# PADCLIP: Pseudo-labeling with Adaptive Debiasing in CLIP for Unsupervised Domain Adaptation

Zhengfeng Lai[1*] Noranart Vesdapunt[2*] Ning Zhou[2] Jun Wu[2]
Cong Phuoc Huynh[2] Xuelu Li[2] Kah Kuen Fu[2] Chen-Nee Chuah[1]
[1]University of California, Davis [2]Amazon
[1]{lzhengfeng, chuah}@ucdavis.edu [2] {solves, ningzho, jwum, conghuyn, xueluli, kahkuen}@amazon.com

## Abstract

*Traditional Unsupervised Domain Adaptation (UDA) leverages the labeled source domain to tackle the learning tasks on the unlabeled target domain. It can be more challenging when a large domain gap exists between the source and the target domain. A more practical setting is to utilize a large-scale pre-trained model to fill the domain gap. For example, CLIP shows promising zero-shot generalizability to bridge the gap. However, after applying traditional fine-tuning to specifically adjust CLIP on a target domain, CLIP suffers from catastrophic forgetting issues where the new domain knowledge can quickly override CLIP's pre-trained knowledge and decreases the accuracy by half. We propose Catastrophic Forgetting Measurement (CFM) to adjust the learning rate to avoid excessive training (thus mitigating the catastrophic forgetting issue). We then utilize CLIP's zero-shot prediction to formulate a Pseudo-labeling setting with Adaptive Debiasing in CLIP (PADCLIP) by adjusting causal inference with our momentum and CFM. Our PADCLIP allows end-to-end training on source and target domains without extra overhead. We achieved the best results on four public datasets, with a significant improvement (+18.5% accuracy) on DomainNet.*

## 1. Introduction

Unsupervised Domain Adaptation (UDA) proposes to reduce data annotation costs by leveraging a labeled source domain to transfer the knowledge into an unlabeled target domain [11, 27, 49, 60, 73]. Prior UDA works focus on bridging the domain gap between source and target domains [4, 12, 25, 27], or increasing network capacity [49, 60] by changing a convolutional neural network (e.g., ResNet [14]) to Vision Transformer (ViT) [9]. All of these past methods are pre-trained on ImageNet [7], but large-scale pre-training is becoming practical and achieves superior performance in many fields [41, 64, 65, 67]. In theory, if the pre-trained dataset is large enough, the domain gap between source and target domains could be bridged by the pre-trained dataset itself. Hence, we argue that large-scale pre-training is an important missing part of UDA.

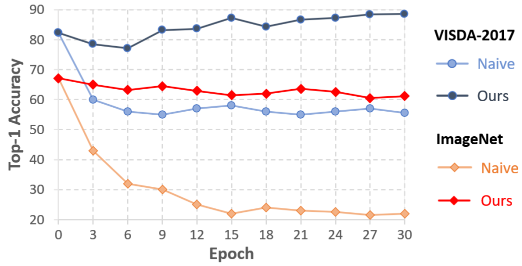


Figure 1: Catastrophic Forgetting. We naively fine-tune CLIP (ResNet-101) on VisDA-2017 source domain training set, and test it on validation sets of VisDA-2017 target domain and ImageNet-1K. CLIP forgets pre-trained knowledge (ImageNet accuracy -45%), resulting in -27% VisDA-2017 accuracy. Our PADCLIP mitigates catastrophic forgetting issues to achieve +6% VisDA-2017 accuracy.

We choose CLIP [41], a vision-language model pre-trained on 400 million image-text pairs. Without fine-tuning, CLIP outperforms SSRT [49], a state-of-the-art UDA method on DomainNet [38]. This is thanks to the large-scale training set, which allows CLIP to disentangle object class from object domain (e.g., "a photo of a dog" vs "a sketch of a dog"): the language supervision in the form of a sentence used by CLIP is more descriptive than a single class label. However, on VisDA-2017, CLIP without fine-tuning underperforms previous work, SDAT [42]. This is because the synthetic data generated from the real-world domain do not exist in CLIP's training set, so we still need to fine-tune CLIP to adapt it for a specific domain task.

We first adopt the traditional approach to fine-tune CLIP on VisDA-2017 [39] but found that CLIP suffers from catastrophic forgetting issues. As shown in Fig.1: before fine-tuning, CLIP has a strong representation power that

[*]Equal contributions. This work was done at Amazon.

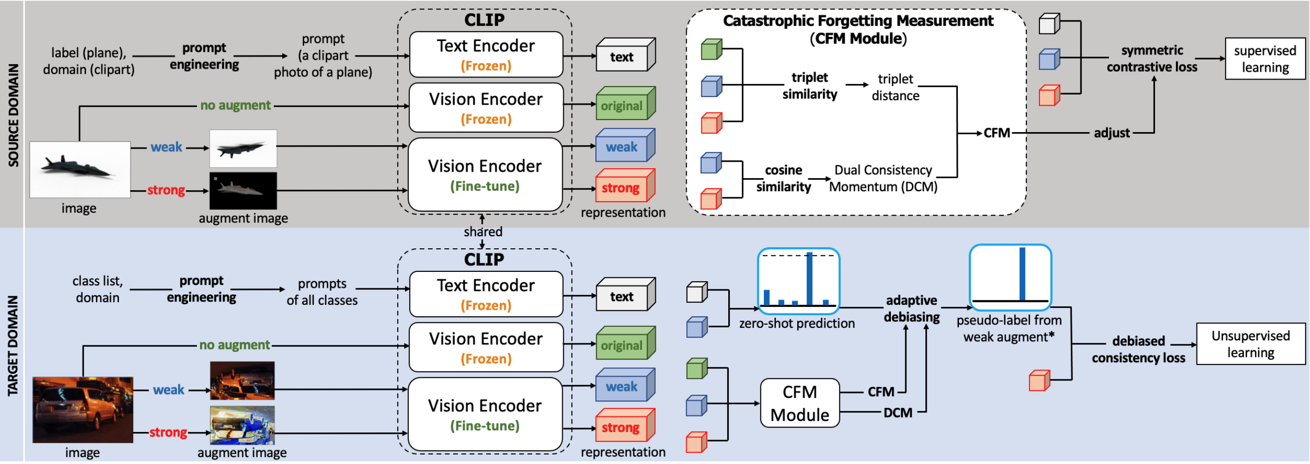


Figure 2: Overview. For the source domain, we convert the label and domain name into a prompt, and obtain text and image representations to train CLIP in a supervised manner. We use CLIP's original representation, and weak/strong augmented representations to measure CFM to adjust the learning rate for mitigating catastrophic forgetting issues. For the target domain, we use zero-shot prediction in CLIP to obtain pseudo-labels and adaptively debias them with CFM (adjust debias factor) and DCM (adjust momentum) for unsupervised learning. *Pseudo-label is converted into a **prompt** to obtain text representation.

can achieve 67% top-1 ImageNet accuracy, while dropping to 22% after fine-tuning on VisDA-2017. The loss of CLIP's representation power causes the accuracy degradation on VisDA-2017. To counter this, it is possible to preserve CLIP's representation power by fine-tuning both CLIP and VisDA-2017 datasets jointly, but CLIP requires several weeks to train a single setting (DomainNet has 30 settings, so it takes a year for a single experiment). Moreover, we anticipate the data imbalance issue during joint training since CLIP's training set is 142 times larger than VisDA-2017. We seek a more practical solution for catastrophic forgetting issues without adding extra overheads.

We attempt to fine-tune CLIP on the UDA dataset with a lower learning rate and observe less catastrophic forgetting issues, but the low learning rate prevents CLIP from learning new knowledge. To solve this problem, we propose to adjust the learning rate with Catastrophic Forgetting Measurement (CFM, Fig. 2) by comparing the original representation (forward original image on original CLIP) and fine-tuned representations (forward augmented images on fine-tuned CLIP). CFM is, however, unstable because every image has a different forgetting rate, so we leverage our observation that CLIP is likely to have similar predictions across all augmentations when the training example is easy (and large difference for the hard example). We propose to measure the consistency between weak (translate, flip) and strong augmentation (perturb visual appearance) as a momentum (Dual Consistency Momentum, DCM) to stabilize CFM. Our method does not introduce extra overhead: since the augmentation is already a part of fine-tuning, original prediction can be cached, and we do not need to fine-tune UDA and CLIP datasets jointly.

We further seek to use CLIP with pseudo-labeling on the target domain, which recently enjoyed success in UDA [35,60,75,76,76]. DebiasPL [57] utilized CLIP for pseudo-labeling, but it was designed for a single domain. After extending to source and target domains (UDA setting), DebiasPL [57] suffers from catastrophic forgetting issues (accuracy decreases by 21% on VisDA-2017 after fine-tuning). To solve this problem, we replace the fixed debias factor in DebiasPL [57] with our CFM, and replace the fixed momentum in DebiasPL with our adaptive momentum (DCM). We further include a domain name into a prompt (such as: "This is a [**sketch**] photo of [car]"). Our method mitigates the catastrophic forgetting issue, and achieves the best results on DomainNet [38], VisDA-2017 [39], Office-Home [54], Office-31 [44]. To summarize, our main contributions are:

- We propose to use CLIP in UDA and discover the catastrophic forgetting issue when fine-tuning CLIP. We propose CFM for CLIP in UDA to mitigate this issue without introducing extra computational overhead.
- We propose pseudo-labeling for CLIP in UDA by extending DebiasPL to multiple domains, and replacing debias factor and momentum with our CFM and DCM. We also introduce a domain name into a prompt.
- We achieve the best results on four benchmarks on both ResNet and ViT, with a large performance improvement on the large-scale dataset (+18.5% accuracy on DomainNet).

## 2. Related Works

**Unsupervised Domain Adaptation (UDA)** adapts a model trained on a source domain with annotated datasets to an unlabeled target domain. Discrepancy-based methods [20,31,47,52] learn domain-invariant features via minimizing the discrepancy between source and target domains, or applying adversarial learning [24, 30, 31, 42, 47, 52] to obtain domain-invariant representations. DANN [11] and CDAN [30] use a domain discriminator to classify source and target samples while the feature extractor tries to fool the domain discriminator with generated domain-invariant features. AaD [62] proposes prediction consistency on the neighboring features, while CPR [17] uses prototypes instead. In theory, our improvement is orthogonal to these methods, and they could complement each other. Recent works found that the cross-attention in Vision Transformer (ViT) [9] is advantageous to feature alignment, and it is more robust to the noisy samples compared to CNN, which is the key to the UDA task [60]. Hence recent works [42,49,60,61] use ViT as the backbone and achieve better performance than CNNs. We also leverage ViT.

**Pseudo-labeling in UDA** aims to leverage unlabeled target domain data [26, 29, 35, 49, 60, 76]. Specifically, the model will generate pseudo labels on the unlabeled data during the training process and use them as the supervision in the following training loop. Consistency regularization is also applied to different disturbed views of the same samples [21, 46, 69] to promote the prediction consistency on the unlabeled data. However, these methods are built on the assumption that the unlabeled data share similar distributions as the labeled data [23,26,48,57], which is usually not true since the source and target are in different domains. Such distribution mismatch may generate low-quality and biased pseudo labels [23], resulting in a poor-performance classifier during the self-training process. Therefore, to relieve the noise and bias, we focus on generating less-biased pseudo labels during the UDA training in this work.

**Vision-Language Models** have shown promising results in learning generic visual representations [19, 34, 41, 68]. Recent models gain their advancement via text representation learning with Transformers [53], contrastive representation learning, and web-scale training datasets [74]. For example, CLIP [41] was trained on 400 million image-text pairs and achieved state-of-the-art performance in many fields [41,64,65,67]. However, the best way to adapt CLIP for downstream tasks is still under study. For example, DebiasPL [57] found that CLIP [41] produces imbalanced prediction and proposes to adaptively debias pseudo-label for a single domain. We extend DebiasPL [57] from a single domain to multiple domains in the UDA setting, and address the domain gap problem by mitigating catastrophic forgetting issues, introducing domain name into the prompt, and dynamically adjusting debias strength and momentum.

**Causal Inference** has been introduced in computer vision tasks to alleviate the dataset bias in domain-specific applications [5,8,16,28,40,51,70]. These methods successfully improve performance in many fields such as image classification [1, 32], semantic segmentation [66], visual representation learning [55] and image captioning [63]. Counterfactual inference is a popular method that was used to capture the bias as the direct causal effect [37], eliminate the confounding effect [70], and disentangle the desired direct effect [3]. Our method is built on top of these works to debias pseudo-labels in our setting.

## 3. Methodology

Given a labeled source domain $\mathcal{D}_s = \{(\boldsymbol{x}^s_i, y^s_i)\}^{N_s}_{i=1}$ and an unlabeled target domain $\mathcal{D}_t = \{(\boldsymbol{x}^t_i)\}^{N_t}_{i=1}$, we aim to optimize a model from the labeled source domain to the unlabeled target domain. $N_s$ and $N_t$ denote the size of the source and target datasets respectively.

### 3.1. CLIP in UDA

We first modify CLIP to be suitable for UDA tasks. CLIP [41] is composed of a vision encoder $f$ (maps image into low dimensional image representations), and a text encoder $g$ (maps sentence into text representations). CLIP requires image-text pairs to jointly train $f$ and $g$ with symmetric cross-entropy loss [58] between the image and text representations. We follow prompt engineering [41] to prepare image-text pairs in UDA datasets. Our label $y^s_z$ denotes a sentence in the format of "a [**DOMAIN**] photo of a [CLASS]", where [CLASS] is a classification class name and [**DOMAIN**] is a domain name in UDA tasks (e.g., a synthetic photo of a person). During testing, we follow CLIP zero-shot inference by comparing image representations with the classification weights generated by the text encoder, denoted as $\{\boldsymbol{\theta}_z\}^K_{z=1}$. By forwarding $K$ descriptions corresponding to $K$ classes, we can calculate the probability that a training image belongs to the $k$-th category.

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

where $T$ is the temperature parameter learned by CLIP, cos refers to cosine similarity [41], and we denote a vector of $\ddot{p}_k$ as $p$ (probability of a sample in a minibatch).

### 3.2. Catastrophic Forgetting Measurement

After formulating CLIP for UDA and attempting to finetune CLIP on UDA datasets, we found that CLIP suffers from catastrophic forgetting issues (Fig. 1). We explored baselines in Tab. 1 and found that the original CLIP (no

Table 1: Baseline. We fine-tune CLIP (ResNet-101) on VisDA-2017 source domain training set, and test on VisDA-2017 target domain validation set. "No fine-tuning" is the best baseline because catastrophic forgetting issues are so severe such that any fine-tuning will cause accuracy drops.

| # | Configuration | Accuracy |
|---|---|---|
| 1 | No fine-tuning | 82.3% |
| 2 | Fine-tune 30 epoch | 55.6% |
| 3 | Early stop (fine-tune 1 epoch) | 73.2% |
| 4 | Lower learning rate 50x on #2 | 75.8% |

fine-tuning) performs the best (any fine-tuning will cause performance degradation), but the degradation can be mitigated by lowering the learning rate. We propose to decrease the learning rate according to the difference between the original CLIP ($o$) and fine-tuned CLIP ($f$)'s representations. Large differences indicate that CLIP forgets the pre-trained knowledge (resulting in a new representation). The original CLIP's representation can be cached by running CLIP on the original image (no augmentation) before fine-tuning, and we seek a meaningful augmentation during fine-tuning.

We follow CLSA [56] to introduce "weak" (no appearance change: translate, flip) and "strong" augmentation (perturb appearance: CTAugment [2], RandAugment [6]), and propose to measure the distance between representations from both augmentations. We lower the learning rate when the difference is large because we observe more mistakes from "strong" ($s(x^s)$) than "weak" ($w(x^s)$) augmentation predictions when CLIP struggles with the hard training example (leads to more forgetting). Combining both of our proposals, we formulate a triplet of original CLIP's representations from the original image ($o(x^s)$), fine-tuned CLIP's representations for "weak" ($f(w(x^s))$) and "strong" ($f(s(x^s))$) augmentation. We use Euclidean Distance to measure the similarity of each pair, and we sum all pairs into our triplet distance ($\lambda^s$). We flip the sign to lower the learning rate when the difference of each pair is large.

$$\lambda^s = 1 - \frac{1}{6B}\sum_{i=1}^{B}(||\tilde{f}(w(x_i^s)) - \tilde{o}(x_i^s)||_2 + \\ ||\tilde{f}(s(x_i^s)) - \tilde{o}(x_i^s)||_2 + ||\tilde{f}(w(x_i^s)) - \tilde{f}(s(x_i^s))||_2) \tag{2}$$

where ~ denotes representation normalization with L2 norm (e.g., $\tilde{o}(x^s) = o(x^s)/||o(x^s)||_2$) to cap each representation between [-1, 1]. The summation of all terms is in the range of [0, 6], so we divide the summation by 6 to cap $\lambda^s$ range to [0, 1], and average distances over the batch size ($B$). $\lambda^s$ is, however, unstable because the input ($x^s$) changes every iteration, so the learning rate will constantly be adjusted (resulting in unstable training). To solve this problem, we propose a momentum to slow down the change of $\lambda^s$. We measure the consistency between "weak" and "strong" representations to use as Dual Consistency Momentum (DCM, $m^s$).

Table 2: Our improvements on DebiasPL. We fine-tune DebiasPL (ResNet-101) on VisDA-2017 source domain training set, and test on VisDA-2017 target domain validation set. Our improvement mitigates catastrophic forgetting issues, and improve pseudo-label quality.

| # | Configuration | Accuracy |
|---|---|---|
| 1 | DebiasPL [57] | 64.4% |
| 2 | Add CFM to supervised loss ($\mathcal{L}_{sup}$, Eq. 12) on #1 | 85.9% |
| 3 | Add domain name to prompt on #2 | 86.5% |
| 4 | Add CFM to pseudo-label (Eq. 10) on #3 | 87.8% |
| 5 | Add DCM to pseudo-label (Eq. 9) on #4 | 88.5% |

$$m^s = \frac{1}{B}\sum_{i=1}^{B} cos(f(w(x_i^s)), f(s(x_i^s))) \tag{3}$$

where $cos$ is a cosine similarity. We use low momentum (small $m^s$, slow $\lambda^s$ changes) when the consistency is low (indicating a hard training example). We combine triplet distance with DCM to define Catastrophic Forgetting Measurement (CFM, $\lambda_z^s$) for each iteration ($z$) with $\lambda_0^s = 0$.

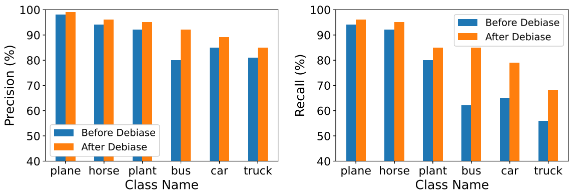
$$\lambda_z^s \leftarrow m^s \lambda_z^s + (1 - m^s)\lambda_{z-1}^s \tag{4}$$

### 3.3. Pseudo-labeling and Inter-class Bias

**Pseudo-labeling in CLIP.** Pseudo-labeling [46, 69] enjoyed success in UDA by leveraging the unlabeled target domain data, but past methods [35, 49, 60, 76] were not designed for CLIP. DebiasPL [57] supports pseudo-labeling for CLIP, but was designed for a single domain. We first follow DebiasPL to generate a soft label on "weak" augmented samples from the target domain as $q = p(y|w(\boldsymbol{x}^t))$, convert a soft label into a hard label by a one-hot encoder ($\mathbb{1}$), and use a fixed threshold ($\tau = 0.4$) to select high confident pseudo-labels. We formulate a consistency loss ($\mathcal{L}_{cp}$) by using a cross-entropy loss ($\mathbf{H}$) to push the prediction from "strong" augmentation ($\boldsymbol{p}(y|s(\boldsymbol{x}^t))$) to be closed to pseudo-label from "weak" augmentation.

$$\mathcal{L}_{cp} = \frac{1}{B}\sum_{i=1}^{B} \mathbb{1}[\max(q_i) \geq \tau] \cdot \mathbf{H}(p(y_i|s(\boldsymbol{x}_i^t)), q_i) \tag{5}$$

**Inter-class Bias in Pseudo-labeling** is caused by the reliance on a trained model to generate pseudo-labels. If multiple classes have similar appearances (e.g., dog vs. wolf), the model tends to have prediction errors, which will generate incorrect pseudo-labels. As the training goes on, these incorrect pseudo-labels will further increase the existing bias and eventually lead to a significant accuracy drop. Past



(a) Precision Score (b) Recall Score

Figure 3: Debiased Pseudo-labeling. Inter-class bias degrades pseudo-label quality and reduces the recall of classes with similar appearance (e.g., bus, car, truck). Our debiased method can mitigate such issues and improve recall.

pseudo-labeling debiased methods [23, 57] perform reasonably well in a generic setting, but the bias in the UDA setting is more severe due to the domain gap between source and target domains. We analyze the pseudo-label quality in Fig. 3 and observe the co-existence effect [50] where many samples in the confusing classes tend to be misclassified into other similar classes (e.g., "bus", "car", and "truck" are all belong to "vehicle"). We seek to simultaneously mitigate inter-class bias while combating catastrophic forgetting.

### 3.4. Pseudo-labeling with Adaptive Debiasing

**Causal Inference.** We follow DebiasPL [57] to use causal inference to mitigate inter-class bias in pseudo-labels. Given the causal graph in Fig. 4, debiasing of predictions can be delineated as the direct causal effect along $x_i^t \rightarrow p$, defined as Controlled Direct Effect (CDE) [43, 50, 57].

$$\text{CDE}(p_i) = [p_i | do(x_i^t), do(D)] - [p_i | do(\hat{x}^t), do(D)] \quad (6)$$

where $do(\cdot)$ denotes the causal intervention [13] that removes the model bias ($M$) from $x^t$, and $\hat{x}^t = \{x_1^t, ..., x_n^t\}$. It is, however, computationally expensive to visit all training samples to measure the counterfactual outcome.

**Debiasing by DebiasPL [57].** We follow DebiasPL [57] to use Approximated Controlled Direct Effect (ACDE) by assuming the model bias is not drastically changed. This assumption holds true in our settings as we aim to fine-tune CLIP and keep the original knowledge. We approximate the first term ($[p_i | do(x_i^t), do(M)]$) in Eq. 6 as

$$p_i' \leftarrow \ddot{m} p_i' + (1 - \ddot{m}) \frac{1}{B} \sum_{i=1}^{B} p_i \quad (7)$$

where $p_i$ is the vector of prediction from Eq. 1, $p'$ is the debiased prediction, and $\ddot{m}$ is a fixed momentum. Then the debiased pseudo-label can be formulated as

$$q_i' = q_i - \mu \log p_i' \quad (8)$$

where $\mu$ is a fixed debias factor, $q$ is a soft label from Eq. 5, and $q'$ is the debiased soft label.

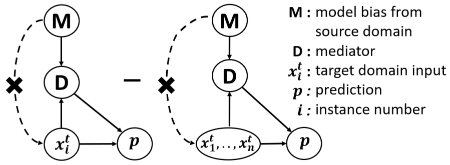


Figure 4: Causal graph via counterfactual reasoning for debiasing model predictions for pseudo-labels.

**Debiasing by CFM.** DebiasPL [57] was designed for a single domain (DebiasPL's setting splits a single dataset (such as CIFAR-10) into labeled (source) and unlabelled (target) data). Hence, it suffers from severe bias due to the domain gap in UDA (Tab. 2). We identify the debias factor ($\mu$, Eq. 8) as one of the root cause because $\mu$ is sensitive (small $\mu$ does not eliminate bias, and large $\mu$ prevents the model from learning new knowledge [50, 57]). Moreover, $\mu$ is set to a fixed value, but the bias is dependent on the domain setting (e.g., real-world vs. synthetic will likely have a higher bias than sketch vs. quick draw), so we propose to adaptively adjust $\mu$. We incorporate catastrophic forgetting information by replacing $\mu$ with CFM ($\lambda_z^t$) to adjust $\mu$ adaptively, and replace the fix momentum ($\ddot{m}$, Eq. 7) with DCM ($m^t$) to also adjust $\ddot{m}$ adaptively. Both CFM and DCM for debiasing are computed on target domain input ($x^t$).

$$p_i' \leftarrow m^t p_i' + (1 - m^t) \frac{1}{B} \sum_{i=1}^{B} p_i \quad (9)$$

$$q_i' = q_i - \lambda_z^t \log p_i' \quad (10)$$

Finally, the debiased consistency loss ($\mathcal{L}_{dcp}$) is formulated by replacing $q$ with $q'$ on the consistency loss (Eq. 5).

$$\mathcal{L}_{dcp} = \frac{1}{B} \sum_{i=1}^{B} \mathbb{1}[\max(q_i') \geq \tau] \cdot \mathbf{H}(p(y_i | s(\boldsymbol{x}_i^t)), q_i') \quad (11)$$

### 3.5. End-to-end Trainable Pipeline

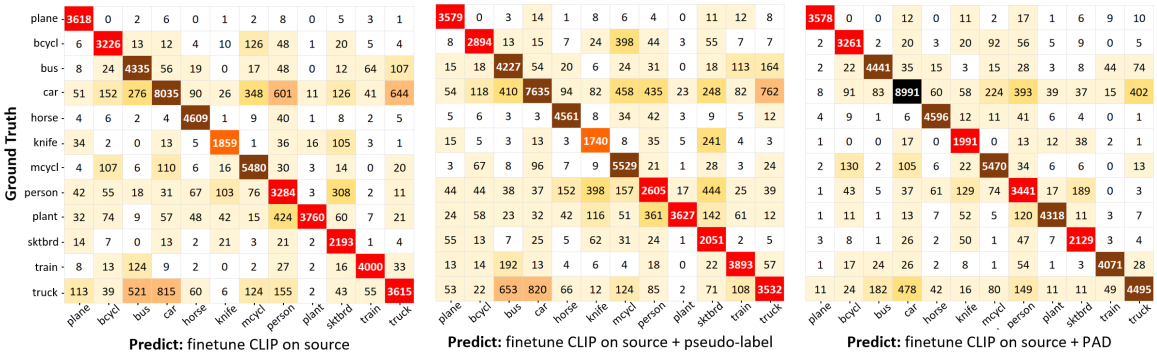
Fig. 2 shows the pipeline. We use symmetric contrastive loss [41] ($\mathcal{L}_{sup}$) and debiased consistency loss ($\mathcal{L}_{dcp}$).

$$\mathcal{L} = \lambda_z^s \mathcal{L}_{sup}(\mathcal{D}_s) + \Lambda \mathcal{L}_{dcp}(\mathcal{D}_t) \quad (12)$$

where $\Lambda = 0.5$ is a constant term for adjusting $\mathcal{L}_{dcp}$. We use CFM ($\lambda_z^t$, Eq. 2) to adjust $\mathcal{L}_{sup}$ to mitigate catastrophic forgetting issues. Our loss formulation allows end-to-end training to simultaneously preserve CLIP knowledge, supervised fine-tuning on the source domain, and fine-tuning with debias pseudo-labels on the target domain.

## 4. Experimental Setup

**Dataset.** We evaluate our proposed methods on four popular UDA datasets. **VisDA-2017 [39]** contains 152k synthetic images and 55k real object images of 12 categories



**Predict:** finetune CLIP on source

| | plane | bcycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| plane | 3618 | 0 | 2 | 6 | 0 | 1 | 2 | 8 | 0 | 5 | 3 | 1 |
| bcycl | 6 | 3226 | 13 | 12 | 4 | 10 | 126 | 48 | 1 | 20 | 5 | 4 |
| bus | 8 | 24 | 4335 | 56 | 19 | 0 | 17 | 48 | 0 | 12 | 64 | 107 |
| car | 51 | 152 | 276 | 8035 | 90 | 26 | 348 | 601 | 11 | 126 | 41 | 644 |
| horse | 4 | 6 | 2 | 4 | 4609 | 1 | 9 | 40 | 1 | 8 | 2 | 5 |
| knife | 34 | 2 | 0 | 13 | 5 | 1859 | 1 | 36 | 16 | 105 | 3 | 1 |
| mcycl | 4 | 107 | 6 | 110 | 6 | 16 | 5480 | 30 | 3 | 14 | 0 | 20 |
| person | 42 | 55 | 18 | 31 | 67 | 103 | 76 | 3284 | 3 | 308 | 2 | 11 |
| plant | 32 | 74 | 9 | 57 | 48 | 42 | 15 | 424 | 3760 | 60 | 7 | 21 |
| sktbrd | 14 | 7 | 0 | 13 | 2 | 21 | 3 | 21 | 2 | 2193 | 1 | 4 |
| train | 8 | 13 | 124 | 9 | 2 | 0 | 2 | 27 | 2 | 16 | 4000 | 33 |
| truck | 113 | 39 | 521 | 815 | 60 | 6 | 124 | 155 | 2 | 43 | 55 | 3615 |

**Predict:** finetune CLIP on source + pseudo-label

| | plane | bcycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| plane | 3579 | 0 | 3 | 14 | 1 | 8 | 6 | 4 | 0 | 11 | 12 | 8 |
| bcycl | 8 | 2894 | 13 | 15 | 7 | 24 | 398 | 44 | 3 | 55 | 7 | 7 |
| bus | 15 | 18 | 4227 | 54 | 20 | 6 | 24 | 31 | 0 | 18 | 113 | 164 |
| car | 54 | 118 | 410 | 7635 | 94 | 82 | 458 | 435 | 23 | 248 | 82 | 762 |
| horse | 5 | 6 | 3 | 3 | 4561 | 8 | 34 | 42 | 3 | 9 | 5 | 12 |
| knife | 15 | 5 | 3 | 13 | 3 | 1740 | 8 | 35 | 5 | 241 | 4 | 3 |
| mcycl | 3 | 67 | 8 | 96 | 7 | 9 | 5529 | 21 | 1 | 28 | 3 | 24 |
| person | 44 | 44 | 38 | 37 | 152 | 398 | 157 | 2605 | 17 | 444 | 25 | 39 |
| plant | 24 | 58 | 23 | 32 | 42 | 116 | 51 | 361 | 3627 | 142 | 61 | 12 |
| sktbrd | 55 | 13 | 7 | 25 | 5 | 62 | 31 | 24 | 1 | 2051 | 2 | 5 |
| train | 13 | 14 | 192 | 13 | 4 | 6 | 4 | 18 | 0 | 22 | 3893 | 57 |
| truck | 53 | 22 | 653 | 820 | 66 | 12 | 124 | 85 | 2 | 71 | 108 | 3532 |

**Predict:** finetune CLIP on source + PAD

| | plane | bcycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| plane | 3578 | 0 | 0 | 12 | 0 | 11 | 2 | 17 | 1 | 6 | 9 | 10 |
| bcycl | 2 | 3261 | 2 | 20 | 3 | 20 | 92 | 56 | 5 | 9 | 0 | 5 |
| bus | 2 | 22 | 4441 | 35 | 15 | 3 | 15 | 28 | 3 | 8 | 44 | 74 |
| car | 8 | 91 | 83 | 8991 | 60 | 58 | 224 | 393 | 39 | 37 | 15 | 402 |
| horse | 4 | 9 | 1 | 6 | 4596 | 12 | 11 | 41 | 6 | 4 | 0 | 1 |
| knife | 1 | 0 | 0 | 17 | 0 | 1991 | 0 | 13 | 12 | 38 | 2 | 1 |
| mcycl | 2 | 130 | 2 | 105 | 6 | 22 | 5470 | 34 | 6 | 6 | 0 | 13 |
| person | 1 | 43 | 5 | 37 | 61 | 129 | 74 | 3441 | 17 | 189 | 0 | 3 |
| plant | 1 | 11 | 1 | 13 | 7 | 52 | 5 | 120 | 4318 | 11 | 3 | 7 |
| sktbrd | 3 | 8 | 1 | 26 | 2 | 50 | 1 | 47 | 7 | 2129 | 3 | 4 |
| train | 1 | 17 | 24 | 26 | 2 | 8 | 1 | 54 | 1 | 3 | 4071 | 28 |
| truck | 11 | 24 | 182 | 478 | 42 | 16 | 80 | 149 | 11 | 11 | 49 | 4495 |

Figure 5: The confusion matrix on VisDA-2017 shows the effect of pseudo-label and the pseudo-label with adaptive debiasing (PAD). Diagonal values are true positive (darker = better) and other values are errors (brighter = better).

Table 3: Accuracies (%) on **VisDA-2017**. "-B" indicates ViT-B (except CDTrans uses DeiT). See full table in Appendix.

| Method | plane | bcycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RN-101 [14] | 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 |
| SDAT [42] | 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 |
| CAN [20] | 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 |
| Ours (RN-101) | 96.7 | 88.8 | 87.0 | 82.8 | 97.1 | 93.0 | 91.3 | 83.0 | 95.5 | 91.8 | 91.5 | 63.0 | **88.5** |
| ViT-B [9] | 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 |
| 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 [49] | 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 [42] | 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 |
| Ours-B | 98.1 | 93.8 | 87.1 | 85.5 | 98.0 | 96.0 | 94.4 | 86.0 | 94.9 | 93.3 | 93.5 | 70.2 | **90.9** |

Table 4: Accuracies (%) on **Office-Home**. "-B" indicates ViT-B (except CDTrans uses DeiT). 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 [14] | 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 [42] | 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 |
| 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 [48] | 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 |
| Ours (RN-50) | 57.5 | 84.0 | 83.8 | 77.8 | 85.5 | 84.7 | 76.3 | 59.2 | 85.4 | 78.1 | 60.2 | 86.7 | **76.6** |
| ViT-B [9] | 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 [42] | 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 [49] | 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 |
| Ours-B | 76.4 | 90.6 | 90.8 | 86.7 | 92.3 | 92.0 | 86.0 | 74.5 | 91.5 | 86.9 | 79.1 | 93.1 | **86.7** |

sampled from Microsoft COCO. **Office-Home [54]** has 15.5k images of 65 categories from 4 domains: Art (Ar), Clipart (Cl), Product (Pr), and Real-world (Rw). **Office-31 [44]** includes 31 classes and 4.6k images from 3 domains: Amazon (A), DSLR (D), and Webcam (W). **DomainNet [38]** is the most challenging dataset that contains 0.6 million images of 345 classes from 6 domains: Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr), Real-world (rel), and Sketch (skt). We strictly follow the protocol of previous works [42, 49, 60]. On VisDA-2017, we use synthetic as the source domain and the real object as the target domain; on the other three datasets, we select one domain as the source and another domain as the target. We have 12, 6, and 30 source-target combinations on Office-Home, Office-31, and DomainNet, respectively.

**Training Configuration.** We experiment on both ViT-B [9] (patch size 16 × 16, batch size 16) and ResNet [14] (batch size 32) as the vision encoder in CLIP [41]. The learning rate is set to $1e^{-6}$ on all datasets, except $1e^{-7}$ on VisDA-2017 because training on VisDA-2017's synthetic data is not stable and the training may diverge. We freeze the text encoder and only train the vision encoder in the CLIP framework. We follow the training process in CLIP to use Adam optimizer with decoupled weight decay regularization [36] incorporated into all weights that are not gains or biases. Cosine schedule [33] is used to decay the learning rate and we train every setting for 30 epochs.

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). See full table in Appendix.

| **MDD+ SCDA** [25] | 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** [9] | 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 |

| **SDAT -B** [42] | clp | inf | pnt | qdr | rel | skt | Avg. |
|---|---|---|---|---|---|---|---|
| clp | - | 22.0 | 41.5 | - | 57.5 | 47.2 | 42.1 |
| inf | 33.9 | - | 30.3 | - | 48.1 | 27.9 | 35.0 |
| pnt | 47.5 | 20.7 | - | - | 58.0 | 41.8 | 42.0 |
| qdr | - | - | - | - | - | - | - |
| rel | 56.7 | 25.1 | 53.6 | - | - | 43.9 | 44.8 |
| skt | 58.7 | 21.8 | 48.1 | - | 57.1 | - | 46.4 |
| Avg. | 49.2 | 22.4 | 43.4 | - | 55.2 | 40.2 | 42.1 |

| **SSRT -B** [49] | 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 | - | 73.6 | 75.4 | 74.6 | 76.4 | 76.3 | 75.3 |
| inf | 55.1 | - | 54.3 | 53.6 | 54.9 | 54.9 | 54.6 |
| pnt | 71.1 | 70.6 | - | 70.0 | 72.7 | 71.7 | 71.2 |
| qdr | 36.8 | 18.0 | 32.0 | - | 31.7 | 34.9 | 30.7 |
| rel | 84.2 | 83.5 | 83.5 | 83.1 | - | 83.6 | 83.6 |
| skt | 68.1 | 66.6 | 67.2 | 66.1 | 67.5 | - | 67.1 |
| Avg. | 63.1 | 62.5 | 62.5 | 69.5 | 60.6 | 64.3 | **63.7** |

Table 6: Accuracies (%) on **Office-31**.

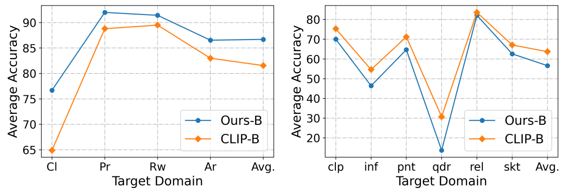
| Method | A→W | D→W | W→D | A→D | D→A | W→A | Avg. |
|---|---|---|---|---|---|---|---|
| ViT-B [9] | 91.2 | 99.2 | 100. | 90.4 | 81.1 | 80.6 | 90.4 |
| SHOT-B [27] | 94.3 | 99.0 | 100. | 95.3 | 79.4 | 80.2 | 91.4 |
| CDTrans [60] | 96.7 | 99.0 | 100. | 97.0 | 81.1 | 81.9 | 92.6 |
| SSRT-B [49] | 97.7 | 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 | 97.9 | 99.2 | 100. | 98.5 | 84.6 | 85.3 | 94.3 |

## 5. Results

### 5.1. Ablation studies

**Importance of Pretrained Data.** Since CLIP is data-hungry [19,34,41,68], we first study the sensitivity of CLIP in the UDA setting. We compare public models from CLIP [41] (trained on 400 million image-text pairs) with Open-CLIP [18] (trained on Conceptual Captions dataset [45], 3 million image-text pairs). Tab. 7 shows the accuracy drops from 82% to 59% when the pre-trained dataset is reduced. Therefore, although CLIP has a strong zero-shot generalization, such capability is learned through large-scale data. In contrast, a small dataset is likely to lack enough samples to disentangle the domain and class. We conclude that a large pre-trained dataset is important, but we keep both datasets throughout our ablation studies to show that our improvement holds true even with small pre-trained data.

**Importance of CFM.** CLIP suffers from catastrophic forgetting issues (Fig. 1), so we use Catastrophic Forgetting Measurement (CFM) to compare representations from the fine-tuned CLIP against the original CLIP. CFM adjusts the learning rate to slow down the forgetting process while accumulating new knowledge from the UDA dataset (Fig. 7). Tab. 7 shows that CFM can recover from -26% accuracy drops to improve accuracy by +1.6% (row: 6-8). Fig. 6 further shows that the improvement from CFM is consistent across all classes on multiple datasets.



(a) OfficeHome [54]

(b) DomainNet [38]

Figure 6: Our improvement compared to the original CLIP [41]: the score is averaged over all tasks using each domain as the target domain. "-B" refers to ViT-B [9].

**Importance of Adaptive Debiasing.** Pseudo-labeling improves our method by +2% (Tab. 7, row: 8,9), and we enhance the pseudo-labeling process with adaptive debiasing (PAD) by our CFM and DCM on top of DebiasPL [57] (row: 9, 10). To further verify the effectiveness of our debias method, we summarize the confusion matrix in Fig. 5 and compute precision/recall in Fig. 3 to show improvement in inter-class confusion ("car", "bus" and "truck" belong to "vehicle" category).

### 5.2. External Comparison

We achieved state-of-the-art results on four public datasets using ViT-B backbone, and we test multiple convolutional backbones for a fair comparison.

**VisDA-2017.** We first use ResNet-101 (RN-101) as the baseline model to perform fair comparisons with recent methods [4, 10, 15, 42, 59, 71]. Tab. 3 shows that our method consistently improves almost all classes, and improves 4.2% on the average accuracy compared to the previous best method, SDAT [42]. We then follow ViT-based methods [42, 49, 60] to use ViT-B (denote as "-B") and achieve superior performance compared to the state-of-the-art methods. We also observe a significant accuracy in-

Table 7: Ablation study. We fine-tune CLIP (ResNet-101) on VisDA-2017 source domain training set, and test on VisDA-2017 target domain validation set. CLIP, trained on 400M, ourperforms 3M (row: 1,6). Finetune CLIP on a source domain leads to catastrophic forgetting and CFM can mitigate it (row: 6-8). Pseudo-label (PL) with Adaptive Debiasing (PAD) further improves accuracy (row: 8-10).

| # | Pretrain | CLIP | Source | CFM | PL | PAD | Accuracy |
|---|---|---|---|---|---|---|---|
| 1 | 3 M | ✓ | ✗ | ✗ | ✗ | ✗ | 59.1% |
| 2 | 3 M | ✓ | ✓ | ✗ | ✗ | ✗ | 44.3% |
| 3 | 3 M | ✓ | ✓ | ✓ | ✗ | ✗ | 62.9% |
| 4 | 3 M | ✓ | ✓ | ✓ | ✓ | ✗ | 65.7% |
| 5 | 3 M | ✓ | ✓ | ✓ | ✓ | ✓ | 67.1% |
| 6 | 400 M | ✓ | ✗ | ✗ | ✗ | ✗ | 82.3% |
| 7 | 400 M | ✓ | ✓ | ✗ | ✗ | ✗ | 55.6% |
| 8 | 400 M | ✓ | ✓ | ✓ | ✗ | ✗ | 83.9% |
| 9 | 400 M | ✓ | ✓ | ✓ | ✓ | ✗ | 86.0% |
| 10 | 400 M | ✓ | ✓ | ✓ | ✓ | ✓ | 88.5% |

crease on "truck", thanks to our adaptive debiasing module that makes "truck" more discriminate from "car".

**Office-Home/31.** For Office-Home [54], we first use ResNet-50 (RN-50) to fairly compare with recent methods [4, 25, 42, 59]. Tab. 4 shows a +2.7% increase from the previous best method, KUDA [48]. We then follow ViT-based methods [42, 49, 60, 61] to use ViT-B, and observe consistent improvement across almost all settings. For office-31 [44], we have a similar observation on consistent improvement compared to the recent methods (Tab. 6).

**DomainNet.** Previous improvement (SSRT [49] vs SDAT [42]) only achieves +3.1% accuracy on DomainNet [38] because this is the largest dataset, several domains have completely different appearances (e.g., infographic vs quickdraw), and the distributions among different domains are imbalanced. Our method, however, achieves +18.5% improvement over the previous best method, SSRT, thanks to our proposed CFM for preserving the original CLIP's pretrained knowledge, and our pseudo-label with adaptive debiasing for improving the pseudo-label quality.

### 5.3. Generalization of Our Method

**Applications.** We observe catastrophic forgetting issue in CLIP across multiple applications where the accuracy decreases when fine-tuning CLIP in Incremental Learning (-9.9%), and Domain Generalization (-14.0%) and CFM can increase the accuracy +5.3%, +2.4% respectively. For few-shot learning, our method outperforms Tip-Adapter-F [72] by 0.5%. Details are in Appendix.

**Vision backbone.** Our method works on both vision language (CLIP), and vision model. With our method, BiT-M-R101x3 [22] (trained on JFT-300M) achieved 88.1% on VisDA-2017 (comparable to CLIP (90.9%, Tab. 3)). However, a small pre-trained dataset is a limitation, as our method has no effect on ResNet-101 with ImageNet-1K due to the minimal effect of catastrophic forgetting issues.

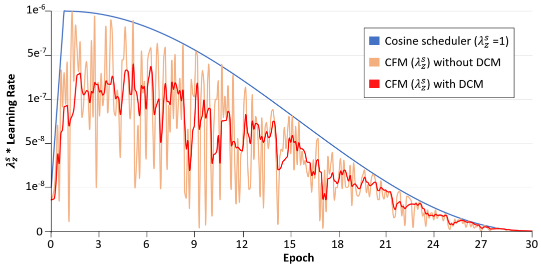


Figure 7: We use cosine scheduler with base learning rate (base $LR$) $1e^{-6}$ (except VISDA 2017 ($1e^{-7}$)) and warm up for 1 epoch. CFM ($\lambda_z^s$) is the loss weight for the source domain ($L_{sup}$) and CFM will adjust the gradient together with the learning rate ($\lambda_z^s * LR$). Low CFM (high catastrophic forgetting) will decrease the gradient, and we further smooth CFM with momentum (DCM).

### 5.4. Computational Complexity

It takes 16.5 hours to train CLIP with ViT-B backbone on an Nvidia Tesla V100 GPU for VisDA-2017. Pseudo-label is a standard method in UDA [35, 49, 60, 76] and adding pseudo-label increases the training times to 23.3 hours. To compute CFM, we forward the entire dataset with the original CLIP (0.6 hours for VisDA-2017) to cache the original CLIP representation (only needs to do once). We did not observe any overhead from CFM during training since CFM simply compare the low dimensional representations (obtain as part of the training). We observe a training time increase to 23.5 hours from our adaptive debiasing, but the overhead is trivial with <1% increases from the pseudo-label setting. Our method does not change the test speed.

## 6. Conclusion

We propose CLIP in the UDA setting. We first include a domain name into a prompt, and we uncover catastrophic forgetting issues when fine-tuning CLIP. We propose to counter this by adjusting the learning rate according to CFM. We add pseudo-labeling by further extending DebiasPL (from a single domain to multiple domains in the UDA setting) with our CFM and DCM to better adjust debias strength and momentum. Our method does not introduce computational overhead, and achieves superior results than the state-of-the-art methods on four public datasets, with a large improvement (+18.5%) on DomainNet. For future work, CFM and DCM could be improved to a more sophisticated function or even learnable.

## References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2015. 3

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020. 4

[3] Michel Besserve, Arash Mehrjou, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. In *ICLR*, 2019. 3

[4] 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, 7, 8

[5] Zhang Chen, Zhiqiang Tian, Jihua Zhu, Ce Li, and Shaoyi Du. C-cam: Causal cam for weakly supervised semantic segmentation on medical image. In *CVPR*, pages 11676–11685, 2022. 3

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 702–703, 2020. 4

[7] 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. 1

[8] Zhang Dong, Zhang Hanwang, Tang Jinhui, Hua Xiansheng, and Sun Qianru. Causal intervention for weakly supervised semantic segmentation. In *NeurIPS*, 2020. 3

[9] 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, 3, 6, 7

[10] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *CVPR*, pages 3937–3946, 2021. 7

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015. 1, 3

[12] 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

[13] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. 5

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

[15] 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. 7

[16] Jianqiang Huang, Yu Qin, Jiaxin Qi, Qianru Sun, and Hanwang Zhang. Deconfounded visual grounding. In *AAAI*, pages 998–1006, 2022. 3

[17] Sungsu Hur, Inkyu Shin, Kwanyong Park, Sanghyun Woo, and In So Kweon. Learning classifiers of prototypes and reciprocal points for universal domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 531–540, 2023. 3

[18] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below. 7

[19] 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. 3, 7

[20] 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. 3, 6

[21] Jiwon Kim, Kwangrok Ryoo, Junyoung Seo, Gyuseong Lee, Daehwan Kim, Hansang Cho, and Seungryong Kim. Semi-supervised learning of semantic correspondence with pseudo-labels. In *CVPR*, pages 19699–19709, 2022. 3

[22] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning, 2019. 8

[23] Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching S Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *ICML*, pages 11828–11843, 2022. 3, 5

[24] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 91–100, 2019. 3

[25] 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, 7, 8

[26] 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. 3

[27] 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, 7

[28] Bing Liu, Dong Wang, Xu Yang, Yong Zhou, Rui Yao, Zhiwen Shao, and Jiaqi Zhao. Show, deconfound and tell: Image

captioning with causal inference. In *CVPR*, pages 18041–18050, 2022. 3

[29] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021. 3

[30] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1645–1655, 2018. 3

[31] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017. 3

[32] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Leon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[33] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[34] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, pages 5206–5215, 2022. 3, 7

[35] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, pages 415–430, 2020. 2, 3, 4, 8

[36] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *ICLR*, 2018. 6

[37] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *CVPR*, pages 12700–12710, 2021. 3

[38] 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, 8

[39] 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. 1, 2, 5

[40] Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *CVPR*, pages 10860–10869, 2020. 3

[41] 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. 1, 3, 5, 6, 7

[42] 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. 1, 3, 6, 7, 8

[43] Lorenzo Richiardi, Rino Bellocco, and Daniela Zugna. Mediation analysis in epidemiology: methods, interpretation and bias. *International journal of epidemiology*, 42(5):1511–1519, 2013. 5

[44] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010. 2, 6, 8

[45] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. 7

[46] 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. 3, 4

[47] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pages 443–450, 2016. 3

[48] 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. 3, 6, 8

[49] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *CVPR*, pages 7191–7200, 2022. 1, 3, 4, 6, 7, 8

[50] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *NeurIPS*, 33:1513–1524, 2020. 5

[51] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3716–3725, 2020. 3

[52] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 3

[53] 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. 3

[54] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 2, 6, 7, 8

[55] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[56] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *arXiv preprint arXiv:2104.07713*, 2021. 4

[57] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *CVPR*, pages 14647–14657, 2022. 2, 3, 4, 5, 7

[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. 7, 8

[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, 3, 4, 6, 7, 8

[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. 3, 6, 7, 8

[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. 3, 6

[63] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect, 2020. 3

[64] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. 1, 3

[65] 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. 1, 3

[66] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2734–2746. Curran Associates, Inc., 2020. 3

[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. 1, 3

[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. 3, 7

[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. 3, 4

[70] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *NeurIPS*, 33:655–666, 2020. 3

[71] Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. Spectral unsupervised domain adaptation for visual recognition. In *CVPR*, pages 9829–9840, 2022. 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. 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] 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. 3

[75] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018. 2

[76] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, pages 5982–5991, 2019. 2, 3, 4, 8