SEMI-SUPERVISED DOMAIN ADAPTATION WITH PROTOTYPICAL ALIGNMENT AND CONSISTENCY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain adaptation enhances generalizability of a model across domains with domain shifts. Most research effort has been spent on Unsupervised Domain Adaption (UDA) which trains a model jointly with labeled source data and unlabeled target data. This paper studies how much it can help address domain shifts if we further have a few target samples (e.g., one sample per class) labeled. This is the so-called semi-supervised domain adaptation (SSDA) problem and the few labeled target samples are termed as "landmarks". To explore the full potential of landmarks, we incorporate a prototypical alignment (PA) module which calculates a target prototype for each class from the landmarks; source samples are then aligned with the target prototype from the same class. To further alleviate label scarcity, we propose a data augmentation based solution. Specifically, we severely perturb the labeled images, making PA non-trivial to achieve and thus promoting model generalizability. Moreover, we apply consistency learning on unlabeled target images, by perturbing each image with light transformations and strong transformations. Then, the strongly perturbed image can enjoy "supervised-like" training using the pseudo label inferred from the lightly perturbed one. Experiments show that the proposed method, though simple, reaches significant performance gains over state-of-the-art methods, and enjoys the flexibility of being able to serve as a plug-and-play component to various existing UDA methods and improve adaptation performance with landmarks provided.

1 INTRODUCTION

Domain adaptation investigates techniques of avoiding severe performance drop when deploying a model on a new domain (target) that has domain gap with the one (source) which the model is trained on. Most existing research focuses on Unsupervised Domain Adaptation (UDA) which trains a model jointly with unlabeled target data and labeled source data. Many effective UDA approaches have been proposed, from early works that project data from both domains to a shared feature space (Gong et al., 2012; Pan et al., 2011), to recent ones that are based on adversarial learning (Tzeng et al., 2015; Long et al., 2018).

This paper makes a slight diversion from the mainstream UDA research direction and investigates how much it can help if we are further provided with a few (e.g., one sample per class) labeled target samples. We call these scarce labeled target samples as "landmarks". This is a practical (with minimal labeling effort) yet under-investigated problem and was referred as semi-supervised domain adaptation (SSDA). Preliminary works before the deep learning era use landmarks to more precisely measure the data distribution mismatch between source and target domains, either based on Maximum Mean Discrepancy (MMD) or domain invariant subspace learning (Ao et al., 2017; Donahue et al., 2013; Yao et al., 2015). Recent ones revisit this problem and establish new evaluation benchmarks in the deep learning context (Saito et al., 2019; Kim & Kim, 2020). However, these pioneering works have not fully realized the value of the precious landmarks as they are mainly used to optimize the cross-entropy loss along with labeled source samples that are of a much larger amount. The contribution of the landmarks is significantly diluted and thus the learned model shall still be biased towards source domain.

In this paper, we propose a Prototypical Alignment and Consistency Learning (PACL) based framework which addresses SSDA by empowering existing UDA methods to comprehensively exploit the limited yet precious landmarks. PACL is model-agnostic in the sense that most existing UDA models can be integrated into the framework and have adaption performance enhanced providing landmarks. A main composing component of PACL is the prototypical alignment (PA) module which calculates a target prototype for each class by averaging feature embeddings of the landmarks from that class. Then, source samples are aligned with the target prototype from the same class. In this way, we achieve class-level domain alignment, a desirable complement of the domain-level alignment fulfilled by UDA methods.

Other techniques of PACL are based on data augmentation. Data augmentation is a common regularization technique of avoiding overfitting for supervised learning (Krizhevsky et al., 2012; Cubuk et al., 2019a). It is also shown recently that data augmentation can significantly advance semi-supervised learning performance when properly applied (Xie et al., 2019; Sohn et al., 2020). Inspired by this, we propose a data augmentation based solution to further address the label scarce problem. We apply strong data augmentation on labeled images and produce highly perturbed ones, which makes it non-trivial to achieve the PA task, thus enhancing model generalibility. We also apply data augmentation on unlabeled samples to perform consistency learning. We apply light and strong transformations on each unlabeled target image, generating a lightly perturbed version and a strongly perturbed one. Then, pseudo label inferred from the former can be applied on the latter in a supervised learning manner. These procedures cast a consistency regularization on the model, smoothing its predictions over image changes, and thus contributing to more desirable results. Experiments show that PACL significantly outperforms state-of-the-art SSDA methods and consistently boosts various UDA methods for adaptation performance with large gains.

In summary, the contributions of this paper are as follows: (1) We propose a general framework that can significantly advance state-of-the-art SSDA performance and flexibly integrate various UDA methods to have adaptation performance boosted by large margins. (2) We propose the PA module which achieves class-wise domain alignment by assigning source samples to the target prototype that from the same class. (3) We propose to enhance DA performance with a novel data augmentation based solution where we apply data augmentation on both labeled and unlabeled images, aiming to address model overfitting and enforce consistency regularization, respectively.

2 Related Work

Domain Adaptation (DA). According to the type of data available in target domain, DA methods can be divided into three categories: Unsupervised Domain Adaptation (UDA), Few-Shot Domain Adaptation (FSDA) and Semi-Supervised Domain Adaptation (SSDA). UDA assumes that target domain data are purely unlabeled. Early methods in the shallow regime address UDA either by reweighting source instances (Huang et al., 2006; Gong et al., 2013) or projecting samples into a domain invariant feature space (Gong et al., 2012; Pan et al., 2011). Recent ones are more in the deep regime and approach UDA by moment matching (Long et al., 2015; Carlucci et al., 2017; Long et al., 2017) or adversarial learning (Tzeng et al., 2015; Ganin et al., 2016; Luo et al., 2017; Long et al., 2018). FSDA assumes that there is no access to unlabeled samples, but a few labeled ones in target domain. To fully utilize the few labeled target samples, existing methods perform class-wise domain alignment using contrastive loss (Motiian et al., 2017) or triplet loss (Xu et al., 2019b). SSDA is a hybrid of FSDA and UDA where we have access to both a few labeled samples and many unlabeled samples from target domain. Early works use the extra labeled target samples to help more precisely measure the data distribution mismatch between source and target domains, either based on Maximum Mean Discrepancy (MMD) or domain invariant subspace learning (Ao et al., 2017; Donahue et al., 2013; Yao et al., 2015; Saito et al., 2019; Tejankar & Pirsiavash, 2019). Saito et al. recently proposed a deep learning based method which alternates between maximizing the classification entropy with respect to the classifier and minimizing it with respect to the feature encoder (Saito et al., 2019). Kim et al. extended this work by explicitly alleviating the intra-domain discrepancy problem (Kim & Kim, 2020). We approach SSDA in a new way by proposing a general framework into which existing UDA methods can be incorporated and served as one component for domain alignment along with other novel components.

Semi-Supervised Learning (SSL). Leveraging unlabeled data along with labeled ones in the training process, SSL has boosted performance with a variety of training strategies, including graph-

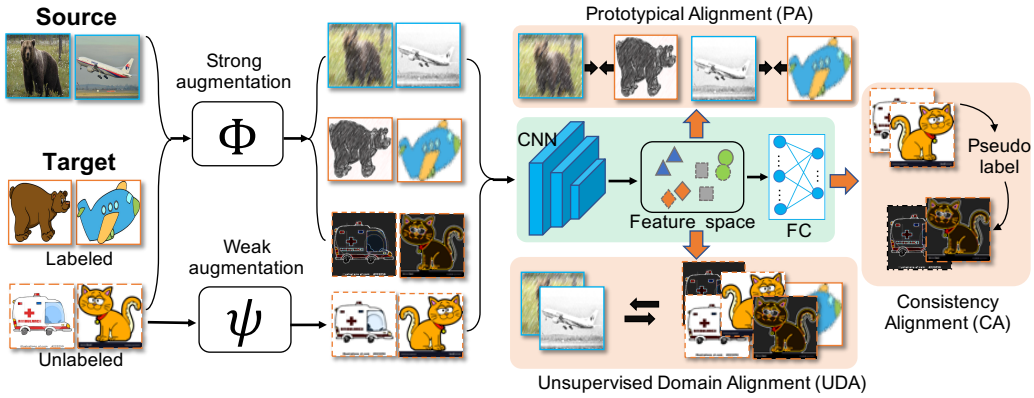

Figure 1: Illustration of the proposed PACL framework. PACL includes three alignment modules: UDA performs domain alignment using labeled source and unlabeled target samples, exactly the same way as existing UDA methods. PA conducts class-level alignment by explicitly pushing close cross-domain samples that are from the same class. CA generates a pseudo label for each unlabeled target sample from its weakly augmented version and applies the pseudo label on its strongly augmented version with supervised learning.

based (Kipf & Welling, 2016), adversarial (Miyato et al., 2016), generative (Dai et al., 2017), model-ensemble (Laine & Aila, 2016), self-training (Li et al., 2019; Sohn et al., 2020), etc. The difference of SSL and SSDA lies that labeled samples of SSL are from the same domain as the unlabeled ones and they are usually of a fair amount, often up to several hundreds or thousands. In SSDA, labeled samples instead come from two different domains, with the majority being out-of-domain (relative to the unlabeled samples). So, compared with SSL, SSDA needs further solving the domain shift problem to more effectively utilize the plenty yet out-of-domain labeled samples. We achieve this by employing off-the-shelf UDA techniques and the proposed prototypical alignment technique.

Few-Shot Learning (FSL). FSL aims to acquire knowledge of *novel classes* with only a few labeled samples (Vinyals et al., 2016; Snell et al., 2017). FSL has very distinct goals from SSDA. FSL emphasizes generalizability of a learned model towards novel classes for which there is no sample (neither labeled nor unlabeled) available during training but a few labeled ones in test. SSDA instead focuses on enhancing generalizability of a model towards unlabeled samples of the classes for which during training there are plenty of labeled samples from source domain, a few labeled ones and many unlabeled ones from target domain. Even with different goals, the way that an FSL method (Snell et al., 2017) utilizes a few labeled samples to recognize other unlabeled ones inspires us to develop the supervised alignment module which achieves class-level domain alignment.

3 Algorithm

Semi-supervised domain adaptation (SSDA) investigates the adaptation from a label-rich source dataset $\mathcal{S} = \{\mathcal{X}_s, \mathcal{Y}_s\}$ to a label-scarce target dataset $\mathcal{T}$. The two datasets are drawn from the same label space $\{1, 2, \ldots, C\}$ but with different data distributions that cause domain shifts. Unlike Unsupervised Domain Adaptation (UDA) problem where the target dataset $\mathcal{T}$ consists of purely unlabeled samples, in SSDA $\mathcal{T}$ is a hybrid of labeled and unlabeled samples, i.e., $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$, where $\mathcal{T}_l = \{\mathcal{X}_t, \mathcal{Y}_t\}$ and $\mathcal{T}_u = \mathcal{X}_u$. Usually, the number of labeled samples in $\mathcal{T}_l$ is very small, even to one sample per class in the most extreme case. We call these labeled samples as "landmarks". Our goal is to learn a domain adaptive model using $\mathcal{S}$, $\mathcal{T}_u$ as well as the landmarks from $\mathcal{T}_l$, and evaluate the model on $\mathcal{T}_u$. Let the model be $h = f \circ g$ with parameters θ, where f generates features from images and g outputs label predictions based on the extracted features.

We can see that the difference of SSDA from UDA is the extra access to landmarks. A naive way to use them is to optimize a cross-entropy loss jointly with an unsupervised alignment loss:

$$\mathcal{L}_{uda} = \mathcal{L}_{ce} + \alpha \mathcal{L}_{ua}, \tag{1}$$

where

$$\mathcal{L}_{ce} = \mathop{\mathbb{E}}_{(\mathbf{s}, y_s) \sim \mathcal{S}} [L(h(\mathbf{s}), y_s)] + \mathop{\mathbb{E}}_{(\mathbf{t}, y_t) \sim \mathcal{T}_l} [L(h(\mathbf{t}), y_t)] \tag{2}$$

is the cross-entropy loss over labeled source samples and target landmarks. $\mathcal{L}_{ua}$ is an UDA loss that exploits unlabeled target samples and labeled source samples for domain alignment.

Naively merging the limited landmarks into source samples for the cross-entropy loss optimization does not fully realize their value, as their contribution would be severely diluted, resulting in a model biased towards source domain. We solve this by proposing a prototypical domain alignment module, which aligns domains in class-level by explicitly pushing source samples towards target samples that from the same class. This module is further enhanced by a data augmentation based technique. Moreover, we propose another data augmentation based technique that is applied on unlabeled target samples. This technique constrains the model to make consistent predictions for different versions of the same image undergone perturbations of different extent. Figure 1 shows our framework.

3.1 Prototypical Alignment

The labeled samples from target domain, i.e., landmarks, though limited, provide reliable (relative to pseudo labels produced by various thresholding techniques (Zou et al., 2019)) class information about target domain. We can utilize these landmarks to achieve a class-level domain alignment, which complements with the domain-level alignment fulfilled by existing UDA methods.

We calculate a target representation, or a target prototype (Snell et al., 2017) for each class by averaging the feature embeddings of the landmarks from that class:

$$\mathbf{c}_k = \frac{1}{|\mathcal{T}_k|} \sum_{(\mathbf{t}_i, y_i) \in \mathcal{T}_k} f(\mathbf{t}_i), \tag{3}$$

where $\mathcal{T}_k$ is the landmark collection from class k. With the target prototypes for all classes $\{\mathbf{c}_k\}_{k=1}^{C}$, we can compute a distribution over classes for a source sample $(\mathbf{s}, y_s)$ based on a softmax over distances to the target prototypes in the embedding space:

$$P(y_s = y|\mathbf{s}) = \frac{\exp(f(\mathbf{s}) - \mathbf{c}_y)}{\sum_{k=1}^{C} \exp(f(\mathbf{s}) - \mathbf{c}_k)}. \tag{4}$$

Then, we can calculate the prototypical alignment loss over all source samples as

$$L_{pa} = \mathop{\mathbb{E}}_{(\mathbf{s}, y_s) \sim \mathcal{S}} \Big[\frac{1}{C} \sum_{y=1}^{C} y \log[-p(y_s = y|\mathbf{s})] \Big]. \tag{5}$$

This learning objective encourages the model to produce features that maintain class discrimination regardless domain shift, and consequently the learned model is more likely to be domain invariant.

3.2 Improving Domain Alignment with Data Augmentation

To further alleviate the scarcity of landmarks, we propose two data augmentation based domain alignment techniques, one for labeled samples and the other for unlabeled samples.

3.2.1 Data Augmentation for Prototypical Alignment

It has been shown that strong augmentation that creates highly perturbed images can lead to significant performance gains in supervised learning (Cubuk et al., 2019a;b). This technique has also been proved effective in semi-supervised learning (Sohn et al., 2020; Xie et al., 2019). Inspired by this, we introduce strong data augmentation to address DA problem. We employ a data augmentation method called RandAugment (Cubuk et al., 2019b), which applies on an image with random augmentation techniques sampled from a transformation set, including color, brightness, contrast adjustments, rotation, polarization, etc.

For each labeled sample from both source and target domains $(\mathbf{s}, y_s) \in \mathcal{S}$ and $(\mathbf{t}, y_t) \in \mathcal{T}_l$, we perform strong data augmentation and obtain $\mathcal{S}'$ and $\mathcal{T}'$ that consist of highly perturbed images. With $\mathcal{S}'$ and $\mathcal{T}'$, the enhanced PA learning objective turns

$$L'_{pa} = \mathop{\mathbb{E}}_{(\mathbf{s}', y_s) \sim \mathcal{S}'} \Big[\frac{1}{C} \sum_{y=1}^{C} y \log[-P(y_s = y|\mathbf{s}')] \Big], \tag{6}$$

where $P(y_s = y|\mathbf{s}')$ are calculated with Eq. (4), and the target prototypes are calculated with $\mathcal{T}'$.

Strong data augmentation produces a wider range of highly perturbed images, which makes the model harder to memorize the few landmarks and therefore enhances the generalizability of the learned model. On the one hand, the model is forced to be insensitive to more diverse changes or perturbations in the image space. On the other hand, the above PA module in essence optimizes the model extracting image features that the intra-class ones are of higher similarity than the inter-class ones across domains. It is harder for the model to achieve this optimization objective with highly perturbed images. Thus, the model is encouraged to mine the most discriminative class semantics from images and at the same time the most subtle distinctions among images from different classes.

3.2.2 CONSISTENCY ALIGNMENT

Inspired by the recent success in semi-supervised learning (Sohn et al., 2020; Xie et al., 2019), we introduce data augmentation to address SSDA and propose the Consistency Alignment (CA) module. For each unlabeled target sample $\mathbf{u} \in \mathcal{U}$, we apply both weak data augmentation ψ and strong data augmentation Φ:

$$\mathbf{u}_w = \psi(\mathbf{u}), \tag{7}$$

$$\mathbf{u}_s = \Phi(\mathbf{u}). \tag{8}$$

The weak data augmentation ψ includes the widely-used image flipping and image translation. Same as the practice for labeled samples, we use RandAugment (Cubuk et al., 2019b) as our strong data augmentation Φ. We feed $\mathbf{u}_s$ and $\mathbf{u}_w$ to our model h, and optimize the following objective function:

$$L_{ca} = \mathop{\mathbb{E}}_{\mathbf{u}\sim\mathcal{U}} \big[\mathbb{1}(\max(\mathbf{p}_w) \geq \sigma) H(\tilde{\mathbf{p}}_w, \mathbf{p}_s\big], \tag{9}$$

where $\mathbf{p}_s$ and $\mathbf{p}_w$ are the classification probabilities of augmented samples $\mathbf{u}_s$ and $\mathbf{u}_w$, respectively. $\tilde{\mathbf{p}}_w = \arg\max(\mathbf{p}_w)$ transforms the classification probability into a one-hot vector; $H(.,.)$ is the cross-entropy of two possibility distributions; $\max(\mathbf{p}_w)$ returns the highest possibility score.

In essence, the above CA module computes a pseudo label for an unlabeled sample with its weakly-augmented version and applies the pseudo label on its strongly-augmented version by computing the standard cross-entropy loss. To mitigate the impact of incorrect pseudo labels, only the samples with confident predictions (the highest probability scores are above a threshold) are used for loss computation. This introduces a form of consistency regularization, enforcing the model to be insensitive to the image perturbations and hence being stronger in classifying unlabeled images.

Pseudo labeling (or self-training) has been investigated before for domain adaptation (Pan et al., 2019; Zou et al., 2019), but our method is clearly distinct from the previous ones. Existing methods usually perform stage-wise pseudo labeling: Each stage consists of a number of training epochs and the latest model is deployed on unlabeled samples at the end of each stage. The confidently predicted samples are selected for model training in the next stage usually in the same way as labeled samples from source domain. Within all training epochs in each stage, the pseudo-labeled samples remain unchanged. Our method instead performs mini-batch-wise pseudo labeling: In each mini-batch, we compute a pseudo label for every sample from its weakly-augmented version and apply the pseudo label on the strongly-augmented one. We discard all of the pseudo labels after each mini-batch, which alleviates the harmful impact of incorrect pseudo labels.

The overall learning objective of our method is a weighted combination of the UDA loss, the PA loss, and the CA loss:

$$L = L_{uda} + \lambda_1 L'_{pa} + \lambda_2 L_{ca}, \tag{10}$$

where λ_1 and λ_2 are the hyper-parameters.

4 EXPERIMENTS

Datasets and evaluation protocols. We conduct experiments on three commonly used datasets, namely, *VisDA2017* (Peng et al., 2017), *DomainNet* (Peng et al., 2019), and *Office-Home* (Venkateswara et al., 2017).

DomainNet consists of 6 domains of 345 categories. Following (Saito et al., 2019), we select the *Real* (R), *Clipart* (C), *Painting* (P), and *Sketch* (S) as the 4 evaluation domains and perform the

	Net	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
ST	AlexNet	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
DANN	AlexNet	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.6	57.3	40.4	42.4
ADR	AlexNet	43.1	46.2	41.4	44.4	39.3	43.6	32.8	36.4	33.1	38.9	29.1	32.4	55.9	57.3	39.2	42.7
CDAN	AlexNet	46.3	46.8	45.7	45.0	38.3	42.3	27.5	29.5	30.2	33.7	28.8	31.3	56.7	58.7	39.1	41.0
ENT	AlexNet	37.0	45.5	35.6	42.6	26.8	40.4	18.9	31.1	15.1	29.6	18.0	29.6	52.2	60.0	29.1	39.8
MME	AlexNet	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
FAN	AlexNet	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
PACL	AlexNet	**55.8**	**62.6**	**54.0**	**59.0**	**56.1**	**60.5**	**46.1**	**50.6**	**54.6**	**50.3**	**45.0**	**48.4**	**62.3**	**67.4**	**52.8**	**57.6**
ST	ResNet-34	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
DANN	ResNet-34	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
ADR	ResNet-34	57.1	60.7	61.3	61.9	57.0	60.7	51.0	54.4	56.0	59.9	49.0	51.1	72.0	74.2	57.6	60.4
CDAN	ResNet-34	65.0	69.0	64.9	67.3	63.7	68.4	53.1	57.8	63.4	65.3	54.5	59.0	73.2	78.5	62.5	66.5
ENT	ResNet-34	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
MME	ResNet-34	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
FAN	ResNet-34	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
PACL	ResNet-34	**75.3**	**79.0**	**74.1**	**77.3**	**75.3**	**79.4**	**65.0**	**70.6**	**72.1**	**74.6**	**68.1**	**71.6**	**79.7**	**82.4**	**72.8**	**76.4**

Table 1: Results on the *DomainNet* dataset.

following cross-domain evaluations: R→C (adaptation from source *Real* to target *Clipart*), R→P, P→C, C→S, S→P, R→S, and P→R. For each set of cross-domain experiment, we evaluate both 1-shot and 3-shot settings, where there are 1 and 3 labeled target samples, respectively. The labeled samples are randomly selected and we use the provided splits for experiments. We evaluate the classification accuracy for all the 7 sets of experiments and also report the mean of the accuracies.

VisDA2017 includes 152,397 synthetic and 55,388 real images from 12 categories. To conduct SSDA evaluation, we randomly select 1 and 3 real images from each of the 12 classes as the landmarks, which corresponds to the 1-shot and 3-shot evaluation settings, respectively. Following the precious UDA methods, we report the per-class classification accuracy and also the Mean Class Accuracy (MCA) over all classes.

Office-Home contains images of 65 categories that are from 4 different domains, namely, *Real* (R), *Clipart* (C), *Art* (A), and *Product* (P). We use the same 1-shot and 3-shot splits as (Saito et al., 2019) and evaluate adaptation performance for all 12 pairs of domains. We report the classification accuracies for all the experimental sets and the corresponding mean accuracy.

Implementation details. The proposed PACL is a general SSDA framework that can incorporate most existing UDA methods and leverage landmark samples to improve adaptation performance. Depending on the UDA method built upon, we can get different PACL variants. But for ease and fairness of evaluation, we conducted most of our experiments with the PACL variant that is based on MME (Saito et al., 2019)[1]. Note that although MME is proposed for SSDA, it can be viewed as an UDA method that naively merges labeled target samples into labeled source samples for the cross-entropy loss optimization. Since we do this in the same way, MME is still compatible with our framework. We adopt exactly the same training procedures and hyper-parameters as MME, except the way to sample labeled data in each mini-batch. Rather than naively sampling data across the whole labeled set, we perform class-balanced sampling: In each mini-batch, we randomly sample M classes with N_s and N_t images for each class from source and target domains, respectively. We set M =10, N_s =10, N_t =1 for 1-shot setting, and N_t =3 for 3-shot setting. We set the balancing hyper-parameters of different losses as $\lambda_1 = 0.1$ and $\lambda_2 = 1$ in Eq. (10), and the confident threshold $\sigma = 0.8$ for pseudo labeling (Eq. (9)) for all experiments.

4.1 Comparative Results

We compare with the following methods, DANN (Ganin et al., 2016), ADR (Saito et al., 2017), CDAN (Long et al., 2018), ENT (Grandvalet & Bengio, 2005), MME (Saito et al., 2019), and FAN (Kim & Kim, 2020). All these methods are either specifically designed or tailored to address the SSDA problem to make fair comparison. We also report results of the baseline method "ST" which trains models with labeled samples from source and target domains, without domain alignment.

DomainNet. Table 1 shows that PACL significantly outperforms existing methods in all the experimental settings. Specifically, with AlexNet as the feature extraction model, PACL attains 8.6 and 9.4 point gains for the 1-shot and 3-shot settings, respectively, over MME which PACL is based on. With ResNet-34, the improvements are 6.4 and 7.5 for the 1-shot and 3-shot settings, respec-

[1]Unless otherwise specified, we use "PACL" to represent this variant in short.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	MCA
1-shot													
ST	82.6	52.8	75.0	57.6	72.7	39.7	80.5	53.3	59.0	64.1	77.5	12.6	60.6
MME	86.6	60.1	80.8	61.9	84.0	69.6	87.0	72.4	73.0	50.9	79.4	**14.7**	68.4
PACL	**94.9**	**81.5**	**88.9**	**81.3**	**95.9**	**92.4**	**92.2**	**83.3**	**95.2**	**77.4**	**88.4**	2.3	**81.1**
3-shot													
ST	74.0	71.7	71.2	64.7	78.5	71.8	69.6	51.4	73.7	49.4	80.8	19.8	64.7
MME	87.2	67.3	74.9	64.5	86.9	85.5	78.8	75.8	84.4	48.0	80.8	**19.9**	71.2
PACL	**95.9**	**82.9**	**88.6**	**84.9**	**95.9**	**92.1**	**93.3**	**83.7**	**95.4**	**79.3**	**88.0**	19.5	**83.3**

Table 2: Results on the *VisDA2017* dataset.

	1-shot	3-shot
S+T	44.1	50.0
DANN	45.1	50.3
ADR	44.5	49.5
CDAN	41.2	46.2
ENT	38.8	50.9
MME	49.2	55.2
FAN	-	55.6
PACL	**52.4**	**59.1**

Table 3: Results on *Office-Home*.

Method	AlexNet		ResNet-34	
	1-shot	3-shot	1-shot	3-shot
ST	29.1	33.3	46.3	50.1
MME	33.3	37.9	61.0	61.9
MME + PA	37.3	39.2	62.4	63.8
MME + PA + SA	39.7	42.2	65.2	67.6
MME + PA + SA + CA	45.0	48.4	68.1	71.6

Table 4: Ablation study for the adaptation from *Real* to *Sketch* on the *DomainNet* dataset.

tively. PACL also performs significantly better than the most recent method FAN in both settings with both backbone networks. These results substantiate the effectiveness of PACL on mitigating domain shifts by comprehensively exploring the few labeled target samples.

VisDA2017. We can see from Table 2 that PACL, built upon on MME, reaches significant gains over MME. The average gains over all class are 12.7 for the 1-shot setting and 12.1 for the 3-shot setting, with the peak gain 31.3 points reached in the *sktbrd* class for the 3-shot setting. These tremendous improvements convincingly evidence the effectiveness of PACL.

Office-Home. We show in Table 3 the mean accuracy over all the 12 sets of cross-domain evaluation experiments for the 1-shot and 3-shot settings. The complete table with results for each set of experiment can be found in the Appendix. From Table 3, we can see that PACL attains remarkable performance boosts over existing methods as well, although the gains are not as significant as those on the other two datasets.

4.2 Additional Empirical Analysis

Ablation study. Built upon existing UDA methods, PACL includes the following new modules/techniques to address the SSDA problem, namely, the Prototypical Alignment (PA) module, the Strong Augmentation (SA) technique that aims to enhance PA and the Consistency Alignment (CA) module. Table 4 shows the ablation study on the adaptation from *Real* to *Sketch* on the *DomainNet* dataset. We can see that all the new modules/techniques contribute to the ultimate performance promotions, thus substantiating their effectiveness.

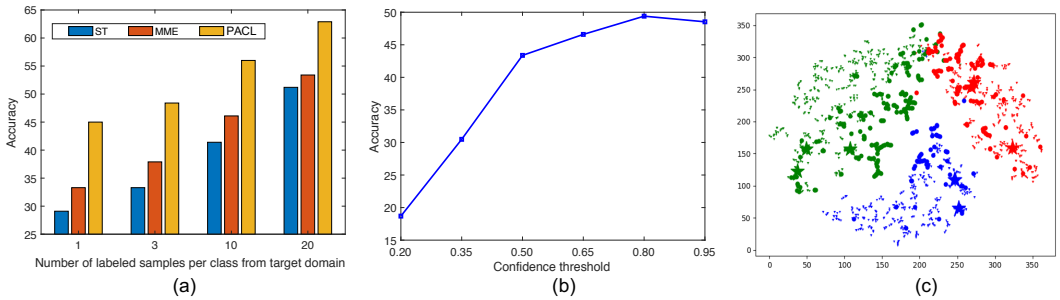

Figure 2: (a): Accuracy with different numbers of labeled samples per class in target domain. (b): Sensitivity analysis with respect to the confident threshold σ (Eq. (9)). (c): t-SNE visualization of the learned features of 3 randomly selected classes. Different colors represent different classes. Shapes "Y", "★" and "●" represent source samples, labeled target samples, and unlabeled target samples, respectively.

Method	Net	Setting	MCA	
Source-only	ResNet-101	UDA	52.4	
HAFN	ResNet-101	UDA	73.9	
SAFN	ResNet-101	UDA	76.1	
			1-shot	3-shot
HAFN + ST	ResNet-101	SSDA	77.0	79.3
SAFN + ST	ResNet-101	SSDA	77.5	79.2
PACL (HAFN)	ResNet-101	SSDA	**83.9**	**85.3**
PACL (SAFN)	ResNet-101	SSDA	83.3	84.5
ST	ResNet-34	SSDA	60.6	64.7
MME	ResNet-34	SSDA	68.4	71.2
PACL (MME)	ResNet-34	SSDA	**81.1**	**83.3**

Table 5: Flexibility evaluation on *VisDA2017*.

Plug-and-play evaluation. As mentioned above, PACL is agnostic to UDA methods built upon. To evaluate this, we apply PACL as a plug-and-play component on different existing UDA methods and see how much adaption performance can be improved. Table 5 shows the results of three UDA methods before and after empowered by PACL, namely HAFN (Xu et al., 2019a), SAFN (Xu et al., 2019a), and MME (Saito et al., 2019). Note that to rigorously evaluate the effectiveness of PACL, we adopt exactly the same training procedures as the UDA methods when we apply PACL on them. We can see from Table 5 that PACL indeed significantly promotes the performance of existing UDA methods and their naive SSDA extensions (e.g., "HAFN+ST" stands for training HAFN additionally with a few labeled target samples for the supervised learning part). For example, "PACL (HAFN)" raises the result of the UDA method HAFN from 73.9 to 83.9 with one sample per class labeled and further to 85.3 with 3 samples per class labeled. These significant and consistent improvements with different UDA methods, different backbone networks and different numbers of landmarks convincingly substantiate the effectiveness of PACL on boosting adaptation performance with minimal labeling effort.

Impact of the number of landmarks. We have shown the superior performance of the standard 1-shot and 3-shot settings above. We wonder how performance changes with the number of landmarks increased. We study this on the adaptation from *Real* to *Sketch* on *DomainNet*. We can see from Fig. 2 (a) that all the three methods enjoy performance boosts with more target samples labeled. Comparatively speaking, PACL consistently reaches the best performance for all the cases. This substantiates the benefit of our method of flexibly exploiting different amount of labeled target samples to help address the domain shift problem.

Sensitiveness with parameters. A common drawback of pseudo labeling based methods is that it is always non-trivial to select the proper confident threshold above which a label prediction is regarded as reliable. However, this problem is less severe in our method because incorrect predictions are not reused, so that errors shall not be accumulated. Existing pseudo labeling methods usually maintain a confident pseudo label set that is usually progressively enlarged every several training epochs and is reused immediately thereafter. We instead generate pseudo labels on the fly in each mini-batch and discard all of them once finished. So, the harmful impact of incorrect pseudo labels can be diluted with a wider range of classification predictions. We conduct a sensitivity analysis on the adaptation from *Real* to *Sketch* on the *DomainNet* dataset under the 3-shot setting. The result is shown in Figure 2 (b). We can see that our method is not very sensitive to the confidence threshold and maintains a fair performance as long as the threshold is pretty high.

Feature visualization. To qualitatively evaluate the alignment results, we plot the t-SNE (Maaten & Hinton, 2008) visualization of the features produced by PACL for the adaptation from *Real* to *Sketch* on *DomainNet* in the 3-shot setting. We can see from Figure 2 (c) that the learned features exhibit favorable clustering structure. Features from different domains lie closely if they belong to the same classes while being apart otherwise. This plot further supports that both feature discriminabilty and domain alignment are well fulfilled by the learned model.

5 Conclusion

We propose in this paper a novel semi-supervised domain adaptation (SSDA) framework within which existing unsupervised domain adaptation (UDA) methods can effectively utilize a few labeled samples from target domain to further mitigate domain shifts. The proposed framework includes a Prototypical Alignment (PA) module which achieves class-wise domain alignment. The PA module is further enhanced by a data augmentation based technique that produces highly perturbed images to mitigate overfitting. A Consistency Alignment (CA) module is incorporated into the framework which enforces consistency regularization on the classifier. Experiments show that the proposed framework reaches state-of-the-art SSDA performance and consistently promotes adaptation performance of various UDA methods with diverse numbers of labeled target samples. Ablation study verifies the contributing roles of all the novel alignment components.

REFERENCES

Shuang Ao, Xiang Li, and Charles X Ling. Fast generalized distillation for semi-supervised domain adaptation. In *AAAI*, 2017. 1, 2

Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, 2017. 2

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019a. 2, 4

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019b. 4, 5

Zihang Dai, Zhilin Yang, Fan Yang, William W. Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad GAN. In *NIPS*, 2017. 3

Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, 2013. 1, 2

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 11

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016. 2, 6

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 1, 2

Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, 2013. 2

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005. 6

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *NeurIPS*, 2006. 2

Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 2020. 1, 2, 6

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, 2016. 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 2

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016. 3

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*, 2019. 3

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 2

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017. 2

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. 1, 2, 6

Zelun Luo, Yuliang Zou, Judy Hoffman, and Fei-Fei Li. Label efficient learning of transferable representations acrosss domains and tasks. In *NeurIPS*, 2017. 2

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 8

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing by virtual adversarial examples. In *ICLR*, 2016. 3

Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017. 2

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Networks*, 22(2):199–210, 2011. 1, 2

Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, 2019. 5

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 5

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *ICLR*, 2017. 6

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

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019. 1, 2, 5, 6, 8, 11, 12

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 3, 4

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. 2, 3, 4, 5

Ajinkya Tejankar and Hamed Pirsiavash. A simple baseline for domain adaptation using rotation prediction. *CoRR*, abs/1912.11903, 2019. 2

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 1, 2

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 5

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 3

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 2, 4, 5

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, 2019a. 8, 11

Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *CVPR*, 2019b. 2

Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015. 1, 2

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *CVPR*, 2019. 4, 5

Algorithm 1. Proposed SSDA framework

Input: Source set $\mathcal{S} = \{\mathcal{X}_s, \mathcal{Y}_s\}$, labeled target set $\mathcal{T}_l = \{\mathcal{X}_t, \mathcal{Y}_t\}$ and unlabeled target set $\mathcal{U} = \{\mathcal{X}_u\}$.
Output: Domain adaptative model h.

while not done **do**

1. Sample from $\mathcal{S} \cup \mathcal{T}_l$ labeled images $\mathcal{B}_l = \{\{\mathbf{s}_{i,j}\}_{i=1}^{N_s}, \{\mathbf{t}_{i,j}\}_{i=1}^{N_t}, y_j\}_{j=1}^{M}$. and sample from $\mathcal{U}$ unlabeled images $\mathcal{B}_u = \{\mathbf{u}_i\}_{i=1}^{N_u}$. We further denote $\mathcal{B}_l = \mathcal{B}_s \cup \mathcal{B}_t$ with $\mathcal{B}_s = \{\{\mathbf{s}_{i,j}\}_{i=1}^{N_s}, y_j\}_{j=1}^{M}$ and $\mathcal{B}_t = \{\{\{\mathbf{t}_{i,j}\}_{i=1}^{N_t}, y_j\}_{j=1}^{M}$.
2. Apply strong data augmentation Φ on $\mathcal{B}_s$, $\mathcal{B}_t$ and $\mathcal{B}_u$, producing $\mathcal{B}'_s = \Phi(\mathcal{B}_s)$, $\mathcal{B}'_t = \Phi(\mathcal{B}_t)$ and $\mathcal{B}^s_u = \Phi(\mathcal{B}_u)$. Meanwhile, apply weak data augmentation ψ on $\mathcal{B}_u$ and get $\mathcal{B}^w_u = \psi(\mathcal{B}_u)$.
3. Calculate target prototypes $\{\mathbf{c}_k\}_{k=1}^{M}$ for each class from $\mathcal{B}'_t$ using Eq. (3).
4. Calculate prototypical loss with $\{\mathbf{c}_k\}_{k=1}^{M}$ using Eq. (6).
5. Calculate cross-entropy loss with $\mathcal{B}'_s$ and $\mathcal{B}'_t$ using Eq. (2).
6. Calculate unsupervised alignment loss with $\mathcal{B}'_s$ and $\mathcal{B}^w_u$ using an existing UDA method.
7. Calculate self-supervised alignment loss with $\mathcal{B}^s_u$ and $\mathcal{B}^w_u$ using Eq. (9).
8. Optimize the model h using Eq. (10).

end while

A Appendix

A.1 Implementation details

Our Semi-Supervised Domain Adaptation (SSDA) framework is general such that any existing unsupervised domain adaptation (UDA) method can be incorporated and has the adaptation performance improved. We verify this by employing three UDA methods, i.e., MME (Saito et al., 2019), HAFN (Xu et al., 2019a) and SAFN (Xu et al., 2019a), which results in three variants of our methods, "PACL (MME)", "PACL (HAFN)", and "PACL (SAFN)", respectively. To make fair comparison, we use the official implementations[2] of the three UDA methods and follow strictly the same training rules and procedures, except the mini-batch sampling strategy. We have introduced the implementation details specific to our framework in the main text. Here we give a more complete description about the implementation of each of our variant.

PACL (MME): Following MME, we use the ImageNet pre-trained models to initialize both the AlexNet and Resnet-34 backbones. For AlexNet, the last linear layer are replaced with randomly initialized linear layer. For ResNet34, the last linear layer are replaced with two fully-connected layers. The models are optimized with the momentum optimizer. We set the initial learning rate as 0.01 for all fully-connected layers while use a smaller learning rate 0.001 for other layers including convolution layers and batch-normalization layers. The learning rate decay policy is the same as in (Ganin & Lempitsky, 2015). Each mini-batch consists of labeled source, labeled target and unlabeled target images. As mentioned in the main text, we perform class balanced sampling for labeled images. For unlabeled images, following MME, we sample in each mini-batch 24 images for ResNet-34 and 32 for AlexNet.

PACL (HAFN). We initialize our backbone with ImageNet pre-trained model. Following HAFN, we perform two-step training, where in the first step we train the model with only labeled data from the source domain; in the second step, we conduct domain adaptive training. For both steps, SGD is employed for model optimization and the learning rate is 10^{-3}. We train both stages with 10 epochs.

PACL (SAFN): Everything is same as "PACL (HAFN)" except the UDA term is a different one. So, the implementation details are almost the same, except that we perform only adaptive training and the model is trained with 50 epochs.

A.2 Pseudo Code

Algorithm 1 outlines the main step of the proposed method.

[2]MME: https://github.com/VisionLearningGroup/SSDA_MME. HAFN and SAFN: https://github.com/jihanyang/AFN.git.

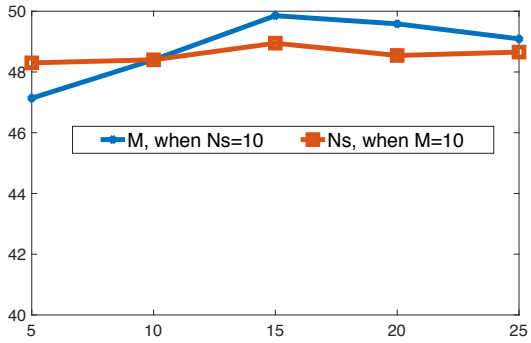

Figure 3: Analysis of the number of class and number of source sample per class in each mini-batch.

	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
One-shot													
S+T	37.5	63.1	44.8	54.3	31.7	31.5	48.8	31.1	53.3	48.5	33.9	50.8	44.1
DANN	42.5	64.2	45.1	56.4	36.6	32.7	43.5	34.4	51.9	51.0	33.8	49.4	45.1
ADR	37.8	63.5	45.4	53.5	32.5	32.2	49.5	31.8	53.4	49.7	34.2	50.4	44.5
CDAN	36.1	62.3	42.2	52.7	28.0	27.8	48.7	28.0	51.3	41.0	26.8	49.9	41.2
ENT	26.8	65.8	45.8	56.3	23.5	21.9	47.4	22.1	53.4	30.8	18.1	53.6	38.8
MME	42.0	69.6	48.3	58.7	37.8	34.9	52.5	36.4	57.0	54.1	39.5	59.1	49.2
PACL	**50.3**	**70.71**	**52.2**	**61.4**	**41.2**	**39.3**	**57.8**	**39.1**	**59.1**	**55.8**	**41.7**	**59.9**	**52.4**
Three-shot													
S+T	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
DANN	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
ADR	45.0	66.2	46.9	57.3	38.9	36.3	57.5	40.0	57.8	53.4	37.3	57.7	49.5
CDAN	41.8	69.9	43.2	53.6	35.8	32.0	56.3	34.5	53.5	49.3	27.9	56.2	46.2
ENT	44.9	70.4	47.1	60.3	41.2	34.6	60.7	37.8	60.5	58.0	31.8	63.4	50.9
MME	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
FAN	51.9	74.6	51.2	61.6	47.9	42.1	65.5	44.5	60.9	58.1	44.3	64.8	55.6
PACL	**55.4**	**75.7**	**56.0**	**67.0**	**52.5**	**46.4**	**67.4**	**48.5**	**66.3**	**60.8**	**45.9**	**67.3**	**59.1**

Table 6: Results on the *Office-Home* dataset.

A.3 Analysis of Sampling Strategies

As mentioned in the main text and also outlined in **Algorithm 1**, we perform class-balanced sampling for labeled data: In each mini-batch, we sample M classes with N_s source images and N_t target images for each class, which results in a mini-batch of size $M * (N_s + N_t)$. We study the impact of batch size to the performance by varying these numbers. Since N_t is a constant for the 1-shot setting ($N_t = 1$) and 3-shot setting ($N_t = 3$), we only vary the values of M and N_s. When evaluating one parameter, we fix the other one. The results of applying our framework on MME (Saito et al., 2019), i.e., "PACL (MME)"" on the adaptation experiment between *Real* and *Sketch* from the *DoaminNet* datasets are shown in Figure 3. We can see from this figure that the proposed method is quite robust with the two hyper-parameters. There are fairly small change (less than 2%) when their values are changed from 5 to 25.

A.4 Complete results on Office Home

Table 6 shows the complete results on the *office-home* dataset, where we can see that PACL reaches the best performance for all the sets of adaptation experiments.

A.5 Complete results on VisDA2017 for flexibility analysis

Table 7 shows the complete results on the *VisDA2017* dataset for flexibility analysis. We can see that PACL consistently improves various existing UDA methods by large margins with different backbone networks and different number of landmark samples.

A.6 Code

Attached in the supplementary material is the code for the proposed PACL framework. Following the instructions there can reproduce our results.

Method	Net	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	MCA
UDA														
Source-only	ResNet-101	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN	ResNet-101	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
DANN	ResNet-101	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
MCD	ResNet-101	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
HAFN	ResNet-101	92.7	55.4	82.4	70.9	93.2	71.2	90.8	78.2	89.1	50.2	88.9	24.5	73.9
SAFN	ResNet-101	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SSDA (1-shot)														
HAFN + ST	ResNet-101	92.6	64.6	88.0	66.3	93.6	84.6	90.8	**80.9**	90.8	60.1	88.0	24.5	77.0
SAFN + ST	ResNet-101	95.5	64.0	80.1	63.2	93.0	87.3	91.1	78.9	90.0	66.6	89.4	**30.6**	77.5
PACL (HAFN)	ResNet-101	**97.4**	**78.7**	**90.0**	86.6	**97.7**	90.2	94.2	78.0	**93.8**	80.7	93.8	25.2	**83.9**
PACL (SAFN)	ResNet-101	96.7	76.5	88.4	**90.3**	97.1	**91.9**	**95.4**	74.9	91.6	**82.9**	**94.8**	19.5	83.3
SSDA (3-shot)														
HAFN + ST	ResNet-101	93.8	77.9	**87.9**	68.6	94.2	88.8	92.0	82.3	91.1	62.1	82.8	30.6	79.3
SAFN + ST	ResNet-101	95.4	75.5	83.0	67.9	94.4	87.6	88.5	**78.6**	93.2	66.7	86.1	33.9	79.2
PACL (HAFN)	ResNet-101	**98.2**	80.3	87.5	86.8	**97.6**	86.0	**94.0**	73.8	**96.3**	**91.6**	94.1	**37.6**	**85.3**
PACL (SAFN)	ResNet-101	97.6	**83.7**	86.1	**91.7**	97.5	**91.3**	92.5	59.1	94.9	90.9	**95.9**	32.5	84.5
SSDA (1-shot)														
ST	ResNet-34	82.6	52.8	75.0	57.6	72.7	39.7	80.5	53.3	59.0	64.1	77.5	12.6	60.6
MME	ResNet-34	86.6	60.1	80.8	61.9	84.0	69.6	87.0	72.4	73.0	50.9	79.4	**14.7**	68.4
PACL (MME)	ResNet-34	**94.9**	**81.5**	**88.9**	**81.3**	**95.9**	**92.4**	**92.2**	**83.3**	**95.2**	**77.4**	**88.4**	2.3	**81.1**
SSDA (3-shot)														
ST	ResNet-34	74.0	71.7	71.2	64.7	78.5	71.8	69.6	51.4	73.7	49.4	80.8	19.8	64.7
MME	ResNet-34	87.2	67.3	74.9	64.5	86.9	85.5	78.8	75.8	84.4	48.0	80.8	**19.9**	71.2
PACL (MME)	ResNet-34	**95.9**	**82.9**	**88.6**	**84.9**	**95.9**	**92.1**	**93.3**	**83.7**	**95.4**	**79.3**	**88.0**	19.5	**83.3**

Table 7: Flexibility analysis of PACL on the *VisDA2017* dataset. "ST" refers the baseline method which trains a model using labeled samples from both source and target domains; no alignment technique is applied. "HAFN + ST" and "SAFN + ST" denote the naive extensions of the methods HAFN and SAFN from UDA to SSDA, by further including labeled target data for the cross-entropy loss optimization. "PACL (MME)", "PACL (HAFN)", and "PACL (SAFN)" are the methods corresponding to incorporating the UDA methods into our framework.