MULTI-PROMPT ALIGNMENT FOR MULTI-SOURCE UNSUPERVISED DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

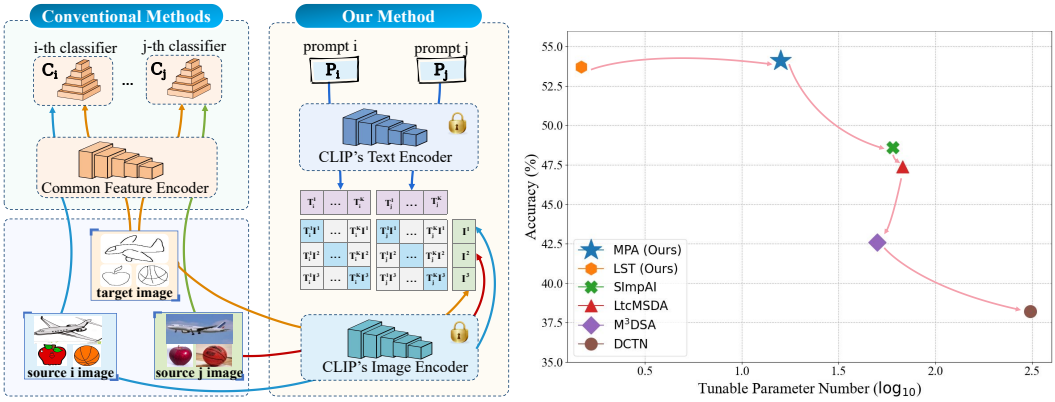
Most existing methods for multi-source unsupervised domain adaptation (UDA) rely on a common feature encoder to extract domain-invariant features. However, learning such an encoder involves updating the parameters of the entire network, which makes the optimization computationally expensive, particularly when coupled with min-max objectives. Inspired by recent advances in prompt learning that adapts high-capacity deep models for downstream tasks in a computationally economic way, we introduce **M**ulti-**P**rompt **A**lignment (MPA), a simple yet efficient two-stage framework for multi-source UDA. Given a source and target domain pair, MPA first trains an individual prompt to minimize the domain gap through a contrastive loss, while tuning only a small set of parameters. Then, MPA derives a low-dimensional latent space through an auto-encoding process that maximizes the agreement of multiple learned prompts. The resulting embedding further facilitates generalization to unseen domains. Extensive experiments show that our method achieves state-of-the-art results on popular benchmark datasets while requiring substantially fewer tunable parameters. To the best of our knowledge, we are the first to apply prompt learning to the multi-source UDA problem and our method achieves the highest reported average accuracy of 54.1% on DomainNet, the most challenging UDA dataset to date, with only 15.9M parameters trained. More importantly, we demonstrate that the learned embedding space can be easily adapted to novel unseen domains.

1 INTRODUCTION

Deep learning has achieved remarkable progress in various computer vision tasks such as image classification (Krizhevsky et al., 2012; He et al., 2016), object detection (Ren et al., 2015; Redmon et al., 2016; Liu et al., 2016) and image segmentation (Long et al., 2015a; Chen et al., 2017). However, these success relies on high capacity models trained in a supervised manner using a massive amount of manually labeled data, which are oftentimes expensive and time-consuming to collect. Furthermore, current deep models are brittle to the presence of domain shift (Quinonero-Candela et al., 2008; Torralba & Efros, 2011; Zhang et al., 2013) in the forms of different image styles, varied lighting conditions, diverse viewpoints, *etc*, between training and testing distributions.

Unsupervised domain adaptation (UDA) is a popular strategy that mitigates domain discrepancies through transferring knowledge learned from a well-labeled source domain to an unlabeled target domain (Pan & Yang, 2010; Csurka, 2017; Wang & Deng, 2018). While significant advances have been achieved, current approaches focus on a single source setting, where all the labeled training data share the same distribution. In practice, however, it is more common for the labeled data to be collected from multiple sources that are diverse in distribution. Naturally, one could still tackle this problem by straightforwardly combining all the data into one single source and apply off-the-shelf UDA methods. However, directly applying single source UDA methods often results in a limited performance, as domain shift also exists among different source domains.

The integration of multiple source domains for improved adaptation results on the unlabeled target domain is generally known as multi-source unsupervised domain adaptation. Inspired by the theoretical analysis of Ben-David et al. (2006), learning domain-invariant feature representations has become a prevailing paradigm for multi-source UDA. One typical approach is to jointly learn a common feature extractor together with domain-specific feature extractors and classifier heads.



(a) Method comparison (b) Tunable parameters vs accuracy on DomainNet

Figure 1: (a) Most conventional multi-source UDA methods use a common feature extractor with domain-specific classifier heads, while we introduce prompt learning to multi-source UDA and omit the repeated need for classifier heads. (b) MPA outperforms all other multi-source UDA methods by a large margin on the DomainNet dataset with roughly one third tunable parameters. We also introduce a **L**atent **S**pace **T**uning (LST) strategy that further reduces the trainable parameters from 15.9M (MPA) to 1.47M while still capable of achieving a high accuracy. See texts for more details.

Various feature distance metrics (Long et al., 2015b; Sun & Saenko, 2016; Kang et al., 2019) or domain adversarial training (Tzeng et al., 2017) can be leveraged that serves as a preliminary alignment between source and target domains. Then different auxiliary losses are carefully designed to further reduce the domain shift. While these methods offer decent results, they require optimizing the entire set of parameters in the network and hence poses a significant challenge for optimization even without the widely used min-max objective (Zhao et al., 2018; Li et al., 2018; Hoffman et al., 2018). Such problem is further amplified if we wish to apply more advanced backbones such as Vision Transformer (Dosovitskiy et al., 2021) for improved performance.

In this paper, we introduce a simple yet efficient approach for multi-source UDA without the need of retraining the entire network and the use of complicated min-max loss functions (See Figure 1 for a comparison). In particular, we build upon prompt learning (Lester et al., 2021; Wei et al., 2021) that has been designed to transfer knowledge learned from large pre-trained vision language models like CLIP (Radford et al., 2021). In prompt learning, image representations are learned contrastively with a piece of language text termed as "prompt" that describes the class of the image. While recent studies (Ge et al., 2022; Ben-David et al., 2022) suggest learnable prompts can be used for the task of UDA, they are restricted to the single source scenario and directly generalizing them to the multi-source setting produces limited results. Furthermore, while prompt learning provides an efficient alternative, we argue that the number of parameters needed to be tuned could be further reduced.

In light of this, we present a surprisingly easy framework, **M**ulti-**P**rompt **A**lignment (MPA), for multi-source UDA. MPA is composed of two stages, one to learn an individual prompt by tuning a small set of parameters for each source and target domain pair and one to mine the relationships among learned prompts through deriving a shared embedding space. The resulting embedding is expected to be domain-invariant and can generalize to unseen domains. More specifically, for the first stage, given a source domain and a target domain, we use CLIP as our backbone and learn one prompt tailored for such a pair. We then align all the learned prompts in a latent space of a lower dimension d_I. This is accomplished by a simple auto-encoder network with a reconstruction loss. Additionally, we incorporate an $\mathcal{L}_1$ loss so that the reconstructed prompts agree on the classification of target images. This is beneficial for prompts to handle situations in which the target data lies near the decision boundary. We conduct extensive experiments on multiple benchmarks and the results show that our method outperforms state-of-the-art methods clearly in the multi-source setting. In particular, on DomainNet (Peng et al., 2019), the most challenging dataset for multi-source UDA so far, MPA surpasses all state-of-the-arts methods. More importantly, as the latent space is optimized with prompts from multiple source domains, it encodes knowledge shared by different domains and could potentially generalize to unseen domains by traversing the space. Consequently, we show how surprisingly easy it is to tune the learned low-dimensional embedding for deployment in unseen domains. In summary, our contributions are three-fold:

- We introduce **M**ulti-**P**rompt **A**lignment (MPA) for multi-source UDA. MPA takes advantage of prompt learning, thus greatly reducing the number of parameters needed for training compared with alternative methods.
- MPA learns a latent embedding space by maximizing the consensus of multiple learned prompts and the resulting low-dimensional embedding facilitates surprisingly easy adaptation to novel unseen domains.
- MPA achieves state-of-the-art results on several popular benchmarks. Specifically, on the large-scale DomainNet dataset, MPA achieves the best reported average accuracy. Moreover, tuning the learned latent subspace offers comparable results with even fewer parameters when generalizing to unseen domains.

2 Related Work

2.1 Multi-Source Unsupervised Domain Adaptation

First studied by Yang et al. (2007), multi-source UDA has drawn increasing attention in the community. Throughout the years, various methods have been studied. For example, in MDAN (Zhao et al., 2018) and DCTN (Xu et al., 2018), a discriminator applied with adversarial losses is trained so that the features from source and target domains are aligned. MFSAN (Zhu et al., 2019) calculates and aligns the maximum mean discrepancy for each source and target pair. Similarly, M^3SDA (Peng et al., 2019) aligns the moment distance for both target and source domains. All these methods require a shared feature extractor to obtain domain-invariant features, which is inevitably difficult to optimize and at the risk of losing semantic information as the number of domains increases. On the contrary, Rakshit et al. (2019) adopt one domain-specific encoder for each source and target pair while Zhao et al. (2020) pretrain a classifier for each source domain and then adversarially map the target images into each trained feature space. The better alignment of these methods is at the cost of significantly increased number of parameters needed for training. To overcome such a trade-off between performance and efficiency, we introduce prompt learning to multi-source UDA. Since each prompt contains substantially fewer parameters compared to a feature extractor, learning distinct prompts for each source and target pair is affordable.

2.2 Prompt Learning

Traditionally, given a pre-trained language model, a common approach in deep learning is fine-tuning the whole model or its task-specific heads to adjust to downstream tasks. While effective, however, two main drawbacks exist. First of all, as the model size keeps increasing, pre-training and fine-tuning has become more and more expensive. Secondly, for each diverse task, fine-tuning needs to be repeatedly conducted. Recently, researchers have shown that learned large-scale language models can handle a wide range of downstream tasks with only a few or even no samples by prepending instructions to the input text (Liu et al., 2021b). Such instruction texts are called prompts. Consequently, prompts can be tuned instead of the entire network for a more efficient adaptation to downstream tasks. Originally, prompts are essentially sequences of manually designed language tokens that are mapped to an embedding space. To date, extensive research has demonstrated that training soft prompts, *i.e.*, prompts with their own parameters learned by deep models, is more effective (Li & Liang, 2021; Lester et al., 2021). The success of prompt learning in NLP has also garnered attention in the vision community that motivated the establishment of many related work. To name a few, Zhou et al. (2022) are the first to apply soft prompt learning to the image recognition task. Ju et al. (2021) explore prompt learning for efficient and lightweight video understanding. While prompting in these studies are limited to the input of text encoders, Jia et al. (2022) prepend token embeddings directly to the image patches.

3 Method

Our goal is to use multiple labeled source domains for improved performance on a target domain while only tuning a small set of parameters. To this end, we leverage prompt learning, which is an effective strategy by learning a small set of parameters to adapt a pretrained large-scale model to different downstream tasks. In the following, we first review prompt learning in CLIP in Section 3.1 and

then elaborate in Section 3.2 our proposed MPA method, which is a two-stage framework that seeks agreement from multiple learned prompts and derives a joint latent space with domain-invariant knowledge.

3.1 An Overview of Prompt Learning in CLIP

CLIP consists of an image encoder and a text encoder, which are jointly trained with a contrastive loss on 400M image and text pairs. The image encoder f, which can either be a ResNet (He et al., 2016) or a Vision Transformer (Dosovitskiy et al., 2021), maps raw images to an embedding space, while the text encoder g is a Transformer (Vaswani et al., 2017) that maps an input text sequence to the same embedding space. A prompt in CLIP usually exists in the form of "a photo of [CLS]" where [CLS] is a class token that can be replaced by a certain class name. This sequence of tokens is first converted into a lower cased byte pair encoding (BPE) representation, which is essentially a unique numeric ID (Zhou et al., 2022). Then the numeric IDs are embedded to a 512 dimension vector that is further passed to the Transformer text encoder. In our work, instead of using manually crafted prompts, we train soft prompts that are directly embedded by the text encoder. Given an image $\boldsymbol{x}$ and a text embedding $\boldsymbol{w}_k$ for class $k \in \{1, 2, ..., K\}$, where K is the total number of categories, CLIP aligns them in a contrastive manner so that:

$$p(y = k|\boldsymbol{x}) = \frac{\exp(< \boldsymbol{w}_k, f(\boldsymbol{x}) > /T)}{\sum_{i=1}^{K} \exp(< \boldsymbol{w}_i, f(\boldsymbol{x}) > /T)} \tag{1}$$

is maximized when the input image $\boldsymbol{x}$ indeed belongs to class k. Here $< \cdot, \cdot >$ denotes the cosine similarity and T is a learnable temperature parameter.

3.2 Multi-Prompt Alignment

Let N denote the total number of domains, where the first $N-1$ domains are source domains and the N-th domain is the target domain. For all $N-1$ source domains, both images and their labels are provided, while for the target domain, we only assume access to their images, as in the standard UDA setting. For multi-source UDA, we wish to learn a domain-invariant latent space so that the domain shift among different source domains as well as the discrepancies between all the source and target domain pairs can be minimized. Unlike current multi-source UDA methods that require a significant amount of parameters for adaption, we leverage prompt learning for alignment among different domains. In the following, we begin by introducing the prompt design, followed by multi-prompt alignment. Then we describe how to learn a joint embedding space that has the potential of generalizing to unseen domains.

Prompt Design. Following (Ge et al., 2022), our prompt for multi-source UDA includes a set of class-specific context vectors $\boldsymbol{v}_i^k, i \in \{1, 2, ..., M_1\}, k \in \{1, 2, ..., K\}$ and another set of domain-specific vectors shared across all classes $\boldsymbol{d}_j^d, j \in \{1, 2, ..., M_2\}, d \in \{s, t\}$. See Figure 2 for an overview. Here, M_1 and M_2 represent the number of tokens, K is the number of classes and s is short for source and t is short for target, resulting in a total of $2K$ categories for training. In other words, each class prompt $\boldsymbol{t}_k^d \in \mathbb{R}^{1 \times (M_1 + M_2) \times 512}$ is a concatenation of a "source prompt" and a "target prompt". Therefore, the prompt for each source and target pair can be derived as:

$$\boldsymbol{P}_i = [\boldsymbol{t}_1^s, \boldsymbol{t}_2^s, ..., \boldsymbol{t}_K^s, \boldsymbol{t}_1^t, \boldsymbol{t}_2^t, ..., \boldsymbol{t}_K^t]^\top, i \in \{1, 2, ..., N-1\}. \tag{2}$$

These prompts serve as learnable parameters that help bridging the domain gap between a source domain and the target domain through a contrastive loss, as will be introduced below.

Learning Individual Prompts. To apply prompt learning to multi-source UDA, we first train individual prompts for each source and target pair using the image and text encoders of CLIP. Given an image $\boldsymbol{x}^s$ sampled from the source domain $\mathcal{D}_s$ whose label is y^*, we optimize the prompts so that the outputs from the image and text encoder are aligned. For an image $\boldsymbol{x}^t$ from the target domain $\mathcal{D}_t$ whose label is unknown, we first leverage the strong zero-shot ability of CLIP to generate static pseudo-label $\hat{y}^*$ for image-text alignment. We only generate pseudo-labels for images whose maximum probability is larger than a fixed threshold τ using Equation 1. While one could use more sophisticated approaches like self-training to generate pseudo labels (Zou et al., 2018; Liu et al.,

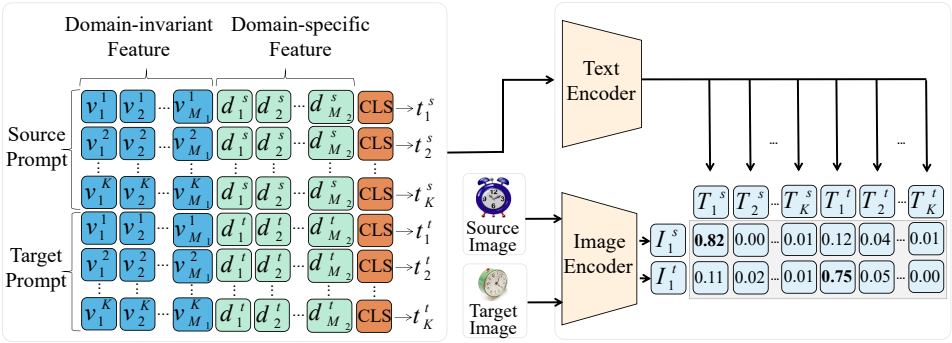


Figure 2: Each source and target pair prompt $\boldsymbol{P}_i$ is the concatenation of a "source prompt" segment and a "target prompt" segment composed of domain-invariant and domain-specific features. Therefore, the size of $\boldsymbol{P}_i$ is $\mathbb{R}^{2K\times(M_1+M_2)\times 512}$. During our prompt training stage, the text encoder and the image encoder of CLIP are both frozen.

2021a), we find that pseudo labels from CLIP are simple and effective. We then train prompts with cross-entropy loss functions. Figure 2 gives an overview of the process. More formally, for a prompt $\boldsymbol{P}_i, i \in \{1, 2, ..., N-1\}$, the objective function for optimization follows:

$$\min_{\boldsymbol{P}_i} -\frac{1}{n_s}\sum_{\boldsymbol{x}^s\sim\mathcal{D}_s} \log P(y=y^*|\boldsymbol{x}^s;\boldsymbol{P}_i) - \frac{1}{n_t}\sum_{\boldsymbol{x}^t\sim\mathcal{D}_t} \log P(y=\hat{y^*}|\boldsymbol{x}^t;\boldsymbol{P}_i). \tag{3}$$

Here, the probability $P(\cdot|\boldsymbol{x}^d;\boldsymbol{P}_i)$ of an image sample belonging to the k-th class is derived from a contrastive loss:

$$P(y=k|\boldsymbol{x}^d;\boldsymbol{P}_i) = \frac{\exp(< g(\boldsymbol{t}_k^d), f(\boldsymbol{x}^d) > /T)}{\sum_{d\in\{s,t\}}\sum_{i=1}^{K}\exp(< g(\boldsymbol{t}_i^d), f(\boldsymbol{x}^d) > /T)}, \tag{4}$$

where $d \in \{s, t\}$ is a domain identifier indicating where the image comes from, T is a learnable temperature parameter, and f and g represents the image and text encoder in CLIP respectively, which are kept frozen during training. This specific design can push the prompts to learn disentangled representation of both class-invariant and class-specific semantic information to boost the performance of domain adaptation methods (Bousmalis et al., 2016; Liu et al., 2018).

Once prompts are learned, the predicted label of an image $\boldsymbol{x}^t$ can be computed as:

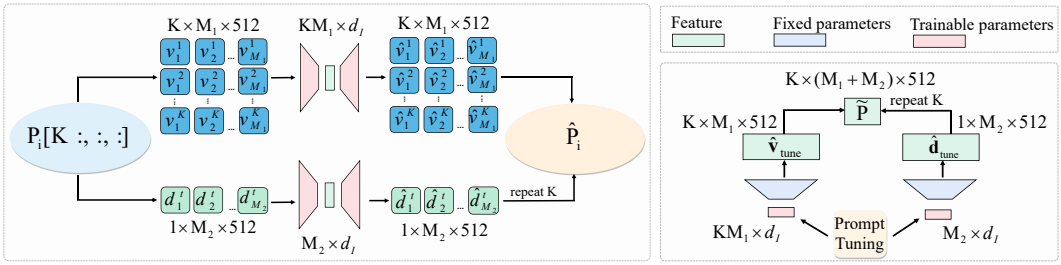
$$\arg\max_k \frac{\exp(< g(\boldsymbol{t}_k^t), f(\boldsymbol{x}^t) > /T)}{\sum_{i=1}^{K}\exp(< g(\boldsymbol{t}_i^t), f(\boldsymbol{x}^t) > /T)}. \tag{5}$$

Multi-Prompt Alignment. Thus far, we have obtained a prompt for each source and target domain pair. However, the number of images as well as the noise level in each source domain varies, and hence these learned prompts might produce inconsistent results even for the same image. In the second stage, we aim to align predictions from different prompts and more importantly, we wish to find a domain-invariant latent space that minimizes the noise in learned prompts as well as can potentially generalize to unseen domains. To this end, we leverage auto-encoders that are trained to reconstruct the learned prompts. More formally, we use two separate auto-encoders, each consisting of a projection function $\textbf{Proj}(\cdot)$ and a back-projection function $\textbf{Proj}_b(\cdot)$. The learned prompts $\boldsymbol{P}_i$ are first projected into a latent subspace of a lower dimension d_I by $\textbf{Proj}(\cdot)$, followed by $\textbf{Proj}_b(\cdot)$ projecting the vectors back into soft prompts $\hat{\boldsymbol{P}}_i$. Instead of projecting the whole prompt, we adjust $\boldsymbol{P}_i$ to only contain the target token segment and feed its domain-specific and domain-invariant pieces to the two auto-encoders respectively. We posit that since these two feature vectors serve different purposes, using two separate auto-encoders will help with the alignment process. The $\textbf{Proj}(\cdot)$ function is implemented by a one-layer feed forward network while $\textbf{Proj}_b(\cdot)$ is implemented by a two-layer nonlinear perceptron:

$$\textbf{Proj}(\boldsymbol{P}_i) = \boldsymbol{W}_1(\boldsymbol{P}_i) + \boldsymbol{b}_1 \quad \text{and} \quad \textbf{Proj}_b(\boldsymbol{d}_I) = \boldsymbol{W}_3(\tanh(\boldsymbol{W}_2\boldsymbol{d}_I + \boldsymbol{b}_1) + \boldsymbol{b}_2 \tag{6}$$

and we optimize a reconstruction loss

$$\mathcal{L}_{AE} = \frac{1}{N-1}\sum_{i=1}^{N-1}\|\hat{\boldsymbol{P}}_i - \boldsymbol{P}_i\|_2^2 \quad \text{where} \quad \hat{\boldsymbol{P}}_i = \textbf{Proj}_b(\textbf{Proj}(\boldsymbol{P}_i)). \tag{7}$$



(a) Reconstruction of prompts (b) Generalize to novel domain

Figure 3: (a) All prompts are first sliced to the size $\mathbb{R}^{K\times(M_1+M_2)\times 512}$ containing only the target token segment. Then they are further partitioned into domain-invariant and domain-specific pieces that are projected into the same latent space and aligned by an auto-encoder structure. (b) The latent subspace learned by the auto-encoder can be utilized for generalizing to novel domain.

Intuitively, one would expect a certain target domain image to be classified as the same category for all the reconstructed prompts $\hat{P}_i$. Inspired by this, we align the reconstructed prompts by introducing an additional $\mathcal{L}_1$ loss to the objective function

$$\mathcal{L}_1 = \frac{2}{(N-1)\times(N-2)} \sum_{j=1}^{N-2} \sum_{i=j+1}^{N-1} |P(y=k^t|\boldsymbol{x}_k, \hat{\boldsymbol{P}}_i) - P(y=k^t|\boldsymbol{x}_k, \hat{\boldsymbol{P}}_j)| \tag{8}$$

The overall loss function now becomes:

$$\mathcal{L} = \mathcal{L}_{CLS} + \alpha\mathcal{L}_1 + \mathcal{L}_{AE}, \tag{9}$$

where $\mathcal{L}_{CLS}$ is the cross entropy loss calculated using the reconstructed prompts $\hat{\boldsymbol{P}}_i$ and static pseudo labels. Here α is a hyper-parameter controlling the weight of the $\mathcal{L}_1$ loss. The whole training procedure is depicted in Figure 3a. Finally, for predicting the labels of target samples, we compute the average of the output logits using each $\hat{\boldsymbol{P}}_i$.

Generalizing to Unseen Domains. Shown in Figure 3b, with multi-prompt alignment, we are able to derive a low-dimensional embedding space that potentially captures the relationships among different domains. With this, we introduce a **L**atent **S**ubspace **T**uning (LST) strategy. Given a novel unseen domain, a domain-invariant feature vector $v_{tune} \in R^{N\times M_1\times d_I}$ with a domain-specific feature vector $d_{tune} \in R^{1\times M_2\times d_I}$ is randomly initialized and passed to the two learned back projection function $\mathbf{Proj}_b(\cdot)$ from MPA. Consequently, an entirely new prompt $\tilde{\boldsymbol{P}}$ can be constructed for predicting image labels on the "unseen" domain. This is achieved by minimizing the following loss function:

$$\min_{d_{tune}, v_{tune}} -\frac{1}{n_{new}} \sum_{\boldsymbol{x}^{new}\sim\mathcal{D}_{new}} \log P(y=\hat{y}^*|\boldsymbol{x}^{new}; d_{tune}; v_{tune}). \tag{10}$$

By doing so, this further decreases the number of tunable parameters by a factor of at least $(N-1)\times\frac{d_c}{d_I}$ times (for example, on the Office-Home dataset, $d_I = 150$, $d_c = 512$ and $N = 4$) as against to MPA when adapting to novel domains and is thus much more computationally efficient.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We conduct experiments on three popular benchmark datasets of UDA to evaluate the effectiveness of MPA, including ImageCLEF, Office-Home and DomainNet. ImageCLEF is a small-scaled dataset consisting of 1,800 images from 12 object categories in 3 different domains: ImageNet ILSVRC 2012(I), Pascal VOC 2012(P), and Caltech-256 (C). Office-Home is a medium scaled dataset consisting of about 15,500 images from 65 categories in 4 different domains: Art, Clipart, Product and Real World. DomainNet is the largest dataset to date, consisting of about 0.6 million images from 345 categories in 6 different domains: clipart, infograph, painting, quickdraw, real and sketch. The top-1 accuracy is used as our evaluation metric.

Implementation Details. For fair comparisons, we adopt a ResNet50 as our backbone on ImageCLEF and Office-Home and a ResNet101 on DomainNet. The weights for ResNet50 and ResNet101 are from CLIP and frozen through our experiments. The prompts and auto-encoders are trained using the mini-batch SGD optimizer with a learning rate of 0.003 and 0.005 while the learned subspace is tuned with 0.0005 learning rate. We use a batch size of 32 and adopt a cosine learning rate scheduler.

For the hyper-parameters of stage one, token length M_1 and M_2 are both set to 16. Pseudo-label threshold τ is set to 0.4 for producing reliable labels. As for stage two, α in Equation 9 is set to 500. The weight matrix of the back projection function $\boldsymbol{W}_2$ has a size of $\mathbb{R}^{384\times d_I}$, where d_I is 100 for ImageCLEF, 150 for OfficeHome and 250 for DomainNet. Therefore, for generalizing to novel domains, only 0.02M, 0.17M and 1.47M parameters are tuned respectively. We report results of the following settings: (1) CLIP: zero-shot CLIP on the target domain, which can be regarded as the baseline of our method. (2) Source Combined: all source domains are combined into one single domain and then popular single-source UDA methods are applied. (3) Multi-Source: results reported from other multi-source UDA methods. Note that we tried to re-implement state-of-the-art methods using backbone networks from CLIP, yet the results are unsatisfactory. Similar trends are also found in (Devillers et al., 2021; Yang et al., 2022) when transferring CLIP to pure vision tasks.

4.2 Comparison to State-of-the-Art

We compare MPA with state-of-the-art methods on three datasets, including DAN (Long et al., 2015b), D-CORAL (Sun & Saenko, 2016), DANN (Ganin et al., 2016), MCD (Saito et al., 2018), DCTN (Xu et al., 2018), MFSAN (Zhu et al., 2019), M^{3}SDA (Peng et al., 2019), MADAN (Zhao et al., 2020), SImpAI (Venkat et al., 2020), LtC-MSDA (Wang et al., 2020), T-SVDNet (Li et al., 2021), PFSA (Fu et al., 2021), DAPL (Ge et al., 2022) and PTMDA (Ren et al., 2022).

	ImageCLEF				**Office-Home**				
	→ C	**→ I**	**→ P**	**Avg**	**→ Ar**	**→ Cl**	**→ Pr**	**→ Rw**	**Avg**
Zero-Shot									
CLIP	95.1	87.3	74.0	85.5	71.5	50.2	81.3	82.4	71.4
Source Combine									
DAN	93.3	92.2	77.6	87.7	68.5	59.4	79.0	82.5	72.4
DANN	93.7	91.8	77.9	87.8	68.4	59.1	79.5	82.7	72.4
D-CORAL	93.6	91.7	77.1	87.5	68.1	58.6	79.5	82.7	72.2
DAPL	95.3	87.0	75.7	86.0	72.8	51.9	82.6	83.7	72.8
Multi-Source									
DCTN	95.7	90.3	75.0	87.0	*N.A.*	*N.A.*	*N.A.*	*N.A.*	*N.A.*
MDDA	*N.A.*	*N.A.*	*N.A.*	*N.A.*	66.7	**62.3**	79.5	79.6	71.0
SImpAI$_{50}$	93.3	91.0	77.5	87.3	70.8	56.3	80.2	81.5	72.2
MFSAN	95.4	93.6	79.1	89.4	72.1	62.0	80.3	81.8	74.1
MPA (ours)	**98.6**	**96.2**	**80.4**	**91.7**	**74.8**	54.9	**86.2**	**85.7**	**75.4**

Table 1: Accuracy (%) on ImageCLEF and Office-Home dataset.

The results on ImageCLEF and Office-Home are shown in Table 1. For ImageCLEF, it is obvious that MPA outperforms other methods on every task with an average accuracy of 91.7%, where there is a 3% increase when adapting to domain C and I. For Office-Home, MPA achieves the best results except when adapting to the domain Clipart. Nevertheless, we achieve an accuracy of 75.4% on average, which is 1.3% higher than the second best method MFSAN. It is worth noting that compared to state-of-the-art method MFSAN on both datasets, MPA only trains 0.78M and 2.36M parameters, while MFSAN has a total of 51.75M and 51.80M parameters for optimizing (66.3 and 21.9 times larger than ours).

Table 2 shows that for DomainNet, MPA improves other multi-source UDA methods by more than 5%. To the best of our knowledge, this is the highest reported accuracy on this dataset so far with less than one third parameters optimized compared with most state-of-the-arts methods. Regarding individual adaptations, MPA achieves best results on most of the adaptation tasks but performs mediocre on the quickdraw domain. Interestingly, the result is even a little worse than CLIP. We hypothesize that this is because of the large domain gap between quickdraw and other domains. While CLIP exhibits strong performance on the DomainNet dataset, its performance on ImageCLEF and

	DomainNet						
	→ **Clp**	→ **Inf**	→ **Pnt**	→ **Qdr**	→ **Rel**	→ **Skt**	**Avg**
Zero-Shot							
CLIP	61.3	42.0	56.1	10.3	79.3	54.1	50.5
Source Combined							
DANN	45.5	13.1	37.0	13.2	48.9	31.8	32.6
MCD	54.3	22.1	45.7	7.6	58.4	43.5	38.5
DAPL	62.4	43.8	59.3	10.6	81.5	54.6	52.0
Multi-Source							
DCTN	48.6	23.5	48.4	7.2	53.5	47.3	38.2
SImpAI$_{101}$	**66.4**	26.5	56.6	18.9	68.0	55.5	48.6
M^{3}SDA-β	58.6	26.0	52.3	6.3	62.7	49.5	42.6
LtC-MSDA	63.1	28.7	56.1	16.3	66.1	53.8	47.4
T-SVDNet	66.1	25.0	54.3	16.5	65.4	54.6	47.0
PFSA	64.5	29.2	57.6	**17.2**	67.2	55.1	48.5
PTMDA	66.0	28.5	58.4	13.0	63.0	54.1	47.2
MPA (ours)	65.2	**47.3**	**62.0**	10.2	**82.0**	**57.9**	**54.1**

Table 2: Accuracy (%) on the DomainNet dataset.

OfficeHome dataset is limited. On the contrary, MPA achieves consistent decent results regardless of the dataset assessed on. Specifically, for all three datasets, MPA surpasses CLIP by 6.2%, 4.0% and 3.6% respectively, demonstrating the success of our strategy on reducing domain discrepancies.

Effectiveness of Generalizing to Unseen Domains Inspired by recent research that shows the generalization ability of large language models when adapting to unseen tasks presented with natural language prompts (Sanh et al., 2022), we are interested in whether MPA also possess such ability. Results from Table 1 and Table 2 are already strong indication of the success of MPA under conventional multi-source UDA setting. Now, we would like to investigate a more efficient framework, the LST strategy as mentioned in Section 3.2, for fast adaptation tasks.

We first present a concrete example. On the Office-Home dataset, LST can be conducted by training MPA using Clipart and Product as source domains and Art as target domain, while fine-tuning the latent subspace on the unseen Real World domain. Applying this to all domains from the Office-Home and DomainNet dataset produces results shown in Table 3, where compared with MPA, the performance only dropped by 0.4% on DomainNet and 1.6% on Office-Home. Nevertheless, LST achieves higher accuracy compared to most baseline methods of Table 1 and Table 2. In particular, LST is still better than CLIP with an increase of 2.4% and 3.2% respectively for the two tested datasets. Notably, only a total of 0.17M and 1.47M parameters are tuned, which is a further drop in the number of parameters (0.17M *v.s.* 2.36M, 1.47M *v.s.* 15.9M in MPA). All of these are convincing evidence showing the strong generalization ability of MPA.

	Office-Home					DomainNet						
	→ **Ar**	→ **Cl**	→ **Pr**	→ **Rw**	**Avg**	→ **Clp**	→ **Inf**	→ **Pnt**	→ **Qdr**	→ **Rel**	→ **Skt**	**Avg**
CLIP	71.5	50.5	81.3	82.4	71.4	61.3	42.0	56.1	10.3	79.3	54.1	50.5
MPA	74.8	54.9	86.2	85.7	75.4	65.2	47.3	62.0	10.2	82.0	57.9	54.1
LST	72.9	52.2	84.9	85.0	73.8	64.6	46.7	61.6	9.8	81.2	57.6	53.6

Table 3: Results (%) of different approaches when generalizing to unseen domains on Office-Home.

Qualitative Results To obtain further insights about MPA, we visualize, in Figure 4, the distributions of prediction confidence of MPA and CLIP for four target domains on the Office-Home dataset. We see that for Art, Prouct and Real World, MPA produces highly confident predictions for the majority of samples. For the Clipart domain where the recognition accuracy is lower than the other domains, MPA is clearly better than CLIP that produces a distribution skewed to the bottom (low confidence).

4.3 Ablation Study

In this section, we conduct extensive experiments to give a more detailed analysis of our method. Otherwise stated, all ablation studies are done on the Office-Home dataset for effiency purposes.

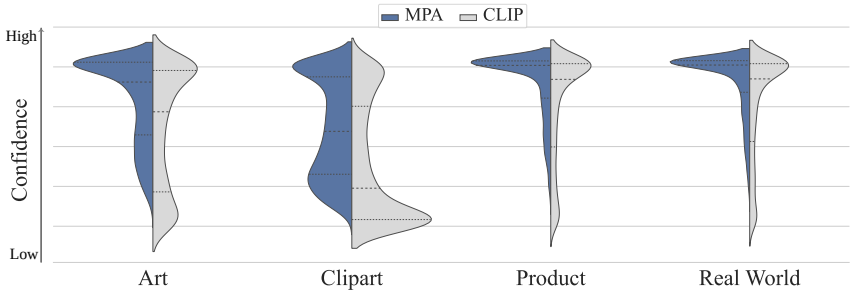


Figure 4: Visualization of the confidence density.

Effectiveness of Prompt Alignment The second stage of our method can be seen as a refinement of the prompt learned during stage one. Thus, we verify how the quality of those individual prompts has changed and the results are shown in Table 4. It is worth pointing out that stage one of MPA improves the accuracy of CLIP by an average of 1.2%, while the second stage improves the first stage by an average of 2.8%, highlighting that the prompt has been refined with better domain-invariant information.

	A → C	**A → P**	**A → R**	**C→ A**	**C→ P**	**C→ R**	**P→ A**	**P→ C**	**P→ R**	**R→ A**	**R→ C**	**R→ P**	**Avg**
CLIP	50.2	81.3	82.4	71.5	81.3	82.4	71.5	50.2	82.4	71.5	50.2	81.3	71.4
Stage One	52.2	83.5	82.1	72.8	83.6	82.8	73.3	52.7	82.4	72.0	51.6	82.1	72.6
Stage Two	**54.1**	**85.9**	**85.2**	**74.3**	**86.0**	**85.3**	**74.6**	**54.1**	**85.3**	**74.3**	**54.2**	**86.0**	**75.4**

Table 4: Performance (%) of individual prompts on Office-Home dataset.

Effectiveness of loss function To validate the effectiveness of our objective function Equation 9, two variants are evaluated: one without the $\mathcal{L}_{AE}$ loss and one without the $\mathcal{L}_1$ loss. Table 5a shows that the removal of either piece will result in a performance degradation of about 0.7%. In particular, $\mathcal{L}_1$ loss appears to have a larger impact on the overall performance. On the Clipart domain, the accuracy dropped by a large margin of 1.5% without the $\mathcal{L}_1$ loss, demonstrating the effectiveness of enforcing consistent predictions on different prompts.

$\mathcal{L}_1$	$\mathcal{L}_{AE}$	**→ Ar**	**→ Cl**	**→ Pr**	**→ Rw**	**Avg**
×	✓	**74.8**	53.4	85.2	85.3	74.7
✓	×	74.0	53.8	85.5	85.4	74.7
✓	✓	**74.8**	**54.9**	**86.2**	**85.7**	**75.4**

(a) Effectiveness of loss function.

# of AE	**→ Ar**	**→ Cl**	**→ Pr**	**→ Rw**	**Avg**
Zero	**75.0**	53.4	85.7	85.4	74.9
One	74.6	54.3	85.9	85.6	75.1
Two	74.8	**54.9**	**86.2**	**85.7**	**75.4**

(b) Effectiveness of auto-encoders.

Table 5: Ablation studies.

Effectiveness of auto-encoders In addition to the objective function, we also tested the effectiveness of utilizing the auto-encoder structure in MPA. For details, zero auto-encoders means we completely discarded the auto-encoder structure and one auto-encoder is for testing the necessity of using two separate auto-encoders for reconstructing domain-specific and domain-invariant prompt. Results from Table 5b show that in general, our design of the auto-encoder structure is beneficial to the overall performance.

5 Conclusion

In this paper, we introduced prompt learning to multi-source UDA and proposed a simple MPA scheme to align the source and target domains. MPA is composed of two stages. The first stage is to train individual source and target pair prompts and the second stage is to align them by auto-encoder structures. Extensive experiments showed that MPA achieved better results on various multi-source UDA tasks with substantially fewer parameters tuned. Moreover, a latent subspace tuning strategy was introduced for generalization to novel unseen domains that further dropped the number of parameters needed for training. Hopefully, our work can inspire future work on applying prompt learning to transfer learning.

REFERENCES

Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *CVPR*, 2021.

Eyal Ben-David, Nadav Oved, and Roi Reichart. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. *TACL*, 2022.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NeurIPS*, 2006.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NeurIPS*, 2016.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017.

Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

Benjamin Devillers, Bhavin Choksi, Romain Bielawski, and Rufin VanRullen. Does language help generalization in vision models? *arXiv preprint arXiv:2104.08313*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Yangye Fu, Ming Zhang, Xing Xu, Zuo Cao, Chao Ma, Yanli Ji, Kai Zuo, and Huimin Lu. Partial feature selection and alignment for multi-source domain adaptation. In *CVPR*, 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *arXiv preprint arXiv:2202.06687*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *NeurIPS*, 2018.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022.

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. *arXiv preprint arXiv:2112.04478*, 2021.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Ruihuang Li, Xu Jia, Jianzhong He, Shuaijun Chen, and Qinghua Hu. T-svdnet: Exploring high-order prototypical correlations for multi-source domain adaptation. In *ICCV*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Yitong Li, David E Carlson, et al. Extracting relationships by multi-domain matching. In *NeurIPS*, 2018.

Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. 2021a.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021b.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *CVPR*, 2018.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015a.

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

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 2010.

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

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. 2008.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Sayan Rakshit, Biplab Banerjee, Gemma Roig, and Subhasis Chaudhuri. Unsupervised multi-source domain adaptation driven by deep adversarial ensemble learning. In *GCPR*, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

Chuan-Xian Ren, Yong-Hui Liu, Xi-Wen Zhang, and Ke-Kun Huang. Multi-source unsupervised domain adaptation via pseudo target domain. *TIP*, 2022.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *ICLR*, 2022.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Naveen Venkat, Jogendra Nath Kundu, Durgesh Singh, Ambareesh Revanur, et al. Your classifier can secretly suffice multi-source domain adaptation. In *NeurIPS*, 2020.

Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *ECCV*, 2020.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018.

Colin Wei, Sang Michael Xie, and Tengyu Ma. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. In *NeurIPS*, 2021.

Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *CVPR*, 2018.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *CVPR*, 2022.

Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM MM*, 2007.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013.

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *NeurIPS*, 2018.

Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *AAAI*, 2020.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022.

Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *AAAI*, 2019.

Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

A APPENDIX

A.1 MPA WITH LIMITED DATA

Since prompt learning is also well-known for its strong few-shot ability, we design a limited data setting where only a small portion of the data is used for training MPA while still tested on the full dataset to assess whether our method has inherited such ability. We think that this framework is also practically meaningful as access to data might be limited in real-life scenario due to privacy, security and storage concerns (Ahmed et al., 2021). Specifically, no more than 50 images of each category are sampled using two distinct sampling strategies. The most direct one is by complete random sampling. On the contrary, strategy two selects high quality images by utilizing CLIP's zero shot inference ability similar to our pseudo label generating process.

Such setting is only experimented on the DomainNet dataset due to its large data scale. Despite the fact that the total number of data was reduced from 0.6M to roughly 0.1M, results from Table 6 show that MPA is still capable of achieving a surprisingly good result where the recognition accuracy only dropped by an average of 0.7% compared with full data training and is still 2.9% better than our baseline CLIP. We also want to point out that the quality of input images seems to be irrelevant to the performance, as results from the two sampling strategies don't vary much.

	$\rightarrow$ **Clp**	$\rightarrow$ **Inf**	$\rightarrow$ **Pnt**	$\rightarrow$ **Qdr**	$\rightarrow$ **Rel**	$\rightarrow$ **Skt**	**Avg**
CLIP	61.3	42.0	56.1	10.3	79.3	54.1	50.5
Random Sampled Data	64.1	46.9	60.5	9.6	81.6	57.3	53.3
CLIP Sampled Data	64.2	47.3	60.9	9.6	81.6	57.6	53.5
All Data	**65.2**	**47.3**	**62.0**	**10.2**	**82.0**	**57.9**	**54.1**

Table 6: MPA with limited data on the DomainNet dataset