach outperforms multiple elaborated baselines by a large margin and improves 5-shot accuracy by up to 4.7 points.

1 INTRODUCTION

Learning new concepts from a very limited number of images is easy for human beings, however, it is quite difficult for most machine learning algorithms, as they usually need plenty of labeled data to model large intra-class variance and complex inter-class relationships. To bridge the gap between humans and machines, few-shot learning (FSL) has recently been proposed, which aims to learn new classes with only a few examples.

A typical FSL paradigm is to first train a base model with a large labeled dataset (called the base set), which is termed the meta-training stage. When deployed in the meta-testing stage, the base model is adapted to new classes with only a few examples (named the support set), and then tested with a query set covering these novel classes. Despite recent progress of FSL (Kim et al., 2020; Tian et al., 2020), most studies follow a single domain assumption, where the base set, support set and query set are all from the same domain. Cross-domain FSL (Tseng et al., 2020) breaks this assumption and considers the domain shift problem between the meta-training stage and meta-testing stage. However, it is still assumed that the support set and query set of novel classes are from the same domain, which is unrealistic in some practical applications. For example, in smart healthcare scenarios, someone may upload a skin picture captured by cell-phone for querying possible skin diseases, while the support samples are usually a few high-quality medical images captured by professional dermoscope devices. Thus, it is a very valuable problem to study the domain shift between the query set and the support set (Gu et al., 2019). Such case can also be found in face recognition and image retrieval areas (He et al., 2018; Liu et al., 2019).

In this work, we aim to deal with a more practical FSL setting termed cross-domain cross-set FSL (CDCS-FSL). The difference between the CDCS-FSL and previous FSL settings is illustrated in Figure 1 (a). Specifically, instead of assuming a consistent domain in the meta-testing stage,we hope that the few-shot learner of new classes can be learned and tested in different domains, e.g. the support set is from the same source domain of the base set while the query set is from a different target domain, or vice versa. Compared with previous cross-domain FSL, this setting is more challenging as it requires learning a well-aligned feature space shared by the source domain and the

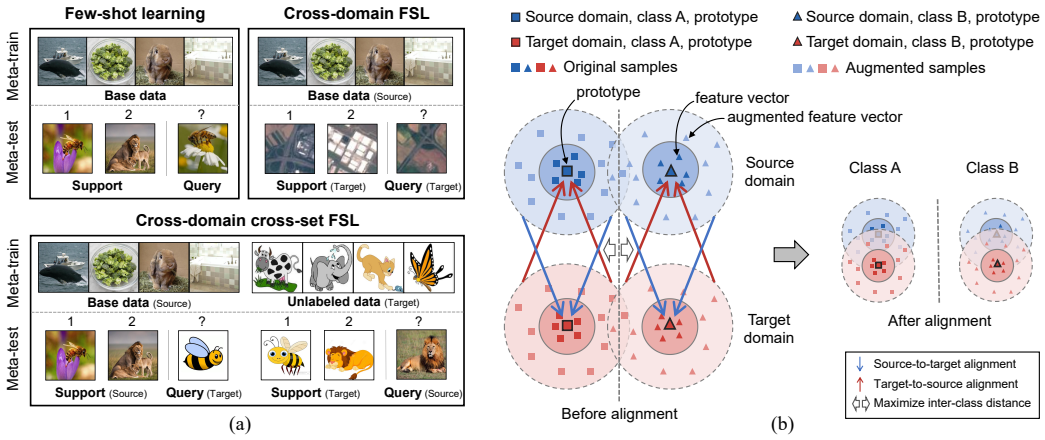

Figure 1: **Problem setup and motivation.** (a) Different from previous FSL settings, CDCS-FSL assumes a domain gap exists between the support set and query set, and additional unlabeled target data are provided in the meta-training stage. (b) To address CDCS-FSL, we propose a bi-directional prototypical alignment strategy, which pushes feature vectors of one domain to be gathered around the prototypes in the other domain, and separates feature vectors of different classes.

target domain. To facilitate feature alignment, we provide additional unlabeled target data in the meta-training stage to allow the learning system to access target domain information beforehand.

To deal with the CDCS-FSL problem, we propose a novel bi-directional prototypical alignment strategy illustrated in Figure 1 (b). The insight of our approach is two-fold: 1) we need aligned representations to alleviate the domain shift problem, and 2) compact representations are desirable to reduce intra-class variance and enlarge inter-class distance so that a small support set can better represent a new class. Specifically, different from previous prototypical alignment methods (Pan et al., 2019; Xie et al., 2018) that directly minimize the *point-to-point* distances between class centers (prototypes), we propose to minimize the *point-to-set* distances between the prototypes and feature vectors, and constrain these distances in two directions: 1) from source features to target prototype and 2) from target features to source prototype. As a result, the feature vectors of the source (or target) domain will be gathered around the prototypes in the other domain, so that the domain gap and the intra-class variance can be reduced simultaneously. Meanwhile, we maximize the inter-class distance between samples from different classes to get a more separable feature space. Inspired by the fact that data augmentation even with strong image transformations generally does not change the sample semantics, we suppose that the augmented samples from different domains should also be aligned, and thus apply the bi-directional prototypical alignment to the augmented samples. Due to significant differences in appearance, these samples may have a more dispersive feature distribution, which can further encourage to learn the underlying invariance and strengthen feature alignment.

We refer to our approach as "**St**rongly **A**ugmented **B**i-directional **P**rototypical **A**lignment", or *stab*PA. We evaluate its effectiveness on two new CDCS-FSL benchmarks adapted from the DomainNet and Office-Home datasets. Remarkably, our approach achieves the best performance over all benchmarks and outperforms baselines with a large margin, e.g. up to 5.6 points gain compared to the state-of-the-art (SOTA) method STARTUP (Phoo & Hariharan, 2021). Our contributions are three-fold. 1) We propose CDCS-FSL, a more practical FSL setting where the support set and the query set are from different domains. 2) We propose a new approach, namely *stab*PA, to address the CDCS-FSL problem, the key of which is to learn prototypical compact and domain aligned representations. 3) Extensive experiments demonstrate that *stab*PA can learn discriminative and generalizable representations and outperforms all baselines by a large margin.

2 Related Work

2.1 Few-Shot Learning

FSL aims to learn new classes with very few labeled examples. Most studies follow a meta-learning paradigm (Vilalta & Drissi, 2002), where a meta-learner is trained on a series of training tasks

(episodes) to learn meta-knowledge across tasks so as to enable fast adaptation to new tasks. The meta-learner can take various forms, such as an LSTM network (Ravi & Larochelle, 2017), a set of initial parameters (Finn et al., 2017), or closed-form solvers (Rusu et al., 2019). Recent advances in pre-training techniques spawn another FSL paradigm: a model is first pre-trained on a large base set to obtain meta-knowledge; then only a few samples are required for model fine-tuning to complete downstream tasks. For example, Chen et al. (2019) propose a standard pre-training and fine-tuning procedure for few-shot classification, where the simple baselines achieve competitive performance to the SOTA meta-learning models. Tian et al. (2020); Chen et al. (2021) show that self-supervised pre-training techniques are useful to learn effective representations for few-shot learning.

As a realistic setting, the cross-domain FSL assumes that the base set in the meta-training (pre-training) stage is from the source domain and the support set and query set in the meta-testing (fine-tuning) stage are both from the target domain. With such domain gap, Chen et al. (2019) show that meta-learning approaches may fail to adapt to novel classes. To alleviate this problem, Tseng et al. (2020) propose a feature-wise transformation layer to learn rich representations that can generalize better to other domains. However, they need to access multiple base datasets from different domains with extra data collection costs. Another work (Ngiam et al., 2018) studies the choice of base datasets and shows that a judicious choice can improve the generalization ability of the learned representations. However, all the above works only consider the domain gap occurring between the meta-training (pre-training) stage and meta-testing (fine-tuning) stage. In this paper, we consider a more challenging setting that the support set and the query set are from different domains. To facilitate the representation adaptation to the target domain, we allow the use of additional unlabeled target images in the meta-training (pre-training) stage.

2.2 Unsupervised Domain Adaptation

Using unlabeled images to alleviate the domain shift problem has been widely investigated in the field of unsupervised domain adaptation (UDA). Early efforts align the marginal distribution of each domain by minimizing a pre-defined distribution discrepancy, such as $\mathcal{H}\Delta\mathcal{H}$-divergence (Ben-David et al., 2010) and Maximum Mean Discrepancy (MMD) (Gretton et al., 2006). Inspired by Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), Ganin et al. (2016) first adopt a domain adversarial neural network (DANN) to learn domain-invariant features, where a domain discriminator is exploited to approximate the domain discrepancy. Since then, various adversarial training based methods are proposed for learning at image level (Hoffman et al., 2018) , feature level (Long et al., 2018a) or output level (Tsai et al., 2018). While another line of works adopt semi-supervised learning techniques. For example, self-training methods (Zheng & Yang, 2021; Zhang et al., 2021; 2019) assign pseudo labels to unlabeled images and train the model with these pseudo labels iteratively. Although these UDA methods are related to our work, they usually assume that the test stage shares the same class categories in the training stage, which is broken by the setting of FSL. Besides, the test data are all from the target domain, while the CDCS-FSL assumes that the domain gap exists between the query set and support set in the meta-testing stage.

3 Problem Setup

Formally, a FSL task often adopts a setting of N-way-K-shot classification, which aims to discriminate between N novel classes with K exemplars per class. Given a support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N\times K}$ where $x_i \in \mathcal{X}_\mathcal{N}$ denotes a data sample in novel classes and $y_i \in Y_\mathcal{N}$ is the class label, the goal of FSL is to learn a mapping function $\phi : \phi(x_q) \to y_q$ which classifies a query sample x_q in the query set $\mathcal{Q}$ to the class label $y_q \in Y_\mathcal{N}$. Besides $\mathcal{S}$ and $\mathcal{Q}$, a large labeled dataset $\mathcal{B} \subset \mathcal{X}_\mathcal{B} \times \mathcal{Y}_\mathcal{B}$ (termed the base set) is often provided for pre-training (meta-training), where the sample set $\mathcal{X}_\mathcal{B}$ and the base class set $\mathcal{Y}_\mathcal{B}$ do not overlap with $\mathcal{X}_\mathcal{N}$ and $\mathcal{Y}_\mathcal{N}$.

Conventional FSL studies assume the three sets $\mathcal{S}$, $\mathcal{Q}$ and $\mathcal{B}$ are from the same domain (e.g. the natural image domain). Recently, cross-domain FSL proposes a more general assumption that the base set is from a source domain and the support set and query set are from a different target domain, i.e., $\mathcal{B} \subset \mathcal{D}_s$ and $\mathcal{S}, \mathcal{Q} \subset \mathcal{D}_t$, where $\mathcal{D}_s$ and $\mathcal{D}_t$ are the source and target domain, respectively. In this paper, we take a step further and propose a new setting where the support set and the query set are from different domains. Specifically, in this setting, we assume the base set also belongs to the source domain, i.e., $\mathcal{B} \subset \mathcal{D}_s$, while for the support set and query set, there are two situations:

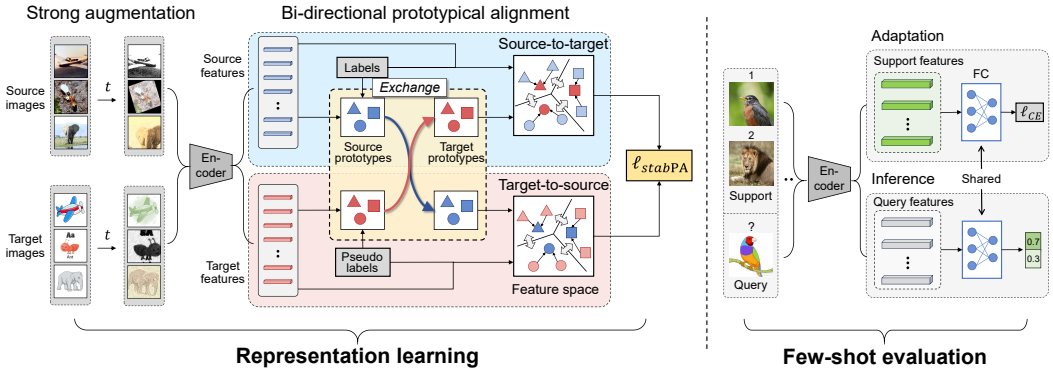

Figure 2: **Framework.** In the representation learning phase, we pre-train a feature extractor with the proposed bi-directional prototypical alignment strategy to learn compact and aligned representations. In the evaluation phase, we fix the feature extractor and train a new linear classification head with the support set. The entire model is tested on the query set.

1. $\mathcal{D}_s - \mathcal{D}_t$: the support set is from the source domain and the query set is from the target domain, i.e., $\mathcal{S} \subset \mathcal{D}_s$ and $\mathcal{Q} \subset \mathcal{D}_t$.
2. $\mathcal{D}_t - \mathcal{D}_s$: the support set is from the target domain and the query set is from the source domain, i.e., $\mathcal{S} \subset \mathcal{D}_t$ and $\mathcal{Q} \subset \mathcal{D}_s$.

We refer to such setting as cross-domain cross-set few-shot learning (CDCS-FSL). To facilitate crossing the domain gap, we allow an additional unlabelled dataset $\mathcal{U}$ from the target domain for meta-training. The class space of the unlabeled dataset has an interaction with the base classes, but does not overlap with novel classes in meta-testing.

4 Approach

Briefly, our approach contains two phases: 1) In the representation learning phase, we pre-train a feature extractor $f : x_i \rightarrow f(x_i)$ with the base set $\mathcal{B}$ and the unlabeled target set $\mathcal{U}$; 2) In the evaluation phase, we fix the feature extractor and train a linear classification head $g : f(x_i) \rightarrow y_i$ on the support set $\mathcal{S}$, and the entire model $\phi = g \circ f$ is used to predict the labels for the query set $\mathcal{Q}$. The framework of our approach is illustrated in Figure 2.

4.1 Bi-directional Prototypical Alignment

To mitigate the domain shift, a straightforward way is to learn aligned representations by minimizing the point-to-point distances between the prototypes of different domains, which are approximated using labeled data within a batch. In our case, although we can use pseudo labels to estimate prototypes of the target domain, the naive prototype alignment is still problematic due to the following two points. First, since the prototypes are estimated on a limited number of samples in a batch, they are likely to deviate from the true class centers and mislead the alignment. This problem will be exacerbated in the iterative process of pseudo-labeling and feature learning, where the noisy pseudo labels may lead to further misalignment between the source and target domains. Second, although minimizing the point-to-point distance can reduce the domain shift in intra-class samples, the feature distribution of different classes may be mixed due to the lack of constraints regarding inter-class relationships. Hence, the discrimination capability of the learned representations is still insufficient.

To overcome these drawbacks, we propose a bi-directional prototypical alignment strategy, which pushes the features from one domain to be clustered around the prototypes of the other domain, meanwhile far away from the prototypes of other classes. Given the source domain base set $\mathcal{B}$ and the target domain unlabeled set $\mathcal{U}$, we first assign pseudo labels to each target sample with an initial classifier ϕ_0 trained on the base set and obtain $\hat{\mathcal{U}} = \{(x_i, \hat{y}_i) | x_i \in \mathcal{U}\}$, where $\hat{y}_i = \phi_0(x_i)$ is the pseudo label. Then, we obtain the source prototypes $\{p_k^s\}_{k=1}^{|\mathcal{Y}_\mathcal{B}|}$ and the target prototypes $\{p_k^t\}_{k=1}^{|\mathcal{Y}_\mathcal{B}|}$

(details can be found below). It should be noted that the prototypes are estimated on the entire datasets $\mathcal{B}$ and $\hat{\mathcal{U}}$, and adjusted together with the update of the feature extractor and pseudo labels.

For a source sample $(x_i^s, y_i^s) \in \mathcal{B}$ of the q-th class, we minimize its feature distance to the prototype of the same class in the target domain, and meanwhile maximize its distances to prototypes of other classes. Here, a softmax loss function for the source-to-target alignment is formulated as:

$$\ell_{s-t}(x_i^s, y_i^s) = -\log \frac{\exp\left(-||f(x_i^s) - p_q^t||/\tau\right)}{\sum_{k=1}^{|\mathcal{Y}_\mathcal{B}|} \exp\left(-||f(x_i^s) - p_k^t||/\tau\right)}, \tag{1}$$

where τ is a temperature factor. Similarly, for a target sample $(x_i^t, \hat{y}_i^t) \in \hat{\mathcal{U}}$ with $\hat{y}_i^t = q$, a target-to-source alignment loss function is as follows:

$$\ell_{t-s}(x_i^t, \hat{y}_i^t) = -\log \frac{\exp\left(-||f(x_i^t) - p_q^s||/\tau\right)}{\sum_{k=1}^{|\mathcal{Y}_\mathcal{B}|} \exp\left(-||f(x_i^t) - p_k^s||/\tau\right)}. \tag{2}$$

Since the initial pseudo labels are more likely to be incorrect, we gradually increase the weight of these two losses following the principle of curriculum learning (Bengio et al., 2009). For the source-to-target alignment, the loss weight starts from zero and converges to one, formulated as:

$$w(t) = \frac{2}{1 + \exp\left(-\alpha t / T_{max}\right)} - 1, \tag{3}$$

where t is the current training step, T_{max} is the maximum training step, and α is a hyperparameter controlling the convergence speed. For the target-to-source alignment, since the pseudo labels become more confident along with the training process, a natural curriculum is achieved by setting a confidence threshold to filter out the target samples with low confidence pseudo labels (Sohn et al., 2020).

Therefore, the total loss for the bi-directional prototypical alignment is

$$\ell_{b\text{PA}} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} w(t) \ell_{s-t}(x_i^s, y_i^s) + \frac{1}{|\hat{\mathcal{U}}|} \sum_{i=1}^{|\hat{\mathcal{U}}|} \mathbb{1}(p(\hat{y}_i^t) > \beta) \ell_{t-s}(x_i^t, \hat{y}_i^t), \tag{4}$$

where $p(\cdot)$ is the confidence of a pseudo label, and β is the confidence threshold below which the data samples will be dropped.

Updating Pseudo Label. The pseudo labels are initially predicted by a classifier ϕ_0 pre-trained on the base set $\mathcal{B}$. As the representations are updated, we update the pseudo labels by re-training a classifier $\phi_t = h \circ f$ based on the current feature extractor f, where h is a linear classification head for the base classes. The final pseudo labels are updated by linear interpolation between the predictions of the initial classifier ϕ_0 and the online updated classifier ϕ_t:

$$\hat{y}_i = \arg\max_k \ \lambda \phi_0(k|x_i) + (1 - \lambda) \phi_t(k|x_i), \tag{5}$$

where λ is the interpolation coefficient. The combination of these two classifiers makes it possible to rectify the label noise of the initial classifier, and meanwhile inhibit the rapid change of pseudo labels of online classifier especially in the early training stage.

Generating Prototypes. Note that we are intended to estimate the prototypes on the entire dataset and update them with representation learning. For the source domain, instead of calculating the mean value of intra-class samples in the feature space, a cheaper way is to approximate prototypes with the normalized weights of the classification head h, as the classifier weights tend to align with class centers in order to reduce classification error (Qiao et al., 2018). Specifically, we set the source prototypes as $p_k^s = W_k$, where W_k is the normalized classification weight for the k-th class. For the target domain, we adopt the momentum technique to update prototypes. The prototypes are initialized as zeros. At each training step, we first estimate the prototypes using target samples in the current batch with their pseudo labels. Then, we update target prototype p_k^t as:

$$p_k^t \longleftarrow m p_k^t + (1 - m) \frac{1}{n_k} \sum_{i=1}^{|\hat{\mathcal{U}}_b|} \mathbb{1}(\hat{y}_i^t = k) f(x_i^t), \tag{6}$$

where n_k is the number of the target samples classified into the k-th class in a target batch $\hat{\mathcal{U}}_b$, and m is the momentum term controlling the update speed.

4.2 *stab*PA

Strong data augmentation has proved to be effective for learning generalizable representations, especially in self-supervised learning studies, e.g. contrastive learning. Given a sample x, strong data augmentation generates additional data points $\{\widetilde{x}_i\}_{i=1}^n$ by applying various strong transformations (Chen et al., 2020; He et al., 2020). The assumption behind strong data augmentation is that the strong transformation does not change the semantics of the original samples.

In this work, we further hypothesize that strongly augmented intra-class samples in different domains can also be aligned. It is expected that strong data augmentation can further strengthen the learning of cross-domain representations, since stronger augmentation provides more diverse data samples and makes the learned aligned representations more robust for various transformations in both the source domain and target domain.

Following this idea, we extend the bi-directional prototypical alignment with strong data augmentation and the proposed entire framework is termed *stab*PA. Specifically, for a source sample (x_i^s, y_i^s) and a target sample $(x_i^t, \hat{y}_i^t)$, we generate their strongly augmented versions $(\widetilde{x}_i^s, y_i^s)$ and $(\widetilde{x}_i^t, y_i^t)$. Within the bi-directional prototypical alignment framework, we minimize the feature distance of a strongly augmented image to its prototype with the same class label in the other domain, and maximize its distance to the prototypes of other classes. Totally, the *stab*PA loss is

$$\ell_{stab\text{PA}} = \frac{1}{|\widetilde{\mathcal{B}}|}\sum_{i=1}^{|\widetilde{\mathcal{B}}|} w(t)\ell_{s-t}(\widetilde{x}_i^s, y_i^s) + \frac{1}{|\widetilde{\mathcal{U}}|}\sum_{i=1}^{|\widetilde{\mathcal{U}}|} \mathbb{1}(p(\hat{y}_i^t) > \beta)\ell_{t-s}(\widetilde{x}_i^t, \hat{y}_i^t), \tag{7}$$

where $\widetilde{\mathcal{B}}$ and $\widetilde{\mathcal{U}}$ are the augmented base set and unlabeled target set, respectively.

To perform strong data augmentation, we apply random crop, Cutout (DeVries & Taylor, 2017), and RandAugment (Cubuk et al., 2020). RandAugment comprises 14 different transformations and randomly selects a fraction of transformations for each sample. In Cubuk et al. (2020), a global magnitude controlling all transformations needs to be optimized via grid search on a validation set. However, random selection of the magnitude for each transformation works well in our study, which is similar to Sohn et al. (2020).

5 Experiments

5.1 Adapting the Datasets for CDCS-FSL

DomainNet. DomainNet (Peng et al., 2019) is a large scale multi-domain image dataset. It contains 345 categories in 6 different domains, about 0.6 million images in total. In our experiments, we choose the *real* domain as the source domain and choose one domain from *painting*, *clipart* and *sketch* as the target domain. Similar to previous work, we randomly split the dataset into 3 parts: base set (228 categories), validation set (33 categories) and novel set (65 categories), and discard 19 categories with too few images. To construct the unlabeled target dataset, we remove the labels of the target base set and validation set. These unlabeled images combined with the labeled source base set are use for pre-training. The validation sets in both domains are used to tune the hyperparameters. We finally report the 5-way 1-shot and 5-way 5-shot accuracies on the novel set.

Office-Home. Office-Home (Venkateswara et al., 2017) contains 65 object categories usually found in office and home settings. We randomly select 40 categories as the base set, 10 categories as the validation set and 15 categories as the novel set. There are 4 domains for each category: *real*, *art*, *clipart* and *product*. We set the source domain as *real* and choose the target domain from the other three domains. The training and testing process are the identical with the DomainNet dataset.

5.2 Comparison with Baselines

We first compare our approach with conventional FSL methods, including ProtoNet (Snell et al., 2017), RelationNet (Sung et al., 2018), MetaOptNet (Lee et al., 2019), Tian et al. (2020) and DeepEMD (Zhang et al., 2020), which all train the model only with source domain data. We also compare to the methods that leverage unlabeled target data to alleviate domain shift, including the adversarial training method DANN (Ganin et al., 2016), semi-supervised learning methods Mean Teacher

Table 1: Comparison to baselines on the DomainNet dataset. We report 5-way 1-shot and 5-way 5-shot accuracies with 95% confidence interval.

	real-painting		real-clipart		real-sketch	
Method	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet (*NeurIPS'17*)	45.36±0.81	**57.23±0.79**	44.65±0.81	58.04±0.81	39.28±0.77	**51.68±0.81**
RelationNet (*CVPR'18*)	42.69±0.77	52.63±0.74	44.12±0.81	57.24±0.80	36.52±0.73	47.32±0.75
MetaOptNet (*CVPR'19*)	44.02±0.77	56.34±0.34	42.46±0.80	57.92±0.79	36.37±0.72	48.20±0.79
Tian et al. (*ECCV'20*)	46.69±0.86	56.87±0.84	48.30±0.85	59.67±0.84	40.23±0.73	50.41±0.80
DeepEMD (*CVPR'20*)	**47.60±0.87**	56.62±0.78	**49.02±0.83**	**60.43±0.82**	**42.75±0.79**	51.66±0.80
DANN (*JMLR'16*)	45.94±0.84	56.83±0.86	47.31±0.86	59.42±0.84	42.44±0.79	53.47±0.75
Mean Teacher (*NeurIPS'17*)	46.92±0.83	57.74±0.84	48.48±0.81	61.54±0.84	43.39±0.81	54.57±0.79
Fixmatch (*NeurIPS'20*)	**48.86±0.87**	**61.62±0.79**	48.70±0.82	**61.94±0.82**	**44.48±0.80**	**55.26±0.83**
STARTUP (*ICLR'21*)	47.53±0.88	58.13±0.82	**49.24±0.87**	61.51±0.86	43.78±0.82	54.89±0.81
*stab*PA (*Ours*)	**50.51±0.85**	**63.19±0.78**	**51.63±0.83**	**62.78±0.85**	**47.54±0.81**	**59.10±0.79**
	painting-real		clipart-real		sketch-real	
ProtoNet (*NeurIPS'17*)	45.25±0.97	**65.60±0.95**	47.50±0.95	65.91±0.78	42.85±0.89	59.46±0.85
RelationNet (*CVPR'18*)	43.04±0.97	61.18±0.90	45.86±0.95	62.65±0.81	41.29±0.96	56.39±0.88
MetaOptNet (*CVPR'19*)	44.31±0.94	63.20±0.89	46.15±0.98	63.51±0.82	40.27±0.95	55.65±0.85
Tian el al. (*ECCV'20*)	46.57±0.99	63.90±0.95	49.66±0.98	65.33±0.80	41.90±0.86	56.95±0.84
DeepEMD (*CVPR'20*)	**47.86±1.04**	63.86±0.93	**50.89±1.00**	**67.46±0.78**	**46.02±0.93**	**60.39±0.87**
DANN (*JMLR'16*)	46.85±0.97	64.29±0.94	50.02±0.94	66.87±0.78	43.66±0.92	60.14±0.81
Mean Teacher (*NeurIPS'17*)	46.84±0.96	64.97±0.94	49.60±0.97	67.39±0.89	44.52±0.89	60.04±0.86
Fixmatch (*NeurIPS'20*)	**49.15±0.93**	**67.46±0.89**	49.18±0.93	66.72±0.81	**45.97±0.95**	**62.46±0.87**
STARTUP (*ICLR'21*)	47.58±0.98	65.27±0.92	**51.32±0.98**	**67.95±0.78**	45.23±0.96	61.97±0.88
*stab*PA (*Ours*)	**51.87±0.99**	**70.84±0.88**	**53.53±1.04**	**71.57±0.81**	**49.18±0.96**	**67.14±0.85**

(Tarvainen & Valpola, 2017), Fixmatch (Sohn et al., 2020), and the cross-domain FSL method STARTUP (Phoo & Hariharan, 2021). For a fair comparison, we re-implement these methods with the same backbone and optimizer. Further details can be found in Appendix A.1. The comparison results are shown in Tables 1 and 2.

***stab*PA vs few-shot learning methods.** On the DomainNet, our approach outperforms all the FSL baselines by a large margin across different domains and situations. Compared to ProtoNet, our approach improves the 5-shot accuracy by 7.4% in the most difficult *real-sketch* situation, and by 5.7% in the easier *clipart-real* situation. Similar results can be found on the Office-Home dataset in Table 2 . The significant improvements indicate that few-shot learners trained on one domain are difficult to adapt to the other domains, while the proposed *stab*PA learning aligned representations across domains can alleviate this problem and improve the cross-domain FSL performance.

***stab*PA vs pseudo-labeling methods.** Similar to our approach, Mean Teacher, Fixmatch and STARTUP train the model with additional unlabeled target images. Particularly, Fixmatch also applies strong data augmentation to unlabeled images. However, our approach outperforms them in all situations. We claim that the strength of *stab*PA derives not only from pseudo labeling and strong augmentation, but also from the proposed bi-directional prototypical alignment strategy. This is particularly evident in the *real-sketch* and *sketch-real* situations (up to 4.7% improvement in 5-shot accuracy), where the domain shift is very significant.

5.3 Results Analysis

5.3.1 Has *stab*PA learned compact and aligned representations?

To verify whether *stab*PA indeed learns compact and aligned representations, we visualize the feature distribution through the pre-training process using t-SNE (Van der Maaten & Hinton, 2008). From Figure 3 (a)-(d), we can see that in the beginning, samples from different classes are heavily mixed. There are no distinct classification boundaries between classes. Besides, samples from two

Table 2: Comparison on the Office-Home dataset.

Method	real-product		real-clipart		real-art	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet (*NeurIPS'17*)	30.72±0.62	39.74±0.64	28.52±0.58	34.81±0.59	26.80±0.47	34.56±0.58
Tian et al. (*ECCV'20*)	33.88±0.69	45.79±0.69	30.44±0.60	38.27±0.64	30.26±0.57	38.80±0.61
Fixmatch (*NeurIPS'20*)	**36.05±0.73**	**48.45±0.70**	**33.79±0.64**	**43.13±0.67**	31.81±0.60	41.48±0.60
STARTUP (*ICLR'21*)	34.62±0.74	47.18±0.71	30.70±0.63	38.10±0.62	**32.06±0.59**	**41.94±0.63**
*stab*PA (*Ours*)	**37.71±0.80**	**50.28±0.73**	**34.04±0.69**	**43.86±0.64**	**32.76±0.62**	**43.05±0.62**
	product-real		clipart-real		art-real	
ProtoNet (*NeurIPS'17*)	30.27±0.62	38.98±0.64	28.44±0.63	35.85±0.59	27.31±0.58	36.27±0.66
Tian et al. (*ECCV'20*)	33.98±0.67	44.27±0.63	30.86±0.66	38.99±0.63	30.30±0.62	41.56±0.72
Fixmatch (*NeurIPS'20*)	**35.83±0.76**	**47.17±0.68**	**33.20±0.74**	**43.20±0.69**	32.32±0.66	44.68±0.72
STARTUP (*ICLR'21*)	34.80±0.68	45.00±0.64	30.17±0.68	38.84±0.70	**32.40±0.66**	**44.71±0.73**
*stab*PA (*Ours*)	**36.78±0.73**	**48.61±0.66**	**34.73±0.71**	**43.41±0.65**	**33.33±0.68**	**46.02±0.77**

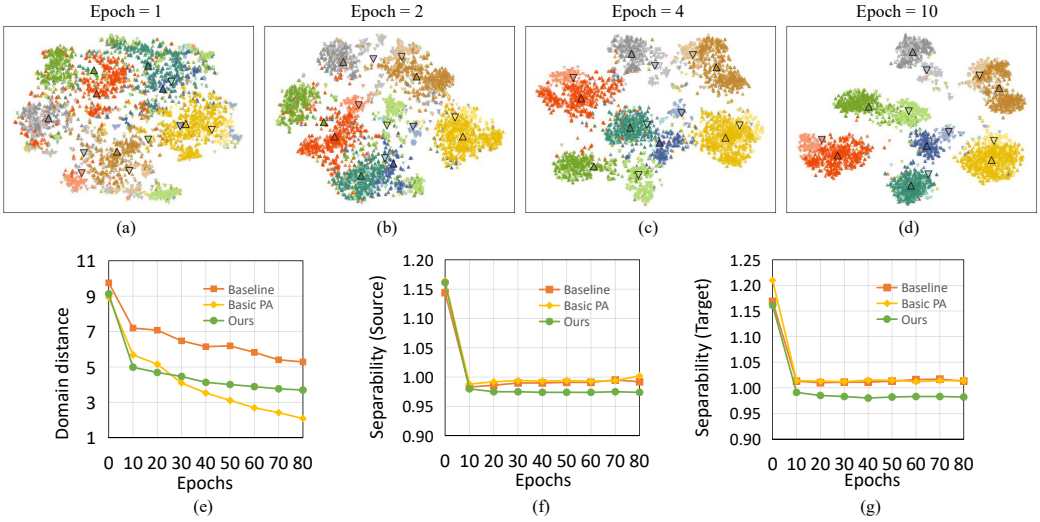

Figure 3: (a)-(d) t-SNE visualization of feature distribution at different training epochs. Samples of the same class are painted in similar colors, where darker triangles represent source samples and lighter reverted triangles represent target samples (best viewed in color). Class centers are marked in black border. (e) Domain distance on novel classes. (f)-(g) Separability among novel classes in the source and target domains. Separability is represented by the average distance ratio, the lower the better.

domains are far away from each other, which indicates the existence of a considerable domain shift (such as the classes in green and orange). However, as training continues, samples from the same class begin to aggregate together, and the margin between different classes gets larger and larger, i.e., the feature distribution becomes more compact. Moreover, we can see that samples from different domains are grouping into their ground-truth classes, even though no label information is given for the target domain. These observations demonstrate that *stab*PA is indeed capable to learn compact and aligned representations.

5.3.2 Can *stab*PA learn generalizable representations for new classes?

To validate the generalization capability of the representations learned by stabPA, we propose two quantitative metrics which indicate the domain distance and class separability among new classes.

Specifically, to the measure domain distance, we first calculate prototypes p_k^s and p_k^t for each novel class in the source and target domain. Then we obtain the Euclidean distance between the two prototypes per class and compute the average distance over all novel classes. We refer to this metric

Table 3: Ablation studies on DomainNet. Mean and 95% confidence interval are reported.

			sketch-real		real-sketch	
ℓ_{s-t}	ℓ_{t-s}	aug	1-shot	5-shot	1-shot	5-shot
×	×	×	41.90±0.86	56.95±0.84	40.23±0.73	50.41±0.80
✓	×	×	44.83±0.95	60.87±0.91	42.86±0.78	52.16±0.78
×	✓	×	44.45±0.92	61.97±0.90	44.20±0.77	54.83±0.79
✓	✓	×	47.59±1.00	64.32±0.86	47.01±0.84	56.68±0.81
✓	✓	✓	**49.18±0.96**	**67.14±0.85**	**47.54±0.81**	**59.10±0.79**

as Prototype Distance (PD), which can be formulated as: $PD = \frac{1}{|\mathcal{Y}_\mathcal{N}|}\sum_{k\in\mathcal{Y}_\mathcal{N}}||p_k^s - p_k^t||$. A small PD value means the two domains are well aligned to each other.

To represent the class separability, for each sample x_i with the class label y_i, we calculate the ratio of its distance to the prototype of class y_i to the distance to the closest neighbouring class prototype. Then the average is computed over all samples in novel classes, which is termed Average Distance Ratio (ADR). Formally, $ADR = \frac{1}{|\mathcal{X}_\mathcal{N}|}\sum_{x_i\in\mathcal{X}_\mathcal{N}}\frac{||f(x_i)-p_{y_i}||}{\min_{k\neq y_i}||f(x_i)-p_k||}$. When ADR is less than 1, most samples can be correctly classified into their ground-truth classes. We calculate the ADR for the source domain and target domain separately to validate whether the learned representations can generalize across different domains.

In experiments, we compare the proposed *stab*PA approach with a FSL baseline method (Tian et al., 2020) that does not leverage target images, and Basic PA which aligns two domains by simply minimizing the point-to-point distance between prototypes of two domains (Xie et al., 2018). The results are presented in Figure 3 (e)-(g). We can notice that all these methods can achieve lower domain distance as training processes, and Basic PA gets the lowest domain distance at the end. However, Basic PA does not improve the class separability as much as our approach, as shown in Figure 3 (f)-(g). The inferior class separability can be understood that Basic PA merely aims to reduce the feature distance between two domains, without taking account of the intra-class and inter-class distances in the learned feature space. Rather than the global alignment adopted by Basic PA, the proposed *stab*PA considers the feature-to-prototype distances across different domains and classes, so that the domain alignment and class separability can be improved at the same time.

5.3.3 Ablation Studies

We conduct ablation studies on various components of the *stab*PA. The results on the DomainNet dataset are shown in Table 3. As all key components are removed, we adopt the baseline method Tian et al. (2020) to train feature extractor with only the source data, which is the first row of the table. When the unlabeled target data are available, applying either source-to-target alignment or target-to-source alignment can improve the performance evidently. Interestingly, we can see that the target-to-source alignment is more effective than the source-to-target alignment (about 1.2 points on average). This is probably because the source prototypes estimated by the ground truth labels are more accurate than the target prototypes estimated by the pseudo labels. Improving the quality of target prototypes may reduce this gap. When combing these two alignments together, we can get better results, indicating that the two kinds of alignment are to some extent complementary to each other. Finally, the best results are obtained by combining the strong data augmentation techniques, verifying that strong data augmentation can further strengthen the cross-domain alignment.

6 Conclusions

In this work, we have investigated a novel problem in FSL, namely CDCS-FSL, where a domain shift exists between the support set and query set. To tackle this problem, we have proposed *stab*PA, a prototype-based domain alignment framework to learn compact and aligned representations. On two widely-used multi-domain FSL datasets, we have built benchmarks and compared our approach to multiple elaborated baselines. Extensive experimental results have demonstrated the advantage of our approach. Through more in-depth analysis, we have also validated the generalization capability of the representations learned by stabPA and the effectiveness of each component of the proposed model.

7 REPRODUCIBILITY STATEMENT

We have uploaded the source code as supplemental materials to ensure reproducibility.

REFERENCES

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020.

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

Wentao Chen, Chenyang Si, Wei Wang, Liang Wang, Zilei Wang, and Tieniu Tan. Few-shot learning with part discovery and augmentation from unlabeled images. *arXiv preprint arXiv:2105.11874*, 2021.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

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

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems*, 19: 513–520, 2006.

Yanyang Gu, Zongyuan Ge, C Paul Bonnington, and Jun Zhou. Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease classification. *IEEE Journal of Biomedical and Health Informatics*, 24(5):1379–1393, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Ran He, Xiang Wu, Zhenan Sun, and Tieniu Tan. Wasserstein cnn: Learning invariant features for nir-vis face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7): 1761–1773, 2018.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pp. 1989–1998. PMLR, 2018.

Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 599–617. Springer, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Jiawei Liu, Zheng-Jun Zha, Di Chen, Richang Hong, and Meng Wang. Adaptive transfer network for cross-domain person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018a.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 1645–1655, 2018b.

Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2239–2247, 2019.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences. In *International Conference on Learning Representations*, 2021.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7229–7238, 2018.

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

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

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7472–7481, 2018.

Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 5423–5432. PMLR, 2018.

Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12414–12424, 2021.

Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision*, 129(4):1106–1120, 2021.

A Appendix

A.1 Implementation Details

Hyperparameters. In our implementation, ResNet-18 (He et al., 2016) is adopted as the backbone, which outputs a 512-d feature vector. Before feeding the vector for prototypical alignment, we apply ℓ_2 normalization for the feature vector and prototypes. The temperature τ for ℓ_{s-t} and ℓ_{t-s} is 0.25 and 0.1, respectively. To control the loss weight for ℓ_{s-t}, T_{max} and α are set as 50,000 and 7, respectively. The loss weight for ℓ_{t-s} is fitted adaptively by setting the confidence threshold $\beta = 0.5$. We set $\lambda = 0.2$ to balance the pseudo label generated by the initial classifier and the online updated classifier. The momentum term m is set as 0.99.

Training. We train our approach for 50 epochs on the DomainNet dataset. On the smaller Office-Home dataset, we train the model for 100 epochs. Adam (Kingma & Ba, 2014) is adopted as the default optimizer with the learning rate as 1e-3. The batch size is set as 256, where source data and target data have the same number in a batch.

Evaluation. During evaluation, we fix the feature extractor and apply ℓ_2 normalization to the output feature vector. The linear classification head for each few-shot task (episode) is randomly initialized, and trained on the support features for 1000 steps with logistic regression. 15 query samples per class are used to evaluate the performance of the learned classifier. We finally report the average accuracy over 600 episodes with 95% confidence interval.

ProtoNet and RelationNet. ProtoNet and RelationNet are two meta-learning methods, which are trained on a series of few-shot tasks (episodes). During training, we randomly sample episodes from the base set, each of which contains $N = 5$ classes and $K = 5$ samples per class serving as the support set, and another 15 samples per class as the query set. We also train ProtoNet and RelationNet for 50 epochs on the DomainNet dataset and 100 epochs on the Office-Home dataset. The number of training episodes of each epoch is particularly defined to make sure the number of seen samples (both the support and query samples) in an epoch is roughly equal to the size of the dataset.

MetaOptNet. MetaOptNet aims to learn an embedding function that generalizes well to novel categories under the linear classification rule. We implement this method based on the official code [1] but replace the backbone network and optimizer to be the same as our approach. Similar to ProtoNet and RelationNet, the training process of MetaOptNet is also episodic.

Tian et al. Tian et al. (2020) follows the transfer learning paradigm, which trains a base model to classify base classes, and leverage the learned representations to classify novel classes by learning a new classification head. We train this baseline with the same optimization method as our approach except that the batch size is set as 128 as only source data are used for training.

DeepEMD. DeepEMD contains two training phases: pre-training and meta-training. We use the output model of Tian et al. as the pre-trained model and then follow the official implementation [2] to finetune the model via meta-training.

DANN. We use a three-layer fully connected network as the domain discriminator to implement DANN, following the Pytorch implementation [3] released by Long et al. (2018b). The gradient reverse layer (Ganin et al., 2016) is adopted to train the feature vector and domain discriminator in an adversarial manner. To stabilize training, the weight of the adversarial loss starts from zero, and gradually grows to one.

Mean Teacher, Fixmatch and STARTUP. All of these approaches use pseudo-labeled samples to train the model. Differently, Mean Teacher predicts pseudo labels with a teacher network that is the ensemble of historical models by aggregating their model weights with exponential moving average (EMA). In our implementation, the smoothing coefficient for EMA is set as 0.99. Fixmatch trains the model with a consistency loss, i.e., enforcing the network prediction for a strongly augmented sample to be consistent with the prediction of its weakly augmented counterpart. We implement Fixmatch based on a publicly available implementation[4]. STARTUP adopts fixed pseudo labels that are predicted by a classifier pre-trained on the base set, and imposes a self-supervised loss on the target data. In our re-implementation, we do not utilize the self-supervised loss item since we find that it does not provide improvement in our case.

A.2 Pseudo Label Update Strategy

Since we resort to pseudo labels for prototype estimation and feature alignment, ensuring the pseudo label accuracy is very important to the effectiveness of our bi-directional prototypical alignment strategy. Pseudo labels can be predicted with a fixed classifier pre-trained on the source base dataset, as in Phoo & Hariharan (2021), or a classifier that is online updated along the representation learning. In our implementation, we combine them together by linearly interpolating their pseudo labels.

[1] https://github.com/kjunelee/MetaOptNet

[2] https://github.com/icoz69/DeepEMD

[3] https://github.com/thuml/CDAN

[4] https://github.com/kekmodel/FixMatch-pytorch

Method	λ	painting-real		real-painting	
		1-shot	5-shot	1-shot	5-shot
ℓ_{s-t}	0.0	48.56±1.04	66.24±0.92	48.89±0.85	58.72±0.79
	0.2	48.76±0.99	67.21±0.91	49.08±0.84	59.33±0.77
	1.0	48.20±0.99	65.79±0.90	48.79±0.87	57.51±0.79
ℓ_{t-s}	0.0	47.83±1.01	66.30±0.92	48.26±0.83	59.17±0.77
	0.2	48.53±1.02	67.09±0.94	48.88±0.84	60.18±0.80
	1.0	48.93±1.01	67.12±0.94	48.53±0.86	59.29±0.80
ℓ_{s-t} & ℓ_{t-s}	0.0	48.35±1.02	66.96±0.91	48.72±0.84	59.78±0.78
	0.2	49.31±1.06	68.15±0.92	49.63±0.84	60.13±0.79
	1.0	48.74±1.02	66.81±0.93	49.26±0.87	59.33±0.81

Table 4: The impact of interpolation coefficient λ.

When the interpolation coefficient $\lambda = 0\,(\text{or}1)$, our approach degenerates to only using the fixed (or online updated) classifier. We assess the effectiveness of this combining strategy on the DomainNet dataset. The results are shown in Table 4.

For the source-to-target alignment, we can see that interpolation with $\lambda = 0.2$ is better than both the fixed and online updated classifier. For the target-to-source alignment, $\lambda = 0.2$ and $\lambda = 1$ compete with each other, but both are better than $\lambda = 0$. When applying alignment from two directions to form our final model, the use of combined pseudo labels is still better than using either of them. The reason for the success of the combination is that the pseudo labels predicted by online updated classifier will change constantly especially in early training stage, which disturbs the model training. Interpolation with the fixed pseudo labels can make the pseudo labels more consistent. On the other hand, the pseudo labels predicted by current classifier can rectify the noise in the fixed pseudo labels, as demonstrated in Zhang et al. (2021).

A.3 Dataset Partition

A.3.1 DomainNet

DomainNet contains 345 categories in total. We discard 19 categories with too few images and randomly split the rest 326 categories into three sets: 228 categories for the base set, 33 categories for the validation set, and 65 categories for the novel set. The detailed categories of each set are listed below:

$$\mathcal{Y}_{base} =$$

{aircraft carrier, airplane, alarm clock, ambulance, animal migration, ant, asparagus, axe, backpack, bat, bathtub, beach, bear, beard, bee, belt, bench, bicycle, binoculars, bird, book, boomerang, bottlecap, bowtie, bracelet, brain, bread, bridge, broccoli, broom, bus, butterfly, cactus, cake, calculator, camera, candle, cannon, canoe, car, cat, ceiling fan, cell phone, cello, chair, church, circle, clock, cloud, coffee cup, computer, couch, cow, crab, crayon, crocodile, cruise ship, diamond, dishwasher, diving board, donut, dragon, dresser, drill, drums, duck, ear, elbow, elephant, envelope, eraser, eye, fan, feather, fence, finger, fire hydrant, fireplace, firetruck, flamingo, flashlight, flip flops, flower, flying saucer, foot, fork, frog, frying pan, giraffe, goatee, grapes, grass, guitar, hamburger, hammer, hand, harp, headphones, hedgehog, helicopter, helmet, hockey puck, hockey stick, horse, hot air balloon, hot tub, hourglass, hurricane, jacket, key, keyboard, knee, ladder, lantern, laptop, leaf, leg, light bulb, lighter, lightning, lion, lobster, lollipop, mailbox, marker, matches, megaphone, mermaid, microphone, microwave, moon, motorbike, moustache, nail, necklace, nose, octagon, oven, paint can, paintbrush, palm tree, panda, pants, paper clip, parachute, parrot, passport, peanut, pear, peas, pencil, penguin, pickup truck, picture frame, pizza, pliers, police car, pond, popsicle, postcard, potato, power outlet, purse, rabbit, radio, rain, rainbow, rake, remote control, rhinoceros, rifle, sailboat, school bus, scorpion, screwdriver, see saw, shoe, shorts, skateboard, skyscraper, smiley face, snail, snake, snorkel, soccer ball, sock, stairs, stereo, stethoscope, stitches, stove, strawberry, submarine, sweater, swing set, sword, t-shirt, table, teapot, teddy-bear, television, tent, the Eiffel Tower, the Mona Lisa, toaster, toe, toilet, tooth, toothbrush, tornado, tractor, train, tree, triangle, trombone, truck, underwear, van, vase, violin, washing machine, watermelon, waterslide, whale, wheel, windmill, wine bottle, zigzag}

$$\mathcal{Y}_{validation} =$$

{arm, birthday cake, blackberry, bulldozer, campfire, chandelier, cooler, cup, dumbbell, hexagon, hospital, house plant, ice cream, jail, lighthouse, lipstick, mushroom, octopus, raccoon, roller coaster, sandwich, saxophone, scissors, skull, speedboat,

spreadsheet, suitcase, swan, telephone, traffic light, trumpet, wine glass, wristwatch}

$$\mathcal{Y}_{novel} =$$

{anvil, banana, bandage, barn, basket, basketball, bed, blueberry, bucket, camel, carrot, castle, clarinet, compass, cookie, dog, dolphin, door, eyeglasses, face, fish, floor lamp, garden, garden hose, golf club, hat, hot dog, house, kangaroo, knife, map, monkey, mosquito, mountain, mouth, mug, ocean, onion, owl, piano, pig, pillow, pineapple, pool, river, rollerskates, sea turtle, sheep, shovel, sink, sleeping bag, spider, spoon, squirrel, steak, streetlight, string bean, syringe, tennis racquet, the Great Wall of China, tiger, toothpaste, umbrella, yoga, zebra}

A.3.2 OFFICE-HOME

There are 65 categories in the Office-Home dataset. We select 40 categories as the base set, 10 categories as the validation set, and 15 categories as the novel set, which are listed below:

$$\mathcal{Y}_{base} =$$

{alarm clock, bike, bottle, bucket, calculator, calendar, chair, clipboards, curtains, desk lamp, eraser, exit sign, fan, file cabinet, folder, glasses, hammer, kettle, keyboard, lamp shade, laptop, monitor, mouse, mug, paper clip, pen, pencil, postit notes, printer, radio, refrigerator, scissors, sneakers, speaker, spoon, table, telephone, toothbrush, toys, tv}

$$\mathcal{Y}_{validation} =$$

{bed, computer, couch, flowers, marker, mop, notebook, pan, shelf, soda}

$$\mathcal{Y}_{novel} =$$

{backpack, batteries, candles, drill, flipflops, fork, helmet, knives, oven, push pin, ruler, screwdriver, sink, trash can, webcam}