Empowering Unsupervised Domain Adaptation with Large-scale Pre-trained Vision-Language Models

Zhengfeng Lai[1]* Haoping Bai[2] Haotian Zhang[2] Xianzhi Du[2]
Jiulong Shan[2] Yinfei Yang[2] Chen-Nee Chuah[1] Meng Cao[2]
[1]University of California, Davis [2]Apple AI/ML
[1]{lzhengfeng, chuah}@ucdavis.edu [2]{haoping_bai, haotian_zhang2, xianzhi, jlshan, yinfei_yang, mengcao}@apple.com

Abstract

Unsupervised Domain Adaptation (UDA) aims to leverage the labeled source domain to solve the tasks on the unlabeled target domain. Traditional UDA methods face the challenge of the tradeoff between domain alignment and semantic class discriminability, especially when a large domain gap exists between the source and target domains. The efforts of applying large-scale pre-training to bridge the domain gaps remain limited. In this work, we propose that Vision-Language Models (VLMs) can empower UDA tasks due to their training pattern with language alignment and their large-scale pre-trained datasets. For example, CLIP and GLIP have shown promising zero-shot generalization in classification and detection tasks. However, directly fine-tuning these VLMs into downstream tasks may be computationally expensive and not scalable if we have multiple domains that need to be adapted. Therefore, in this work, we first study an efficient adaption of VLMs to preserve the original knowledge while maximizing its flexibility for learning new knowledge. Then, we design a domain-aware pseudo-labeling scheme tailored to VLMs for domain disentanglement. We show the superiority of the proposed methods in four UDA-classification and two UDA-detection benchmarks, with a significant improvement (+9.9%) on DomainNet.

1. Introduction

The domain gap between curated datasets from a source domain and real-world applications (target domain) can significantly downgrade the models' performance, including both image classification [21, 52, 60, 73] and object detection [2, 9]. However, curating the dataset by humans for each application domain can be time-consuming and labor-intensive. To relieve the annotation costs, unsupervised domain adaptation (UDA) is proposed to train a model for an

*This work was mainly done at Apple.

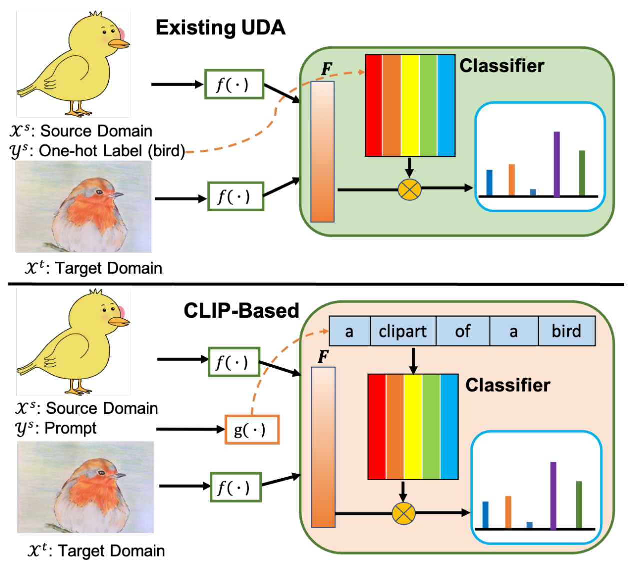

Figure 1. Comparison between existing UDA methods and our proposed CLIP-based method for UDA classification.

unlabeled target domain by leveraging a source domain that is well-annotated to transfer the knowledge across the domain shift [11, 35, 52, 60, 73].

Prior standard UDA methods [3, 15, 32, 35] are built on *ImageNet pre-trained* Convolutional Neural Networks (CNN, e.g., ResNet [20]), which serves as the vision encoder and achieves impressive results on small-sized UDA classification benchmarks, such as Office-31 [46]. To get a better alignment across different domains, recent works [52, 60, 78] use the *pre-trained* Vision Transformers (ViT) [10] as the backbone since the cross-attention layer in ViT can achieve better feature alignment between different domains [60, 78]. However, although *pre-trained* ViT-based methods have shown improvement compared to the ResNet-based methods, the results on large-scale benchmarks, such as DomainNet [42], are still limited. As shown in Table 1, the most recent method [78] can only get the average accu-

Table 1. The importance of pre-trained architectures from DomainNet [42]. The score is averaged on six domains.

Method	Backbone	Source	Target	Accuracy
RN-101 [20]	ResNet-101	✓	✗	26.6
MDD [28]	ResNet-101	✓	✓	28.6
MDD+SCDA [32]	ResNet-101	✓	✓	33.3
ViT-B [10]	ViT-B-16	✓	✗	**38.1**
SSRT [52]	ViT-B-16	✓	✓	45.2
PMTrans [78]	ViT-B-16	✓	✓	**52.4**

racy at 52.4%, which is far from satisfactory. Therefore, it is urgent to strengthen the performance of UDA algorithms on such large-scale datasets in real-world scenarios.

Moreover, the importance of pre-trained architectures is barely mentioned in the previous works [60, 78]. As shown in Table 1, we find a pre-trained ViT-B [10] fine-tuned only on the labeled source dataset has already outperformed other ResNet-based methods, including a complicated approach that combines MDD [28] and SCDA [32] (**38.1% vs 33.3%**). In other words, ViT-B can beat these methods even it has not seen any image from the unlabeled target domain. Therefore, from the above observation, we hypothesize that pre-trained model backbones and pre-trained data are an important missing piece for effective UDA in practical settings. Vision-Language pre-trained models (e.g., CLIP [44] and GLIP [29, 70]) have shown their power in learning generic and distinctive visual representations via language supervision, where each image will have more descriptive information compared to a single label [24, 39, 44]. However, there are very few works applying such models in UDA tasks [16].

Existing UDA methods include discrepancy minimization and adversarial training to learn domain-invariant representations by applying domain discriminators [18, 37, 40]. However, aligning domains and reducing the discrepancy could hurt the learning performance and result in the loss of semantic information [16, 53]. Such loss occurs due to the entangled nature of semantic and domain information, especially when dealing with intricate data distribution where the manifold structures are complex [1]. To alleviate this issue, another branch of methods [2, 5, 30] focuses on preserving the semantic information to highlight class discriminability. However, these techniques face a nuanced balance challenge between aligning domains and retaining semantic attributes, as these two objectives could be adversarial. Exploring disentangled semantic and domain representations could provide an alternative avenue, allowing for the potential disregard of domain alignment. Compared to conventional UDA methods that aim to learn domain-invariant representations by aligning the source and target domains, we hypothesize that VLMs are naturally good domain adapters due to the language alignment involved during training to disentangle the domain and class information: vision-language alignment loss has the potential to disentangle domain and class information. The main difference between traditional UDA and CLIP-based methods is summarized in Fig. 1.

However, these large-scale pre-trained VLMs have the following two challenges: 1) they have billions of parameters that require heavy computational resources to tune; 2) such big models may suffer from the overfitting problem, where the original knowledge learned from the 400M dataset (CLIP [44]) can significantly deteriorate through standard fine-tuning [27]. In this work, we propose an end-to-end pipeline to efficiently adapt these VLMs to the UDA tasks. We first freeze the text encoder and propose Prompt Task-dependent Tuning to tune the prompt for the downstream tasks carefully. Second, we freeze the vision encoder but propose a Visual Feature Refinement to fine-tune the visual representations instead of tuning the entire encoder. Lastly, we adapt pseudo-labeling from semi-supervised classifiers into language-based pseudo-labeling and incorporate domain information, called Domain-aware Pseudo-Labeling, to leverage the unlabeled target domain.

2. Related Works

Unsupervised Domain Adaptation (UDA) is initially studied for image classification tasks [2, 12]. Recent UDA methods aim to learn discriminative domain-invariant features and achieve domain alignment via metric learning and adversarial training. The metric learning-based methods use various metrics to reduce the domain discrepancy and learn the domain-invariant representations. For example, some works [25, 31, 38] use Maximum Mean Discrepancy (MMD) loss to measure the divergence between the source and target domain. On the other hand, adversarial training-based methods use an adversarial loss to encourage samples from different domains to be deprived from the domain information, thus the model can fully focus on the semantic attributes. Recent works [52, 60] found that the cross-attention module in Vision Transformer (ViT) [10] is beneficial to feature alignment. Hence these works [52, 60, 61] use ViT as their encoder and achieve superior results than CNNs.

Vision-Language Models (VLMs) have shown promising results in learning generic visual representations [24, 39, 44, 68] with language-vision alignment. Recent models scale up the architectures with Transformers [41, 54], advancing the power via contrastive representation learning, and web-scale training datasets [76]. For example, CLIP [44] was pre-trained on 400 million image-text pairs and achieved state-of-the-art performance in various downstream tasks [44, 64, 66, 67]. On the other hand, GLIP [29] was pre-trained on 27 million grounding data to leverage massive image-text pairs. It can achieve 60.8 Average Precision (AP) on COCO validation set after fine-tuning, show-

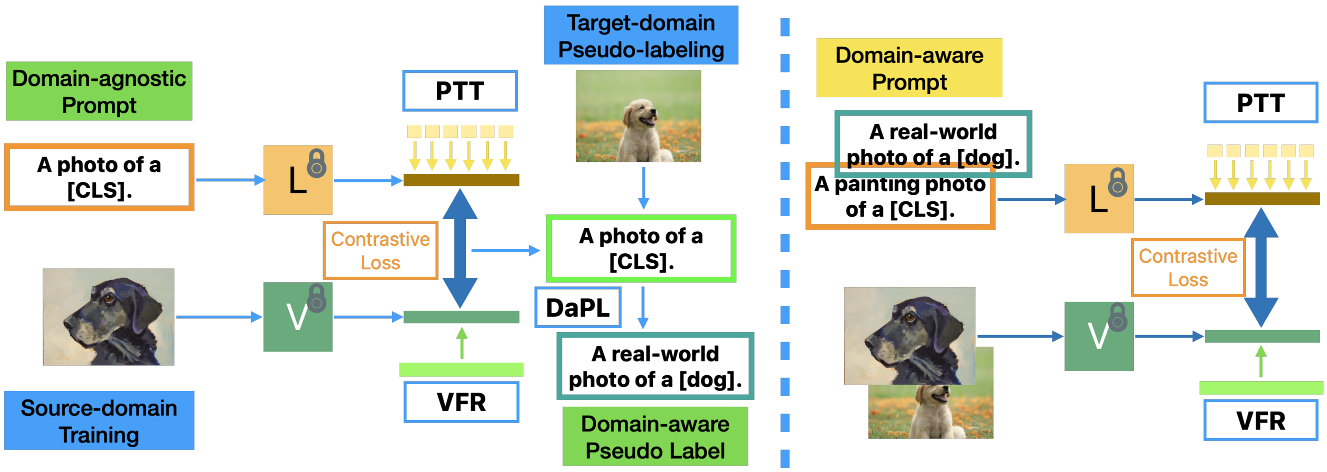

Figure 2. Overview. We propose Prompt Task-dependent Tuning (PTT) and Visual Feature Refinement (VFR) to adapt VLMs to the specific task. Then we design a three-stage scheme to achieve domain adaptation: 1) learn class representations on the source domain with domain-agnostic prompts; 2) generate pseudo labels on the target domain and convert them into domain-aware prompts; 3) joint training with domain-aware prompts from both source and target domains.

ing its semantic-rich learned representations. However, the best way to adapt VLMs for downstream tasks is still under study.

Efficient adaptation of VLMs is the key to the downstream tasks. We focus on parameter-efficient learning compared to full-model fine-tuning that may involve billions of parameters [65]. Existing works can be divided into two groups: prompt tuning (PT) [23, 39, 47, 49, 57, 75, 76] and adapter-style tuning (AT) [14, 65, 72]. PT-based methods focus on generating appropriate prompts for the downstream tasks. However, they freeze both vision and text encoders, limiting models' learning ability. AT-based methods focus on refining the vision or text features. For example, CLIP-adapter [14] designed a residual feature connection to preserve the original knowledge and learn the new knowledge. However, such methods have a hyper-parameter to set the residual amount for preservation, which requires additional experiments to tune manually. In this work, we aim to propose a module that can preserve pre-trained knowledge while keeping the maximum flexibility for gaining new visual concepts.

3. Methodology

Given a source domain of labeled data $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ and a target domain of unlabeled data $\mathcal{D}_t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{N_t}$, we aim to train a model to adapt from the source domain to the unlabeled target domain. N_s and N_t denote the source and target domain data samples, respectively. In this section, we first propose a parameter-efficient method for adapting VLMs to downstream tasks. We use CLIP [44] for classification and GLIP [29] for detection as two examples. As shown in Fig. 2, We freeze both vision and text encoders. Then, we propose Prompt Task-dependent Tuning (PTT) to fine-tune the prompt to fit the downstream task. Then, instead of fine-tuning the entire vision encoder, we propose Visual Feature Refinement (VFR) for CLIP and VFR+ designed in a pyramid architecture tailored for GLIP on the source domain to learn the class representations. Lastly, we propose Domain-Aware Pseudo-labeling to leverage the target domain and achieve domain disentanglement while preserving the semantic information.

3.1. Adapt VLM for UDA

We first modify CLIP to be suitable for UDA classification tasks. CLIP [44] has a vision encoder $f(\cdot)$ that maps images into low-dimensional visual representations and a text encoder $g(\cdot)$ that converts sentences into text representations. CLIP requires image-text pairs to train these two encoders jointly via contrastive loss [58]. Inspired by a recent work that fine-tuning should follow the same way as pre-training [19], we keep this contrastive loss by preparing image-text pairs for training instead of building new linear layers and using the loss associated with downstream tasks. This is advantageous to preserve the original knowledge and keep the language-vision alignment. Specifically, the text for CLIP can be "a [**DOMAIN**] photo of a [CLASS]", where [CLASS] is the class name and [**DOMAIN**] is the domain name in UDA tasks (e.g., a painting photo of a dog). In the testing phase, we employ CLIP's zero-shot inference approach, where we assess image representations by matching them against the classification weights produced by the text encoder, denoted as $\{\boldsymbol{\theta}_z\}_{z=1}^{K}$. By feeding K descriptions corresponding to K classes, we get the probability of the image belonging to the k-th category.

$$p_k = P(\hat{y}_z = k|\boldsymbol{x}) = \frac{\exp(\cos(\boldsymbol{\theta}_k, f(\boldsymbol{x})/T)}{\sum_{z=1}^{K} \exp(\cos(\boldsymbol{\theta}_z, f(\boldsymbol{x})/T)} \quad (1)$$

where T is the temperature parameter learned by CLIP, cos refers to cosine similarity [44]. We denote a vector of p_k as p (probability of a sample in a batch).

3.2. PTT: Prompt Task-dependent Tuning

For pre-trained VLMs, the text input (prompt) plays an essential role in downstream tasks [76]. For example, adding "a" before the class token can bring more than 5% of accuracy improvement on CLIP's zero-shot performance on Caltech101 [76]. This illustrates that even a slight perturbation in the prompt can result in a considerable difference in performance. On the other hand, adding task-relevant context and tuning the sentence structure can further improve the zero-shot accuracy. However, manual tuning can be labor-extensive, and there is no guarantee of obtaining the optimal structure for the downstream tasks. Inspired by GLIP [29], we incorporate a linear layer to convert the prompt tokens to fit our specific task. In the fine-tuning stage, we freeze the text encoder and will only tune this linear layer for the prompt, as shown in Fig. 2. The objective of the linear layer is to introduce trainable perturbations to the prompt, enhancing its adaptability to downstream tasks. Specifically, if the original language embedding is denoted as g, we add a linear layer to convert it to g' and the vision-language alignment is optimized via the fine-tuning stage.

3.3. VFR: Visual Feature Refinement

As the large-scale architectures of VLMs may involve billions of parameters, it is impractical to fine-tune the entire model on the downstream tasks in a low-data setting. Instead of fine-tuning the entire vision encoder, we focus on adapter-style tuning (AT) [14, 65, 72] to achieve the following two goals: 1) inheriting the large-scale pre-trained knowledge, which has been verified as transferable; 2) adapting and learning the task-specific knowledge from the limited data. Existing methods such as CLIP-Adapter [14] design a residual feature connection to fuse the pre-trained and new knowledge. In general, their tuning can be formulated as

$$\mathbf{f} = f(\boldsymbol{x}) + \alpha \mathbf{W}(f(\boldsymbol{x})), \quad (2)$$

where $f(\boldsymbol{x})$ is the pre-trained features from the vision encoder and α is the scaling factor. $\mathbf{W}$ is a trainable and lightweight module consisting of several layers. However, such methods [14, 65, 72] have two limitations. First, α is a hyper-parameter that needs to be tuned to control the weights between pre-trained and new knowledge, which may not be scalable if we have multiple downstream tasks and need to tune it for every new task. Second, they may heavily rely on pre-trained knowledge, which prevents a thorough exploitation of the new knowledge and thus results in limited learning flexibility compared to the full-model fine-tuning. Therefore, we aim to propose a visual feature refinement module to alleviate the above two issues.

3.3.1 VFR for CLIP

Inspired by [65] that focuses on tuning the text-based classifier, we aim to tune the vision encoder for learning new concepts in the downstream task, e.g., labeled source domain. To exploit new knowledge without being constrained by pre-trained knowledge, we modify Eqn. 2 with a set of tunable parameters $\mathbf{w}$, which is independent of the pre-trained knowledge to increase the flexibility of learning new visual concepts. Therefore, new class-level representations specific to the source domain can be appropriately supplemented with the pre-trained knowledge. We use a vector to store and tune the set of parameters $\mathbf{w}$, written as

$$\mathbf{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathbf{w}, \quad (3)$$

where we do not introduce any scaling ratio as an additional hyper-parameter. $\mathbf{w}$ is implemented as a linear parameter layer and will be self-scaled in the backpropagation, enabling reliable preservation of pre-trained knowledge and flexible exploitation of new visual concepts.

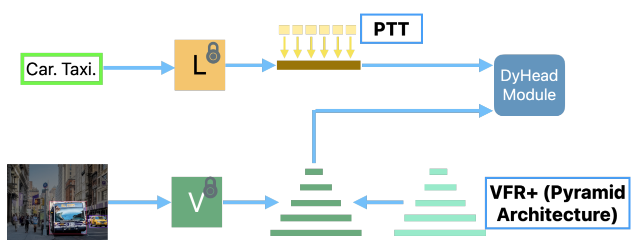

Figure 3. VFR+ for GLIP. We design a pyramid architecture to refine the visual features from the backbone and then input them into DyHead Module [7] for the detection objective. Note that every single layer in VFR+ is independent of each other to increase the flexibility of learning new knowledge.

3.3.2 VFR+ for GLIP

Although CLIP has shown strong image-level representations, it lacks a fine-grained understanding of images for object detection tasks [29], which indicates that CLIP may not be applicable to UDA detection tasks. We will focus on GLIP [29] pre-trained on 27M grounding data. In this subsection, we modify VFR as VFR+ for adapting GLIP for UDA object detection. The detection model typically has a vision backbone, Feature Pyramid Network (FPN), and the detector. We propose a pyramid architecture and modify

VFR for GLIP on object detection tasks, called VFR+. As GLIP uses Swin Transformer [36] as the vision backbone and DyHead [7] as the detector, we design a five-layer pyramid architecture to fine-tune the visual features outputted from the vision backbone, as shown in Fig. 3. Note that the five linear layers are independent of each other to increase the learning flexibility.

3.4. DaPL: Domain-aware Pseudo-Labeling for Domain Disentanglement

After we verify the efficient adaptation of VLMs to the downstream tasks, we design a three-stage pipeline to achieve domain disentanglement for the UDA tasks. Specifically, we first adapt VLMs to learn domain-agnostic semantic attributes, e.g., class discrimination. Then, we propose a language-based pseudo-labeling scheme on the unlabeled target domain to generate pseudo labels. Lastly, we convert them into domain-aware pseudo labels and perform domain-disentanglement training. We use CLIP as the example for this subsection.

Domain-agnostic task adaptation on the source domain. In this stage, we aim to focus on the class representations and adapt the VLMs to learn specific classes in the downstream tasks. In other words, we will disregard the domain information but rather entirely focus on adapting VLMs to semantic attributes. As the source domain is labeled, we prepare a domain-agnostic prompt for each image as "A photo of a [CLS]", e.g., "A photo of a dog". This is similar to few-shot learning with VLMs. A recent paper found that fine-tuning should use the same loss as the pre-training [19]. Therefore, we keep the contrastive loss used in CLIP to train the tuning layers (PTT and VFR) so that CLIP can be well fine-tuned to this specific task (learn the required classes in the task). The training objective is shown below:

$$\begin{aligned}\mathcal{L}_{\text{con}} := \sum_{i=1}^{B} - \log \frac{\exp\left(\mathbf{f}(\boldsymbol{x_i}) \cdot \mathbf{g}(\boldsymbol{t_i})\right)}{\sum_{j=1}^{B} \exp\left(\mathbf{f}(\boldsymbol{x_i}) \cdot \mathbf{g}(\boldsymbol{t_j})\right)} + \\ \sum_{i=1}^{B} - \log \frac{\exp\left(\mathbf{f}(\boldsymbol{x_i}) \cdot \mathbf{g}(\boldsymbol{t_i})\right)}{\sum_{j=1}^{B} \exp\left(\mathbf{f}(\boldsymbol{x_j}) \cdot \mathbf{g}(\boldsymbol{t_i})\right)},\end{aligned} \tag{4}$$

where we set a batch with B images with their corresponding prompts $D = \{(\boldsymbol{x_1}, \boldsymbol{t_1}), \dots (\boldsymbol{x_B}, \boldsymbol{t_B})\}$. This is the same pre-training objective in CLIP [44], which could be interpreted as undergoing training using a substitute classification task that comprises one image and B classes derived from text embeddings, and conversely, one text and B classes obtained from image embeddings.

Domain-aware pseudo-labeling on the target domain. Pseudo-labeling is a common way used in semi-supervised learning [50, 69] to leverage unlabeled data. Subsequently, it is introduced in UDA tasks [9, 52, 60] to leverage the unlabeled target domain. However, previous pseudo-labeling is based on traditional classifiers: use the prediction generated from the classifier on the unlabeled sample and assign the artificial label as supervision during the self-training process. Now we introduce our adaptation of pseudo-labeling in a format of pseudo prompt for VLMs. Since our prompts in the source-domain fine-tuning are domain-agnostic, we first prepare a domain-agnostic prompt for the inference, such as "A photo of a [CLS]". After CLIP's inference on the target domain (Eqn. 1), we will complete the pseudo prompt with the classification results. Therefore, the prompt will be "A photo of a [dog]". Then we feed the domain information from the target into the prompt and further refine it as "A **real-world** photo of a [dog]" as our final pseudo prompt for the unlabeled target domain.

Domain-disentanglement training for domain adaptation. We propose to use VLMs for the UDA tasks by exploiting its mixed power from the visual encoder $f(\cdot)$ and text encoder $g(\cdot)$. Specifically, both encoders can transform the input pair into two disentangled latent representations: domain representation and intrinsic class representation. We argue that this structure may naturally benefit the UDA tasks: the similarity score will be optimized (the distance between the image and text embeddings will be minimized) if the domain and the class representations are aligned. From the above sections, we generate domain-aware pseudo prompts from the target domain. Meanwhile, we will also convert domain-agnostic prompts in the source domain into domain-aware prompts. Therefore, our final fine-tuning data will be from both the source and target domains. Take "painting $\rightarrow$ real-world" as one example. We have "A painting photo of a [CLS]" for the source domain and "A real-world photo of a [CLS]" as the pseudo prompt for the target domain. We optimize the training objective by aligning the text and vision encoders via Eqn. 4, which can disentangle the domain information while learning new concepts via the loss optimization process. Optimizing this contrastive loss (Eqn. 4) will maximize the distance between negative pairs while minimizing the distance between positive pairs. The domain and class representations can be disentangled naturally, which subsequently maximizes the probability of the correct label in Eqn. 1. On the other hand, a recent work [19] found that keeping the contrastive loss in fine-tuning will help preserve the original knowledge.

4. Experimental Results

UDA classification datasets. For UDA classification tasks, we select four popular benchmarks. **(1) VisDA-2017** [43] sets 152k synthetic images as the source domain and 55k real-world images of 12 categories as the target domain. **(2) Office-Home** [55] includes 15,500 images of 65 categories from four domains: Real-world (Rw), Art (Ar), Clipart (Cl), and Product (Pr) images. **(3) Office-31** [46] has three domains: Webcam (W), Amazon (A),

Table 2. Accuracies (%) on **VisDA-2017**. "-B" indicates ViT-B backbone. See full table in Appendix.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
RN-101 [20]	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
CaCo [22]	90.4	80.7	78.8	57.0	88.9	87.0	81.3	79.4	88.7	88.1	86.8	63.9	80.9
SUDA [71]	91.5	79.7	71.9	66.5	88.5	81.1	85.6	79.5	86.2	86.5	79.9	74.3	80.9
MCC+NWD [3]	96.1	82.7	76.8	71.4	92.5	96.8	88.2	81.3	92.2	88.7	84.1	53.7	83.7
SDAT [45]	95.8	85.5	76.9	69.0	93.5	97.4	88.5	78.2	93.1	91.6	86.3	55.3	84.3
MSGD [59]	97.5	83.4	84.4	69.4	95.9	94.1	90.9	75.5	95.5	94.6	88.1	44.9	84.6
CAN [26]	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
AaD [62]	97.4	90.5	80.8	76.2	97.3	96.1	89.8	82.9	95.5	93.0	92.0	64.7	88.0
SDAT+MIC [21]	96.7	88.5	84.2	74.3	96.0	96.3	90.2	81.2	94.3	95.4	88.9	56.6	86.9
Ours (RN-101)	97.2	89.3	87.6	83.1	98.4	95.4	92.2	82.5	94.9	93.2	91.3	64.7	**89.2**
ViT-B [10]	99.1	60.7	70.6	82.7	96.5	73.1	97.1	19.7	64.5	94.7	97.2	15.4	72.6
TVT-B [61]	92.9	85.6	77.5	60.5	93.6	98.2	89.4	76.4	93.6	92.0	91.7	55.7	83.9
SHOT-B [60]	97.9	90.3	86.0	73.4	96.9	98.8	94.3	54.8	95.4	87.1	93.4	62.7	85.9
CDTrans [60]	97.1	90.5	82.4	77.5	96.6	96.1	93.6	88.6	97.9	86.9	90.3	62.8	88.4
SSRT-B [52]	98.9	87.6	89.1	84.8	98.3	98.7	96.3	81.1	94.9	97.9	94.5	43.1	88.8
SDAT-B [45]	98.4	90.9	85.4	82.1	98.5	97.6	96.3	86.1	96.2	96.7	92.9	56.8	89.8
PMTrans [78]	98.9	93.7	84.5	73.3	99.0	98.0	96.2	67.8	94.2	98.4	96.6	49.0	87.5
Ours-B	98.4	94.3	89.0	85.4	98.5	98.3	96.1	86.3	95.1	95.2	92.5	70.9	**91.7**

Table 3. Accuracies (%) on **Office-Home**. "-B" indicates ViT-B. See full table in Appendix.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
RN-50 [20]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
SDAT [45]	58.2	77.1	82.2	66.3	77.6	76.8	63.3	57.0	82.2	74.9	64.7	86.0	72.2
MSGD [59]	58.7	76.9	78.9	70.1	76.2	76.6	69.0	57.2	82.3	74.9	62.7	84.5	72.4
AaD [62]	59.3	79.3	82.1	68.9	79.8	79.5	67.2	57.4	83.1	72.1	58.5	85.4	72.7
KUDA [51]	58.2	80.0	82.9	71.1	80.3	80.7	71.3	56.8	83.2	75.5	60.3	86.6	73.9
DAPL [16]	54.1	84.3	84.8	74.4	83.7	85.0	74.5	54.6	84.8	75.2	54.7	83.8	74.5
Ours (RN-50)	58.1	85.0	84.5	77.4	85.0	84.7	76.5	58.8	85.7	75.9	60.4	86.4	**76.5**
ViT-B [10]	54.7	83.0	87.2	77.3	83.4	85.5	74.4	50.9	87.2	79.6	53.8	88.8	75.5
CDTrans [60]	68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
TVT-B [61]	74.9	86.8	89.5	82.8	88.0	88.3	79.8	71.9	90.1	85.5	74.6	90.6	83.6
SDAT-B [45]	70.8	87.0	90.5	85.2	87.3	89.7	94.1	70.7	90.6	88.3	75.5	92.1	84.3
SSRT-B [52]	75.2	89.0	91.1	85.1	88.3	89.9	85.0	74.2	91.3	85.7	78.6	91.8	85.4
SDAT+MIC [21]	80.2	87.3	91.1	87.2	90.0	90.1	83.4	75.6	91.2	88.6	78.7	91.4	86.2
Ours-B	78.2	90.4	91.0	87.5	91.9	92.3	86.7	79.7	90.9	86.4	79.4	93.5	**87.3**

and DSLR (D). **(4) DomainNet** [42] is the most challenging and largest UDA benchmark that has 0.6M images of 345 categories from six domains: Real-world (rel), Quickdraw (qdr), Painting (pnt), Infograph (inf), Clipart (clp), and Sketch (skt) images. For Office-Home, Office-31, and DomainNet, we will traverse every domain as the source domain and the rest as the target domains. For example, in Table 3, we use Art (Ar) as the source domain and then use Clipart (Cl), and Product (Pr) as the target domains.

Table 4. Accuracies (%) on **Office-31**.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
ViT-B [10]	91.2	99.2	100.	90.4	81.1	80.6	90.4
CDAN+TN [56]	95.7	98.7	100.	94.0	73.4	74.2	89.3
SHOT-B [35]	94.3	99.0	100.	95.3	79.4	80.2	91.4
CDTrans [60]	96.7	99.0	100.	97.0	81.1	81.9	92.6
SSRT-B [52]	97.7	99.2	100.	98.6	83.5	82.2	93.5
TVT-B [61]	96.4	99.4	100.	96.4	84.9	86.1	93.8
Ours-B	98.1	99.4	100.	98.7	84.4	85.5	94.4

UDA detection datasets. For UDA object detection, we follow the previous works [2, 9, 34] and test it on the following settings. **(1). Weather Shift: Cityscapes → Foggy Cityscapes.** We evaluate our method on the domain shift from normal to adverse weather (foggy) for this setting. We use the labeled images from Cityscapes [6] as the source domain and then use Foggy Cityscapes [48] as the target domain. **(2). Camera Shit: KITTI → Cityscapes.** In this setting, we consider different cameras in domain adaptation. We use KITTI [17] as the source domain (collected from vehicle-mounted cameras) and Cityscapes [6] as the target domain. Following the recent works [2, 9, 34], we report the performance of the car category.

5. Results

5.1. Adapt CLIP for UDA Classification

VisDA-2017. Table 2 summarizes the accuracies of different methods on VisDA-2017 [43]: we use "-B" to refer to ViT-B backbone and "RN-101" to refer to ResNet-101 backbone. To have a fair comparison, we first use RN-101 and compare our results with the recent algo-

Table 5. Accuracies (%) on **DomainNet**. In each sub-table, the column-wise means source domain and the row-wise means target domain. “-B” indicates ViT-B (except CDTrans uses DeiT).

ResNet-101 [20]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	19.3	37.5	11.1	52.2	41.0	32.2
inf	30.2	-	31.2	3.6	44.0	27.9	27.4
pnt	39.6	18.7	-	4.9	54.5	36.3	30.8
qdr	7.0	0.9	1.4	-	4.1	8.3	4.3
rel	48.4	22.2	49.4	6.4	-	38.8	33.0
skt	46.9	15.4	37.0	10.9	47.0	-	31.4
Avg.	34.4	15.3	31.3	7.4	40.4	30.5	26.6

MIMTFL [13]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	15.1	35.6	10.7	51.5	43.1	31.2
inf	32.1	-	31.0	2.9	48.5	31.0	29.1
pnt	40.1	14.7	-	4.2	55.4	36.8	30.2
qdr	18.8	3.1	5.0	-	16.0	13.8	11.3
rel	48.5	19.0	47.6	5.8	-	39.4	32.1
skt	51.7	16.5	40.3	12.3	53.5	-	34.9
Avg.	38.2	13.7	31.9	7.2	45.0	32.8	28.1

CDAN [37]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	20.4	36.6	9.0	50.7	42.3	31.8
inf	27.5	-	25.7	1.8	34.7	20.1	22.0
pnt	42.6	20.0	-	2.5	55.6	38.5	31.8
qdr	21.0	4.5	8.1	-	14.3	15.7	12.7
rel	51.9	23.3	50.4	5.4	-	41.4	34.5
skt	50.8	20.3	43.0	2.9	50.8	-	33.6
Avg.	38.8	17.7	32.8	4.3	41.2	31.6	27.7

MDD+SCDA [32]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	20.4	43.3	15.2	59.3	46.5	36.9
inf	32.7	-	34.5	6.3	47.6	29.2	30.1
pnt	46.4	19.9	-	8.1	58.8	42.9	35.2
qdr	31.1	6.6	18.0	-	28.8	22.0	21.3
rel	55.5	23.7	52.9	9.5	-	45.2	37.4
skt	55.8	20.1	46.5	15.0	56.7	-	38.8
Avg.	44.3	18.1	39.0	10.8	50.2	37.2	33.3

ViT-B [10]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	27.2	53.1	13.2	71.2	53.3	43.6
inf	51.4	-	49.3	4.0	66.3	41.1	42.4
pnt	53.1	25.6	-	4.8	70.0	41.8	39.1
qdr	30.5	4.5	16.0	-	27.0	19.3	19.5
rel	58.4	29.0	60.0	6.0	-	45.8	39.9
skt	63.9	23.8	52.3	14.4	67.4	-	44.4
Avg.	51.5	22.0	46.1	8.5	60.4	40.3	38.1

CD-Trans [60]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	29.4	57.2	26.0	72.6	58.1	48.7
inf	57.0	-	54.4	12.8	69.5	48.4	48.4
pnt	62.9	27.4	-	15.8	72.1	53.9	46.4
qdr	44.6	8.9	29.0	-	42.6	28.5	30.7
rel	66.2	31.0	61.5	16.2	-	52.9	45.6
skt	69.0	29.6	59.0	27.2	72.5	-	51.5
Avg.	59.9	25.3	52.2	19.6	65.9	48.4	45.2

PMTrans [78]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	34.2	62.7	32.5	79.3	63.7	54.5
inf	67.4	-	61.1	22.2	78.0	57.6	57.3
pnt	69.7	33.5	-	23.9	79.8	61.2	53.6
qdr	54.6	17.4	38.9	-	49.5	41.0	40.3
rel	74.1	35.3	70.0	25.4	-	61.1	53.2
skt	73.8	33.0	62.6	30.9	77.5	-	55.6
Avg.	67.9	30.7	59.1	27.0	72.8	56.9	52.4

SSRT -B [52]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	33.8	60.2	19.4	75.8	59.8	49.8
inf	55.5	-	54.0	9.0	68.2	44.7	46.3
pnt	61.7	28.5	-	8.4	71.4	55.2	45.0
qdr	42.5	8.8	24.2	-	37.6	33.6	29.3
rel	69.9	37.1	66.0	10.1	-	58.9	48.4
skt	70.6	32.8	62.2	21.7	73.2	-	52.1
Avg.	60.0	28.2	53.3	13.7	65.3	50.4	45.2

Ours -B	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	70.2	72.4	73.1	75.5	74.9	73.2
inf	54.8	-	54.6	50.8	56.1	56.2	54.5
pnt	69.9	68.5	-	64.3	74.6	70.2	69.5
qdr	35.3	16.6	29.5	-	30.2	32.3	28.8
rel	85.1	82.2	83.0	81.2	-	80.3	82.4
skt	67.4	65.9	66.4	62.3	65.6	-	65.5
Avg.	62.5	60.7	61.2	66.3	60.4	62.8	62.3

Table 6. Results on UDA detection: **Cityscapes→Foggy Cityscapes (%)**. ZS refers to zero-shot, SO refers to source-only setting.

Method	Reference	person	rider	car	truck	bus	train	motor	bike	mAP
SIGMA [33]	CVPR’22	44.0	43.9	60.3	31.6	50.4	51.5	31.7	40.6	44.2
AT [34]	CVPR’22	45.5	55.1	64.2	35.0	56.3	54.3	38.5	51.9	50.9
OADA [63]	ECCV’22	47.8	46.5	62.9	32.1	48.5	50.9	34.3	39.8	45.4
MGA [77]	CVPR’22	45.7	47.5	60.6	31.0	52.9	44.5	29.0	38.0	43.6
MIC [21]	CVPR’23	50.9	55.3	67.0	33.9	52.4	33.7	40.6	47.5	47.6
HT [9]	CVPR’23	52.1	55.8	67.5	32.7	55.9	49.1	40.1	50.3	50.4
GLIP ZS [29]	-	36.0	11.2	55.1	20.6	39.2	1.5	28.8	40.3	29.1
GLIP SO [29]	-	52.5	53.1	63.3	37.8	53.6	43.1	38.0	49.3	48.8
Ours	-	54.1	56.7	66.5	42.1	57.5	50.2	44.3	53.3	**53.1**

rithms [3,21,22,26,45,59,62,71]. We show that our method consistently improves almost all classes and 4.2% of improvement on the average accuracy compared to SDAT [52]. Then we follow [52,60,78] to use ViT-B as the encoder and show the superiority of our method under this setting.

Office-Home/31. We summarize the results on Office-Home [55] in Table 3. We first follow the recent methods [16,45,51,62] to use RN-50 (ResNet-50) as the image encoder and show the superiority of our framework with at least 2.0% of improvement compared to DAPL [16], a recent work adapting CLIP to UDA. Then we follow ViT-based methods [21,45,52,60] to use ViT-B as the encoder. It is worthwhile to mention that our framework can consistently improve across different domains. We have similar observations in Office-31 [46] in Table 4.

DomainNet. On the most challenging DomainNet [42] (as shown in Table 5), we achieve 62.3% of average accuracy, with an impressive 9.9% improvement over PMTrans [78]. Some domains in this benchmark have large gaps from others, especially *inf* and *qdr*. Transferring the knowledge from other domains to these two is difficult due to the domain gap. On the other hand, the distributions are heterogeneous and can be imbalanced among different domains, which makes this benchmark more difficult. However, our proposed method can achieve improvement in almost all settings. We conclude that the VLMs are naturally good at domain disentanglement, and the large-scale pretraining is beneficial to UDA tasks.

5.2. Adapt GLIP for UDA Detection

Adverse Weather Adaptation. Object detectors may face various weather conditions, and adverse weather conditions can downgrade their performance. Therefore, for this setting, we evaluate our model on weather shift: from normal to adverse weather (foggy). The results are summarized in Table 6. Our proposed methods can bring 2.7%

improvement on mAP compared to HT [9]. Moreover, our GLIP adapter consistently improves in almost all categories, showing the power of large-scale pre-training. GLIP ZS refers to GLIP zero-shot performance on the target domain. GLIP SO refers to GLIP full-model fine-tuning on the source domain only. We can see GLIP itself has a strong ability during fine-tuning, achieving 48.8% of mAP. Our proposed adaption can further improve it to 53.1%.

Camera Shift Adaptation. Real-world cameras have significantly different configurations (e.g., resolutions, positions), and such differences may affect the detectors' performance. Following the practice of previous work [2, 9, 33, 74, 77], we use KITTI $\rightarrow$ Cityscapes to study the effectiveness on camera shift adaptation: we only train and test the detectors for the sharing category "Car" in these two datasets. The results are summarized in Table 7. Our GLIP-adapted detector can achieve +1.9% compared to the combination of PT and CMT [2].

Table 7. UDA detection task across different cameras (from KITTI to Cityscapes).

Method	Reference	AP (Car)	Gain
Source	-	40.3	-
MGA [77]	CVPR'22	48.5	+8.2
TIA [74]	CVPR'22	44.0	+3.7
SIGMA [33]	CVPR'22	45.8	+5.5
OADA [63]	ECCV'22	47.8	+7.5
PT [4]	ICML'22	60.2	+19.9
HT [9]	CVPR'23	60.3	+20.0
PT + CMT [2]	CVPR'23	64.3	+24.0
Ours	-	66.2	+25.9

5.3. Ablation Studies

Ablation studies for PTT, VFR, and DaPL. We summarize the ablation studies in Table 8. CLIP zero-shot performance is strong in this case at 82.3%. Our PTT fine-tuned on the source domain can achieve 2.2% improvement. After we apply VFR to refine the visual features, we get 3.4% improvement. Lastly, we include the target domain with Domain-aware Pseudo-Labeling (DaPL) and achieve 91.7%. This shows the effectiveness of the proposed modules for adapting VLMs for UDA classification tasks.

Effectiveness of adaptation and the comparison with full-model fine-tuning. As VLMs have rich semantic knowledge, it is essential to verify if we can preserve CLIP's performance and achieve knowledge fusion. In Office-Home, our improvement in every single domain is consistent compared to the original CLIP. For the Clipart domain, we achieved over 14% of improvement, showing the effectiveness of the proposed method. On the other hand, we compare the proposed adaptation method with full-model fine-tuning (FMFT). As shown in Table 8, our

Table 8. Ablation study on VisDA-2017 with ViT-B backbone. (FMFT refers to Full-Model Fine-Tuning.)

#	Source	Target	PTT	VFR	DaPL	FMFT	Accuracy
1	✗	✗	✗	✗	✓	✗	82.3%
2	✓	✗	✓	✗	✗	✗	84.5%
3	✓	✗	✓	✓	✗	✗	87.9%
4	✓	✓	✓	✓	✓	✗	**91.7%**
5	✓	✓	✗	✗	✗	✓	88.1 %
6	✓	✓	✗	✗	✓	✓	**92.1** %

adaptation can achieve 87.9% if we only use the source domain, which is competitive compared to FMFT (88.1%). With DaPL on the target domain, FMFT can further achieve 92.%. Considering that we only train a few layers instead of the full model, our adaptation method is effective and practical in reducing the computational cost.

Effectiveness of adapting VLMs with VFR. To test the effectiveness of our adaptation way, we compare it in **few-shot learning** settings with other VLM adaptation methods [14, 72, 76]. The results are summarized in Table 9. Compared to the recent methods refining the visual features [14,72,76], we achieve superior results via VFR under 16-shot learning setting.

Table 9. Few-shot classification on ImageNet [8].

Methods	Shot	Accuracy
CLIP [44]	0	62.53%
CLIP + CoOp [76]	16	66.60%
CLIP-Adapter [14]	16	65.39%
Tip-Adapter [72]	16	64.78%
Tip-Adapter-F [72]	16	68.56%
Ours	16	69.15%

6. Discussion

In this work, we apply Vision-Language Models (VLMs) for UDA tasks: UDA classification and UDA detection. We verify that VLMs are naturally advantageous in domain disentanglement and thus can achieve domain alignment and semantic-attributes retainment. We propose efficient adaptation for VLMs on both prompt tuning and visual feature refinement. We formulate domain-aware pseudo-labeling for VLMs by using zero-shot prediction and fuse domain information. Extensive experimental results on six challenging benchmarks verify the effectiveness of our proposed method on both UDA classification and detection, especially on large-scale datasets.

Limitations. As VLMs are large-scale pre-trained, the comparison may not be fully fair. Our main focus is to introduce VLMs for UDA tasks and show the impact of language supersion on vision tasks.

References

[1] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, page 2060. NIH Public Access, 2019. 2

[2] Shengcao Cao, Dhiraj Joshi, Liang-Yan Gui, and Yu-Xiong Wang. Contrastive mean teacher for domain adaptive object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23839–23848, 2023. 1, 2, 6, 8

[3] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *CVPR*, pages 7181–7190, 2022. 1, 6, 7

[4] Meilin Chen, Weijie Chen, Shicai Yang, Jie Song, Xinchao Wang, Lei Zhang, Yunfeng Yan, Donglian Qi, Yueting Zhuang, Di Xie, and Shiliang Pu. Learning domain adaptive object detection with probabilistic teacher. In *ICML*, 2022. 8

[5] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pages 1081–1090, 2019. 2

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 6

[7] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2021. 4, 5

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[9] Jinhong Deng, Dongli Xu, Wen Li, and Lixin Duan. Harmonious teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23829–23838, 2023. 1, 5, 6, 7, 8

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1, 2, 6, 7

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015. 1

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17:2096–2030, 2016. 2

[13] Jian Gao, Yang Hua, Guosheng Hu, Chi Wang, and Neil M Robertson. Reducing distributional uncertainty by mutual information maximisation and transferable feature learning. In *ECCV*, pages 587–605. Springer, 2020. 7

[14] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 3, 4, 8

[15] Zhiqiang Gao, Shufei Zhang, Kaizhu Huang, Qiufeng Wang, and Chaoliang Zhong. Gradient distribution alignment certificates better adversarial domain adaptation. In *ICCV*, pages 8937–8946, 2021. 1

[16] 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. 2, 6, 7

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 6

[18] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, volume 27, 2014. 2

[19] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023. 3, 5

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 6, 7

[21] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11721–11732, 2023. 1, 6, 7

[22] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *CVPR*, pages 1203–1214, 2022. 6, 7

[23] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022. 3

[24] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916, 2021. 2

[25] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 2

[26] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019. 6, 7

[27] Zhengfeng Lai, Noranart Vesdapunt, Ning Zhou, Jun Wu, Cong Phuoc Huynh, Xuelu Li, Kah Kuen Fu, and Chen-Nee

Chuah. Padclip: Pseudo-labeling with adaptive debiasing in clip for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16155–16165, 2023. 2

[28] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. Maximum density divergence for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3918–3930, 2020. 2

[29] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2, 3, 4, 7

[30] Shuang Li, Shiji Song, Gao Huang, Zhengming Ding, and Cheng Wu. Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE transactions on image processing*, 27(9):4260–4273, 2018. 2

[31] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11516–11525, 2021. 2

[32] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *ICCV*, pages 9102–9111, 2021. 1, 2, 7

[33] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5300, 2022. 7, 8

[34] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7581–7590, 2022. 6, 7

[35] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039, 2020. 1, 6

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5

[37] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1645–1655, 2018. 2, 7

[38] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017. 2

[39] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, pages 5206–5215, 2022. 2, 3

[40] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020. 2

[41] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *ECCV*, pages 529–544. Springer, 2022. 2

[42] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 1, 2, 6, 7

[43] 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, 6

[44] 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*, pages 8748–8763, 2021. 2, 3, 4, 5, 8

[45] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *ICML*, pages 18378–18399, 2022. 6, 7

[46] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010. 1, 5, 7

[47] Aadarsh Sahoo, Anshuman Senapati, Abir Das, Yoon Kim, Rogerio Feris, and Rameswar Panda. Frustratingly simple contrastive prompt tuning for vision-language models. 3

[48] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018. 6

[49] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022. 3

[50] 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. In *NeurIPS*, 2020. 5

[51] Tao Sun, Cheng Lu, and Haibin Ling. Prior knowledge guided unsupervised domain adaptation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 639–655. Springer, 2022. 6, 7

[52] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *CVPR*, pages 7191–7200, 2022. 1, 2, 5, 6, 7

[53] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, pages 8725–8735, 2020. 2

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 2

[55] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 5, 7

[56] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *NeurIPS*, 2019. 6

[57] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022. 3

[58] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *IEEE International Conference on Computer Vision*, 2019. 3

[59] Haifeng Xia, Taotao Jing, and Zhengming Ding. Maximum structural generation discrepancy for unsupervised domain adaptation. *PAMI*, 2022. 6, 7

[60] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *ICLR*, 2022. 1, 2, 5, 6, 7

[61] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 520–530, 2023. 2, 6

[62] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems*, 2022. 6, 7

[63] Jayeon Yoo, Inseop Chung, and Nojun Kwak. Unsupervised domain adaptation for one-stage object detector using offsets to bounding box. In *European Conference on Computer Vision*, pages 691–708. Springer, 2022. 7, 8

[64] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. 2

[65] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. 3, 4

[66] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021. 2

[67] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022. 2

[68] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, pages 18123–18133, 2022. 2

[69] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS*, 34:18408–18419, 2021. 5

[70] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022. 2

[71] Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. Spectral unsupervised domain adaptation for visual recognition. In *CVPR*, pages 9829–9840, 2022. 6, 7

[72] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *ECCV*, pages 493–510. Springer, 2022. 3, 4, 8

[73] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, pages 7404–7413, 2019. 1

[74] Liang Zhao and Limin Wang. Task-specific inconsistency alignment for domain adaptive object detection. In *CVPR*, 2022. 8

[75] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 3

[76] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *Int. J. Comput. Vis.*, pages 1–12, 2022. 2, 3, 4, 8

[77] Wenzhang Zhou, Dawei Du, Libo Zhang, Tiejian Luo, and Yanjun Wu. Multi-granularity alignment domain adaptation for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9581–9590, 2022. 7, 8

[78] Jinjing Zhu, Haotian Bai, and Lin Wang. Patch-mix transformer for unsupervised domain adaptation: A game perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3561–3571, 2023. 1, 2, 6, 7