Split to Merge: Unifying Separated Modalities for Unsupervised Domain Adaptation

Xinyao Li[1]   Yuke Li[2*]   Zhekai Du[1]   Fengling Li[3]   Ke Lu[1]   Jingjing Li[1*]

[1]University of Electronic Science and Technology of China
[2]Boston College
[3]University of Technology Sydney

xinyao326@outlook.com, lidwh@bc.edu, zhekaid@std.uestc.edu.cn
fenglingli2023@gmail.com, kel@uestc.edu.cn, lijin117@yeah.net

Abstract

Large vision-language models (VLMs) like CLIP have demonstrated good zero-shot learning performance in the unsupervised domain adaptation task. Yet, most transfer approaches for VLMs focus on either the language or visual branches, overlooking the nuanced interplay between both modalities. In this work, we introduce a Unified Modality Separation (UniMoS) framework for unsupervised domain adaptation. Leveraging insights from modality gap studies, we craft a nimble modality separation network that distinctly disentangles CLIP's features into language-associated and vision-associated components. Our proposed Modality-Ensemble Training (MET) method fosters the exchange of modality-agnostic information while maintaining modality-specific nuances. We align features across domains using a modality discriminator. Comprehensive evaluations on three benchmarks reveal our approach sets a new state-of-the-art with minimal computational costs. Code: https://github.com/TL-UESTC/UniMoS.

1. Introduction

Unsupervised domain adaptation (UDA) [2, 7, 29, 30] aims to apply knowledge trained on a source domain to an unlabeled target domain, a process invaluable in data-scarce scenarios. Conventional methods often struggle with bridging the gap between source and target domains, finding it challenging to develop consistent features across domains [19, 39]. In image classification, aligning vision features while neglecting semantic content can lead to difficulties in differentiating complex samples [10, 41]. Vision-language models (VLMs) such as CLIP [36] and ALIGN [13] circumvent these issues through joint multimodal pretraining on

Modality / Prob.	couch	chair (√)
image modality	0.002	0.998
text modality	0.832	0.130

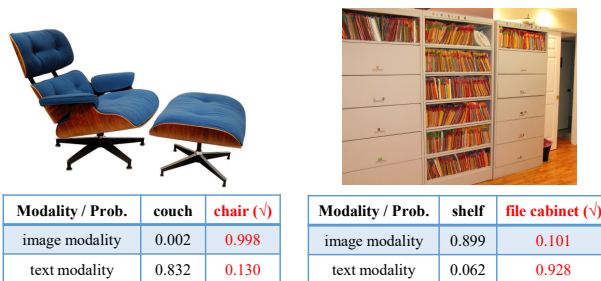| Modality / Prob.	shelf	file cabinet (√)
image modality	0.899	0.101
text modality	0.062	0.928

Figure 1. Examples of modality-specific information from task Art→RealWorld in Office-Home dataset. The digits are top-2 highest classification probabilities given by both modalities.

images and texts. This extensive pretraining endows publicly available VLMs with robust zero-shot transfer abilities and a broad base of conceptual knowledge, making them highly suitable for comprehensive UDA. They facilitate alignment across both visual and textual modalities, enhancing adaptability and applicability in diverse contexts.

Previous studies have shown promising results by adapting VLMs like CLIP for unsupervised domain adaptation (UDA). For instance, DAPrompt [10] introduces learning both domain-agnostic and domain-specific text embeddings, while PADCLIP [17] focuses on fine-tuning the vision branch of CLIP for adaptive visual feature extraction. However, recent research [14, 24] highlights a *modality gap* in VLMs, revealing that, despite training efforts, vision and text features often remain distinctly distributed. We argue that adapting a single modality is less than ideal due to the existence of unique, modality-specific cues in misaligned textual and visual components. We suggest that certain samples are best classified using specific modalities, a hypothesis supported by empirical observations in Fig. 1. This figure shows differing classification patterns when each modality is adapted independently to the unlabeled target data. Text modality results derive from CLIP's

---

*Corresponding author.

zero-shot capabilities, while image results come from linear probing with target pseudo-labels. For instance, visually straightforward items like a cushioned chair are accurately classified by the vision linear classifier after tuning on target dataset. However, pretrained CLIP can erroneously categorize such items under visually similar classes. In contrast, complex items with nuanced semantic details, like a file cabinet resembling a shelf, may confuse the vision classifier, while CLIP's broader knowledge base facilitates correct zero-shot predictions. In summary, while the vision branch effectively discerns class-specific visual patterns, the text branch leverages semantic information to clarify ambiguous cases. This observation lays the groundwork for a multimodal adaptation framework that synergistically combines the strengths of both modalities.

A direct approach to domain adaptation involves concurrently fine-tuning vision branch and crafting textual prompts, which risks disturbing the image-text representation pairs in pretrained CLIP and is computationally intensive [9, 57]. As a more efficient alternative, we propose to explicitly disentangle CLIP-extracted visual features into two complementary parts. The first component retains the language-associated semantic knowledge inherent in CLIP, while the second focuses on vision-specific attributes crucial for distinguishing between nuanced visual categories.

We devised a set of modality separation networks with dual branches to project CLIP-encoded visual features into distinct language-associated components (LAC) and vision-associated components (VAC). An orthogonal regularization is employed to ensure these branches yield discrete, disentangled representations. Each component is optimized based on its inherent modality strengths. For the LAC branch, we utilize knowledge distillation on target data to harness the rich semantic content from the original pretrained CLIP model. Additionally, we implement a debiasing method to mitigate dataset bias in CLIP's zero-shot results. For the VAC branch, the locality structure within visual feature spaces [20, 53, 54] is leveraged to generate visual pseudo-labels for supervised learning on target data. We then introduce a novel Modality-Ensemble Training (MET) strategy that synergistically merges outputs from both modalities. A weight generator dynamically assembles these predictions, supervised by VAC pseudo-labels on target data and actual labels on source data. Importantly, the text modality output remains isolated during MET to preserve independent training and maintain pretrained semantics. Additionally, a modality discriminator is utilized to align LAC and VAC across domains for unsupervised domain adaptation. Trained on source data to distinguish between LAC and VAC, this discriminator is frozen on the target domain, directly updating the separation networks to produce domain-invariant LAC and VAC. This approach ensures a consistent modality separation across domains, facilitating simultaneous adaptation in both modalities.

Contributions: 1. We investigate the modality gap phenomenon in the context of applying Vision-Language Models (VLMs) to unsupervised domain adaptation, revealing the limitations of adapting a single modality; 2. We introduce a novel framework, Unified Modality Separation (UniMoS), which, coupled with a Modality-Ensemble Training (MET) approach, facilitates effective multimodal adaptation; 3. Our comprehensive analysis and validations underscore and efficiency of the proposed UniMoS, demonstrating its ability to set new state-of-the-art benchmarks while maintaining low computational demands.

2. Related work

Unsupervised domain adaptation (UDA). A core challenge in UDA is aligning representations between the source domain and unlabeled target domain. Prior techniques can be categorized as discrepancy-based [18, 28, 59] and adversarial methods [7, 30, 39]. Discrepancy-based methods explicitly minimizes divergence metrics including MMD [18], MDD [59], etc. Adversarial methods extract domain invariant features via a min-max game between the feature extractor and domain discriminator [7, 30]. Recent works focus on exploiting target data structures via self-training techniques [15, 53–55, 60]. ICON [55] learns an invariant classifier with consistent predictions to remove the spurious correlation inconsistency in the target domain. EIDCo [60] combines Mixup [56] with IDCo loss [5, 12] to explore target data distribution. Vision transformer (ViT) [6] and its variants have also gained popularity due to their superior performance [51, 52, 63]. PMTrans [63] mixes patch representations in SwinTransformer [27] as an intermediate domain bridge. CDTrans [51] aligns features extracted by DeiT [44] via cross-attention.

Vision-language models (VLMs) have shown great generalization abilities due to extensive multimodal pretraining [13, 36, 47, 48]. CLIP [36] is trained from 400 million text-image pairs, while ALIGN [13] leverages more than one billion text-image pairs. Subsequent works have built on pretrained VLMs in various ways. Some learn prompt texts to transfer VLMs to downstream tasks [10, 33, 38, 61, 62], while others incorporate additional tunable layers on the frozen pretrained encoder [9, 57]. Beyond utilizing existing VLMs, research also aims to improve VLM training [14, 31]. Liang *et al.* [24] reveal that VLMs exhibit a modality gap, failing to perfectly align multimodal features. Jiang *et al.* [14] conduct theoretical analysis on modality gap and propose latent space regularization to preserve modality-specific information. MaPLe [16] utilize prompt learning on both modality branches to improve alignment. Our approach is fundamentally different since we disentangle VLM-extracted features posteriorly instead of training VLM from scratch, requiring far
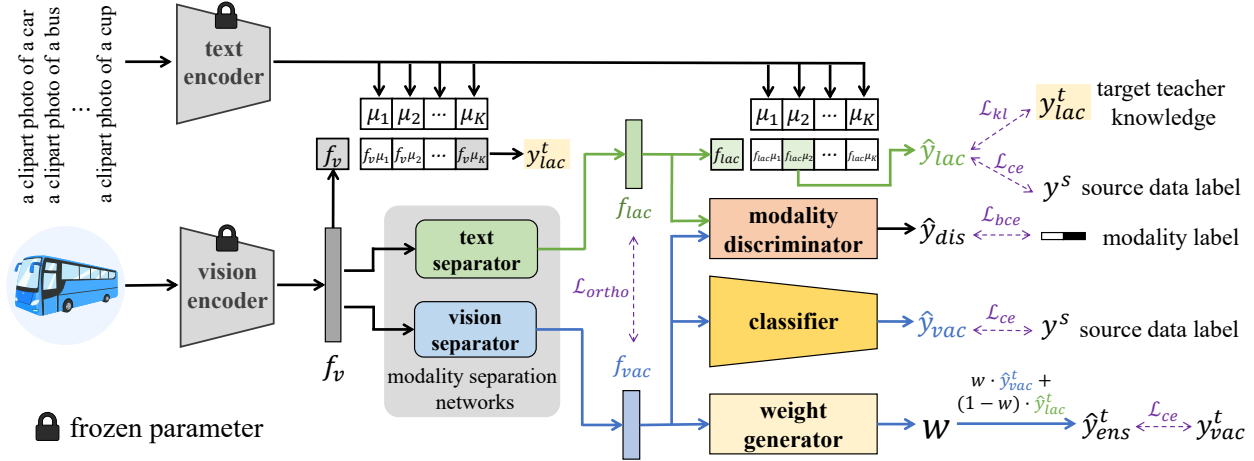
Figure 2. Framework of our method. We freeze the pretrained vision and text encoder of CLIP. CLIP-extracted vision features are disentangled into language-associated components (f_{lac}) and vision-associated components (f_{vac}) by the modality separation networks. We obtain zero-shot results from CLIP as teacher knowledge, and distill the knowledge to LAC. We then introduce a weight generator to assemble the modality outputs to train VAC. A modality discriminator is applied to align LAC and VAC from both domains.

less computation costs. Besides, our method requires no labeling on target domain. VLMs have also been adopted for UDA [10, 17, 41]. DAPrompt [10] proposes to learn domain-specific and domain-agnostic textual prompts for each class. AD-CLIP [41] learns domain invariant prompts by conditioning on image style and content features. PAD-CLIP [17] dynamically adjusts learning rate while tuning the CLIP vision branch to prevent catastrophic forgetting. However, these methods perform adaptation on either the visual or textual modality in isolation. Our work aims to address this limitation by proposing a unified adaptation framework of the multimodal features.

3. Method

3.1. Problem formulation

In this study, superscripts differentiate domains, with symbols lacking superscripts applicable to both domains. We consider a labeled source domain $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ and aim to develop a model generalizable to an unlabeled target domain $\mathcal{D}^t = \{(x_i^t)\}_{i=1}^{N^t}$. CLIP [36] features a vision encoder g_{vis} and a text encoder g_{txt}. The vision feature for an image input x is denoted as $f_v = g_{vis}(x)$. Employing the zero-shot inference strategy from [17], we construct naive prompts $\{(t_i)\}_{i=1}^{K}$ as *a [DOMAIN] photo of a [CLASS]*, with K representing the number of classes, *[DOMAIN]* indicating domain specifics, and *[CLASS]* the class name. Text features are then derived as $\mu_i = g_{txt}(t_i)$. μ_i and f_v are both features with d_v dimension. Classification is based on the highest cosine similarity between f_v and μ_i:

$$\hat{y}_{zs} = \arg\max_i \ \cos(\mu_i, f_v). \tag{1}$$

Eq. (1) may not be ideal for unlabeled target data due

to the existence of modality gap [24]. To tackle this, we conceptualize vision inputs as a composite of a vision-associated component (VAC) and a language-associated component (LAC), denoted as $f_v = \{f_{vac}, f_{lac}\}$. This leads us to obtain modality-specific classification results y_{vac} and y_{lac}, before constructing a cross-modality output:

$$y_{ens} = w \cdot y_{vac} + (1 - w) \cdot y_{lac}, \tag{2}$$

where w is a set of learnable weights harmonizing the contributions of VAC and LAC. This design seeks to balance modality-agnostic information sharing and modality-specific information capturing.

Instead of separating LAC and VAC during training, we utilize y_{ens} to guide VAC learning, incorporating complementary modality information for a holistic cross-modal training. Furthermore, a fixed weight w may lack flexibility across diverse datasets and scenarios, potentially obscuring the distinction between VAC and LAC. Addressing this, we introduce a dynamic w that adeptly discriminates between modalities within y_{ens}, calibrating their influence in the training. This strategy ensures tailored training approaches for different datasets or training stages, facilitating modality-specific information utilization. Next we detail on the training and aligning of LAC and VAC.

3.2. Modality separation networks

We first introduce the modality separation networks that disentangles CLIP-extracted features, which comprise two separators as shown in Fig. 2. These networks partition CLIP-extracted vision features into LAC and VAC using the text separator G_{txt} and the vision separator G_{vis}, respectively. The separated components are defined as $f_{lac} = G_{txt}(f_v)$ and $f_{vac} = G_{vis}(f_v)$. Both separators are linear layers preserving the dimensionality of f_v, such that

$f_{vac}, f_{lac} \in \mathbb{R}^{d_v}$. Drawing on deep feature separation principles [3], we apply an orthogonal loss to maintain the distinctness of LAC and VAC:

$$\mathcal{L}_{ortho} = |f_{lac}^s \cdot f_{vac}^{s\top}|_F^2 + |f_{lac}^t \cdot f_{vac}^{t\top}|_F^2. \quad (3)$$

Different outputs for LAC and VAC are then generated. For the text modality, we utilize the zero-shot inference of CLIP to classify LAC by calculating the cosine similarity between LAC and CLIP's text features, forming logits:

$$\hat{y}_{lac} = (\hat{l}_1, \hat{l}_2, \cdots \hat{l}_k), \quad \hat{l}_i = \cos(\mu_i, f_{lac})/T, \quad (4)$$

where T is temperature in pretrained CLIP. For VAC, we route it through a linear classifier with layers $\Phi_1 \in \mathbb{R}^{d_v \times d_b}$ and $\Phi_2 \in \mathbb{R}^{d_b \times K}$, producing the bottleneck feature with dimension d_b and output via:

$$f_b = \Phi_1(f_{vac}), \quad \hat{y}_{vac} = \Phi_2(f_b). \quad (5)$$

We provide implementation details in Supplementary.

3.3. Modality-ensemble training

Having obtained disentangled components, we design customized training paradigm for each modality. A learnable weight further connects both modalities, establishing a unified modality-ensemble training framework.

Learning LAC. To preserve the rich semantic content in pretrained CLIP, we distill this knowledge to LAC. For the target data, zero-shot similarity scores derived from pretrained CLIP serve as the teacher knowledge:

$$y_{lac}^t = (l_1 - \bar{l}, l_2 - \bar{l}, \cdots, l_k - \bar{l}), \quad l_i = \cos(\mu_i, f_v^t)/T, \quad (6)$$

with $\bar{l} = \frac{1}{K}\sum_{k=1}^K l_k$ normalizing the CLIP outputs and T the temperature of pretrained CLIP. The teacher knowledge in Eq. (6) guides the distillation for the unlabeled target LAC, while for the source data, cross-entropy loss is applied directly using labeled source data. The overall training loss for LAC combines Eq. (4) and Eq. (6) as follows:

$$\mathcal{L}_{lac} = \text{KL}(\hat{y}_{lac}^t, y_{lac}^t) + \alpha \text{CE}(\hat{y}_{lac}^s, y^s), \quad (7)$$

where α adjusts the influence of source data supervision, $\text{KL}(\cdot, \cdot)$ is the Kullback-Leibler divergence, and $\text{CE}(\cdot, \cdot)$ represents the standard cross-entropy loss.

Obtaining pseudo label for VAC.

Focusing on image modality, we aim to enhance the locality structure of vision representations—high inter-class discriminability and tight intra-class distribution—a feature that CLIP-extracted vision features lack, as detailed in Fig. 4a. To instill these locality structures within VAC, we utilize a K-means-based deep clustering approach [4, 21] to generate pseudo-labels for unlabeled target data. We calculate the clustering centroids for class k as follows:

$$\phi_k = \frac{\sum_{x^t} \delta_k(\text{softmax}(\hat{y}_{ens}^t)) \cdot f_b^t}{\sum_{x^t} \delta_k(\text{softmax}(\hat{y}_{ens}^t))}, \quad (8)$$

where $\hat{y}_{ens}^t$ represents target ensemble outputs discussed below, and δ_k selects the k_{th} logit. To mitigate imbalances in text modality predictions from CLIP [17, 49], we implement Approximated Controlled Direct Effect (ACDE) [49] to adjust similarity scores obtained in Eq. (4):

$$\tilde{y}_{lac}^t = \hat{y}_{lac}^t - \tau \log \hat{p}, \quad \hat{p} \leftarrow m\hat{p} + (1-m)\frac{1}{B}\sum_{i=1}^B p_i, \quad (9)$$

where m is momentum, τ is a debiasing factor, B is the batch size, and $p_i = \text{softmax}(\hat{y}_{lac}^t)$ denotes classification probability of LAC. The ensemble outputs, used in the centroid calculation, are then defined as $\hat{y}_{ens}^t = w \cdot \hat{y}_{vac}^t + (1 - w) \cdot \tilde{y}_{lac}^t$. For any given target bottleneck feature f_b^t, we compute its cosine similarity with all centroids, assigning the class with the highest similarity as the pseudo-label:

$$y_{vac}^t = \arg\max_k \cos(f_b^t, \phi_k). \quad (10)$$

Learning VAC. We now train vision component on unifies outputs from both modalities. Utilizing Eq. (9) and Eq. (5), the target ensemble output $\hat{y}_{ens}^t$ is formulated as:

$$\hat{y}_{ens}^t = w \cdot \hat{y}_{vac}^t + (1 - w) \cdot \tilde{y}_{lac}^t, \quad (11)$$

with the weight $w = W(VAC^t)$ produced by the weight generator W, as depicted in Fig. 2. Referring to Sec. 3.1, we optimize $\hat{y}_{ens}^t$ rather than $\hat{y}_{vac}^t$ directly, with $\tilde{y}_{lac}^t$ serving as an auxiliary in training VAC and thus detached from the computational graph in Eq. (11).

To enhance individual discriminability and global diversity, thereby preserving the locality structure of vision representations, we follow state-of-the-art [1, 20, 21] to apply an information maximization loss $\mathcal{L}_{im}$ comprising two components. The entropy loss $\mathcal{L}_{ent}$ improves individual certainty:

$$\mathcal{L}_{ent} = -\mathbb{E}_{x^t \in \mathcal{D}^t}\left[\sum_{k=1}^K \delta_k(\hat{y}_{ens}^t)\log\delta_k(\hat{y}_{ens}^t)\right], \quad (12)$$

and the diversity loss fosters diverse class distributions:

$$\mathcal{L}_{div} = -\sum_{k=1}^K \bar{q}_k \log \bar{q}_k, \quad (13)$$

where $\bar{q}_k = -\mathbb{E}_{x^t \in \mathcal{D}^t}\delta_k(\hat{y}_{ens}^t)$. Hence, $\mathcal{L}_{im}$ is defined as:

$$\mathcal{L}_{im} = \mathcal{L}_{ent} - \mathcal{L}_{div}. \quad (14)$$

The training of VAC is supervised by target pseudo labels for the vision modality, obtained through Eq. (10), while source labels directly optimize $\hat{y}_{vac}^s$:

$$\mathcal{L}_{vac} = CE(\hat{y}_{ens}^t, y_{vac}^t) + \beta CE(\hat{y}_{vac}^s, y^s) + \mathcal{L}_{im}, \quad (15)$$

where β modulates the impact of source data supervision.

4

3.4. Aligning source and target by discriminator

To achieve domain adaptation on both modalities, we introduce a modality discriminator D to align VAC and LAC from both domains. Our approach utilizes a singular modality discriminator trained on the source domain to differentiate LAC from VAC, and then assesses alignment on the target domain. Proper alignment across domains would enable D to discern LAC and VAC on the target domain without direct training. The modality discriminator is trained using binary cross-entropy loss:

$$\mathcal{L}_{bce} = -[y_{dis} \log \hat{y}_{dis} + (1 - y_{dis}) \log(1 - \hat{y}_{dis})], \quad (16)$$

where y_{dis} represents the modality label (0 for VAC, 1 for LAC) and $\hat{y}_{dis}$ is the output of D.

Notably, D is only trained on the source domain using Eq. (16). On the target domain, only the separators G_{vis} and G_{txt} are updated to minimize Eq. (16), aligning target LAC and VAC with the source ones.

3.5. Training and inference

Training. As depicted in Fig. 2, the pretrained text encoder and vision encoder are frozen, and we optimize parameters of G_{txt}, G_{vis}, Φ_1, Φ_2, W, D, denoted as $\theta_{G_{txt}}$, $\theta_{G_{vis}}$, θ_{Φ_1}, θ_{Φ_2}, θ_W, θ_D, respectively. Combining Eq. (7), Eq. (3), Eq. (15), Eq. (16), we define the following optimization problem:

$$\theta_{G_{txt}} = \underset{\theta_{G_{txt}}}{\arg\min} \ \mathcal{L}_{lac} + \gamma\mathcal{L}_{ortho} + \gamma\mathcal{L}_{bce}, \quad (17)$$

$$\theta_{G_{vis}} = \underset{\theta_{G_{vis}}}{\arg\min} \ \mathcal{L}_{vac} + \gamma\mathcal{L}_{ortho} + \gamma\mathcal{L}_{bce},$$

$$\theta_W, \theta_{\Phi_1}, \theta_{\Phi_2} = \underset{\theta_W, \theta_{\Phi_1}, \theta_{\Phi_2}}{\arg\min} \ \mathcal{L}_{vac},$$

$$\theta_D = \underset{\theta_D}{\arg\min} \ \gamma\mathcal{L}_{bce},$$

where γ is hyperparameter controlling regularization terms $\mathcal{L}_{bce}$ and $\mathcal{L}_{ortho}$. We present detailed training procedure of UniMoS in Supplementary.

Inference. At inference, the final mixed prediction on target data is obtained using $\hat{y}_{ens}^t$ from Eq. (11), with a fixed mixup weight w. The objective is to maximize accuracy by leveraging the strengths of both modalities for improved classification, as supported by our observations (Fig. 1), thus integrating outputs from both modalities to harness their combined advantages.

4. Experiments

4.1. Datasets and implementation details

We extensively evaluate our method on three mainstream UDA benchmarks. **Office-Home** [46] consists of 65 categories divided into 4 distinct domains. On **VisDA-**

2017 [34], the goal is to transfer knowledge from 152k synthetic images (source domain) to 55k images of real items (target domain). **DomainNet** [35] is the most challenging UDA benchmark so far, containing 0.6 million samples from 345 categories divided into 6 distinct domains. Following previous works, we additionally provide results on **Mini-DomainNet** [25, 40, 58], a subset of DomainNet with 4 domains and 126 categories.

We conduct all experiments on an NVIDIA RTX 2080Ti GPU. Since our method does not involve updating CLIP's pretrained parameters or prompts, the CLIP-extracted vision and text features are obtained via one single forward and saved in memory, thus greatly saving computation costs. For all tasks, we adopt SGD optimizer with batch size 32, and set momentum m in Eq. (9) to 0.99 and debias factor τ in Eq. (9) to 0.5. For Office-Home and Domain-Net, we train 50 epochs with initial learning rate 3e-3 and adopt annealing strategy [8] for learning rate decay. We train for 10 epochs with initial learning rate 9e-4 on VisDA due to fast convergence. The fixed mixup weight described in Sec. 3.5 is set to 0.3 for all tasks. We set regularization weight γ in Eq. (17) to 0.01 across all datasets.

4.2. Benchmark results

Office-Home. Tab. 1 gives classification accuracies on 12 adaptation tasks on Office-Home using the ResNet50 [11] backbone. To ensure a fair comparison, we categorize CLIP-based methods into two groups: 'none-tuning' and 'full-tuning'. The former involves learning prompts or additional modules without adjusting the pretrained CLIP backbones, while the latter optimizes the pretrained parameters of CLIP for specific tasks. It is evident from the results that our proposed UniMoS consistently outperforms both 'none-tuning' and 'full-tuning' methods. Notably, we obtain +1.3% performance boost than the strong baseline PADCLIP, which fine-tunes the CLIP vision backbone. Our method requires no parameter update or data forwarding of CLIP backbones, thus is much computationally cheaper. Especially on tasks that take P as target domain, we achieve up to +5.4% performance boost than PADCLIP, demonstrating the superiority of multimodal adaptation.

VisDA-2017. Tab. 2 shows class-wise classification accuracies on VisDA using ResNet101 [11]. Our method achieves the best performance among 'none-tuning' CLIP methods, while slightly falling behind PADCLIP. The reason is that CLIP has not been trained on the synthetic images like those from source domain of VisDA, resulting in incompatibilities between the VisDA dataset and CLIP. Similar observations are made by PADCLIP [17], which opts to fine-tune the vision branch of CLIP to address this challenge. Nevertheless, our approach outperforms typical UDA methods.

DomainNet. Tab. 3 presents classification accuracies of 30 cross-domain adaptation tasks on the most challeng-

Table 1. UDA results on Office-Home. Best results are marked in bold font. Methods with '*' are based on CLIP.

Method	Backbone	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
SourceOnly [11]	ResNet50	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
ParetoDA [22]		56.8	75.9	80.5	64.4	73.5	73.7	65.6	55.2	81.3	75.2	61.1	83.9	70.6
SDAT [37]		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 [50]		58.7	76.9	78.9	70.1	76.2	76.6	69.0	57.2	82.3	74.9	62.7	84.5	72.3
Fixbi [32]		58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
CST [26]		59.0	79.6	83.4	68.4	77.1	76.7	68.9	56.4	83.0	75.3	62.2	85.1	72.9
ATDOC [23]		60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	83.1	74.8	61.5	87.2	73.2
KUDA [42]		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
EIDCo [60]		**63.8**	80.8	82.6	71.5	80.1	80.9	72.1	61.3	84.5	**78.6**	65.8	87.1	75.8
ICON [55]		63.3	81.3	84.5	70.3	82.1	81.0	70.3	**61.8**	83.7	75.6	**68.6**	87.3	75.8
PADCLIP* [17]	ResNet50-full-tuning	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
CLIP* [36]	ResNet50-none-tuning	51.7	81.5	82.3	71.7	81.5	82.3	71.7	51.7	82.3	71.7	51.7	81.5	71.8
DAPrompt* [10]		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
ADCLIP* [41]		55.4	85.2	85.6	76.1	85.8	86.2	**76.7**	56.1	85.4	76.8	56.1	85.5	75.9
UniMoS* (ours)		59.5	**89.4**	**86.9**	75.2	**89.6**	**86.8**	75.4	58.4	**87.2**	76.9	59.5	**89.7**	**77.9**

Table 2. UDA results on VisDA-2017. Best results are marked in bold font. Methods with '*' are based on CLIP.

Method	Backbone	plane	bicycle	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
SourceOnly [11]	ResNet101	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
ParetoDA [22]		95.9	82.8	81.3	58.7	93.9	93.7	85.9	83.0	91.9	92.0	87.1	51.8	83.2
MSGD [50]		97.5	83.4	84.4	69.4	95.9	94.1	90.9	75.5	95.5	94.6	88.1	44.9	84.5
ATDOC [23]		95.3	84.7	82.4	75.6	95.8	97.7	88.7	76.6	94.0	91.7	91.5	61.9	86.3
CAN [15]		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
FixBi [32]		96.1	87.8	90.5	**90.3**	96.8	95.3	92.8	**88.7**	**97.2**	94.2	90.9	25.7	87.2
PADCLIP* [17]	ResNet101-full-tuning	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**
CLIP* [36]	ResNet101-none-tuning	**98.2**	83.9	90.5	73.5	97.2	84.0	95.3	65.7	79.4	89.9	91.8	63.3	84.4
DAPrompt* [10]		97.8	83.1	88.8	77.9	97.4	91.5	94.2	79.7	88.6	89.3	**92.5**	62.0	86.9
ADCLIP* [41]		98.1	83.6	**91.2**	76.6	**98.1**	93.4	**96.0**	81.4	86.4	91.5	92.1	64.2	87.7
UniMoS* (ours)		97.7	88.2	90.1	74.6	96.8	95.8	92.4	84.1	90.8	89.0	91.8	**65.3**	88.1

ing benchmark DomainNet. Rows represent source domains and columns represent target domains. Our method reaches comparable performance with the strong baseline PADCLIP. One significant observation is that our UniMoS obtains lower accuracy than PADCLIP (6.6% lower in average) on tasks with qdr as target. This discrepancy arises from a significant domain gap between qdr and other domains, which is challenging to bridge without tuning the CLIP backbones. However, despite this, we manage to achieve superior performance to PADCLIP on all other tasks, thereby mitigating the overall 6.6% performance drop. We also surpass all competing methods significantly. Tab. 4 compares UniMoS with available CLIP-based none-tuning methods on Mini-DomainNet, where our method achieves significant performance boost. Detailed results on Mini-DomainNet are given in Supplementary.

4.3. Ablation study

In this section we validate the efficacy of each module in UniMoS. Tab. 5 presents averaged accuracies on Office-Home and VisDA-2017 by removing specific modules while maintaining other settings identical. The 'w/o

debiasing' is obtained by skipping the debias procedure in Eq. (9). A primary observation is that the removal of any module leads to a performance drop to varying degrees, underscoring the positive contribution of each module to the overall outcome. The 'w/o learnable weight' row is obtained by replacing w in Eq. (11) with a constant 0.5, resulting in the most pronounced performance decline. This emphasizes the significance of dynamic weights, which enables VAC to focus on vision-specific parts. Further insights into the effects of dynamic w are detailed in Sec. 4.4.

Tab. 6 ablates on the choice of backbones. We experiment with three backbones on Office-Home, and our UniMoS consistently outperforms all competing methods, proving that our method is generalizable across various models. Detailed results are given in Supplementary.

4.4. Discussions

Effectiveness of learnable weight w. To better understand how the learnable ensemble weight w in Eq. (2) boosts performance, Fig. 3 compares the training process with and without dynamic w. We set a fixed weight of 0.5 for both LAC and VAC in 'w/o learnable w', and learn dynamic

Table 3. UDA results on DomainNet. Best results are marked in bold font. Methods with '*' are based on CLIP.

DeiT -B [44]	clp	inf	pnt	qdr	rel	skt	avg
clp	-	24.3	49.6	15.8	65.3	52.1	41.4
inf	45.9	-	45.9	6.7	61.4	39.5	39.9
pnt	53.2	23.8	-	6.5	66.4	44.7	38.9
qdr	31.9	6.8	15.4	-	23.4	20.6	19.6
rel	59.0	25.8	56.3	9.2	-	44.8	39.0
skt	60.6	20.6	48.4	16.5	61.2	-	41.5
avg	50.1	20.3	43.1	10.9	55.5	40.3	36.7

ViT -B [6]	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

SSRT -B [43]	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

CDTrans -DeiT [51]	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 -Swin [63]	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

CLIP -B* [36]	clp	inf	pnt	qdr	rel	skt	avg
clp	-	70.1	70.1	70.1	70.1	70.1	70.1
inf	46.4	-	46.4	46.4	46.4	46.4	46.4
pnt	61.7	61.7	-	61.7	61.7	61.7	61.7
qdr	13.7	13.7	13.7	-	13.7	13.7	13.7
rel	82.9	82.9	82.9	82.9	-	82.9	82.9
skt	62.6	62.6	62.6	62.6	62.6	-	62.6
avg	53.5	58.2	55.1	64.7	50.9	55.0	56.2

DAPrompt -B* [10]	clp	inf	pnt	qdr	rel	skt	avg
clp	-	73.0	73.8	72.6	73.9	73.5	73.4
inf	50.8	-	50.1	49.6	50.6	50.3	50.3
pnt	70.2	69.6	-	68.9	70.4	69.9	69.8
qdr	17.2	14.4	13.9	-	14.3	13.9	14.7
rel	84.9	84.8	84.9	84.7	-	84.6	84.8
skt	65.8	65.4	65.8	64.9	65.9	-	65.6
avg	57.8	61.4	57.7	68.1	55.0	58.4	59.8

PADCLIP -B* [17]	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

UniMoS -B* (ours)	clp	inf	pnt	qdr	rel	skt	avg
clp	-	76.5	77.2	76.6	77.5	77.8	**77.1**
inf	55.1	-	55.0	54.6	55.3	55.2	55.0
pnt	72.3	71.5	-	69.4	72.5	72.6	**71.7**
qdr	25.0	22.9	23.6	-	23.7	25.1	24.1
rel	86.0	85.9	85.8	85.5	-	85.9	**85.8**
skt	68.5	67.8	68.2	67.5	68.0	-	**68.0**
avg	61.4	64.9	62.0	70.7	59.4	63.3	63.6

Table 4. UDA results on Mini-DomainNet. Best results are marked in bold font. All compared methods are CLIP-based.

Method	Backbone	Acc.	Method	Backbone	Acc.
CLIP	ResNet50	71.2	CLIP	ViT-B	82.8
DAPrompt		74.8	DAPrompt		85.8
ADCLIP		75.2	ADCLIP		86.9
UniMoS (ours)		**78.0**	**UniMoS (ours)**		**87.3**

Table 5. Ablation study on Office-Home and VisDA-2017.

Method	Office-Home	VisDA-2017
w/o $\mathcal{L}_{ortho}$	77.4	87.6
w/o debiasing	77.3	87.7
w/o $\mathcal{L}_{im}$	77.0	87.8
w/o $\mathcal{L}_{distill}$	77.0	87.6
w/o learnable weight	76.9	86.2
w/o modality discriminator	77.6	87.9
UniMoS (full design)	**77.9**	**88.1**

Figure 3. Effects of learnable ensemble weight w on Office-Home.

weights (shown as 'Learned weight w' in the figure) in other settings. Our first observation is that in our full design, accuracy of VAC increases steadily as the training progresses. Leveraging complementary modality-specific information from LAC, the final mixed outputs achieve higher accuracy than VAC alone. As stated in Sec. 3.1, the goal of dynamic ensemble weight is to adaptively identify and preserve modality-specific information. We show in Fig. 3 that, the learned weight changes in each epoch to fit the training process. Without its support (w/o learnable w), the accuracy of VAC outputs drops significantly and finally converges with LAC. This occurs because employing a static weight would compromise the modality separation effects, causing both modalities to collapse to poor performance. In the example given by Fig. 3, accuracy of 'VAC output w/o learnable w' is 3.5% lower than that of the full de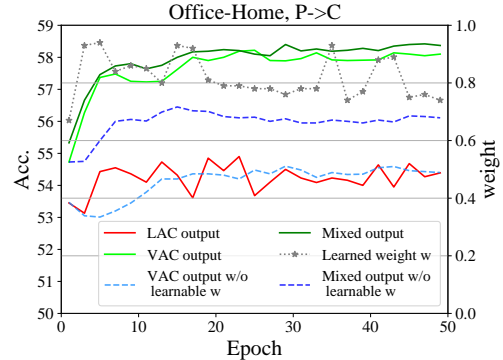sign 'VAC output'. The phenomenon indicates the significance of training with dynamic ensemble weights in our design. More examples are given in Supplementary.

Computation analysis. Tab. 7 compares computation costs of different methods on VisDA dataset. Our method necessitates training only a few linear layers without updating CLIP backbones, bringing great parameter efficiency. Furthermore, only one forward through CLIP is needed, which allows UniMoS achieve more than **47×** training speed boost than PADCLIP. Prompt learning methods like DAPrompt also requires no training on CLIP backbones, but they require extensive iterative forwarding of data through CLIP to learn optimal prompts per class, leading to low computing efficiency and scalability to larger datasets. When running on DomainNet with 345 classes, DAPrompt requires more than 22G GPU memory, while our method requires less than 3G. More details are in Supplementary.

Feature distribution visualization. To demonstrate the efficacy of modality separation and feature alignment, we perform t-sne [45] visualization on various features at different phases of our method. Fig. 4a shows CLIP-extracted vision features from source (green) and target (blue) domain, along
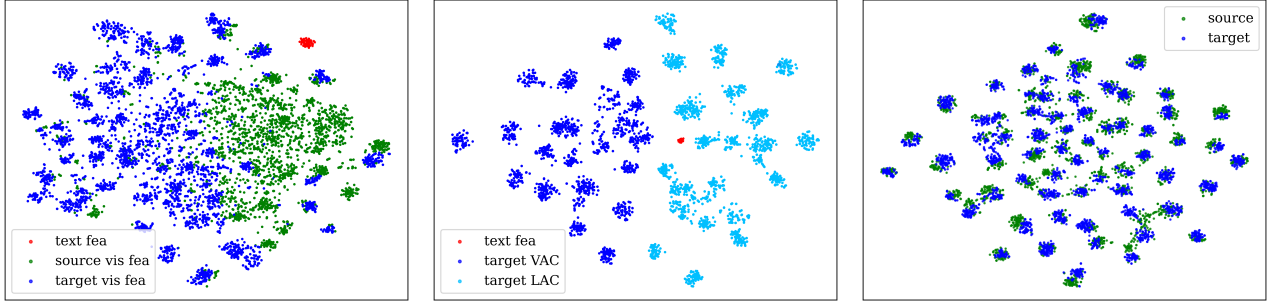
(a) CLIP-extracted feature distribution.

(b) Modality separation effect.

(c) Aligned bottleneck feature distribution.

Figure 4. T-sne visualization [45] of the effects of UniMoS on A→P task from Office-Home. UniMoS effectively disentangles CLIP-extracted vision features (Fig. 4a) into LAC and VAC (Fig. 4b, obtained by randomly selecting 25 classes), and constructs clear cross-domain locality structures (Fig. 4c).

Table 6. Results using different backbones on Office-Home. Best results are marked in bold font. All compared methods are based on CLIP.
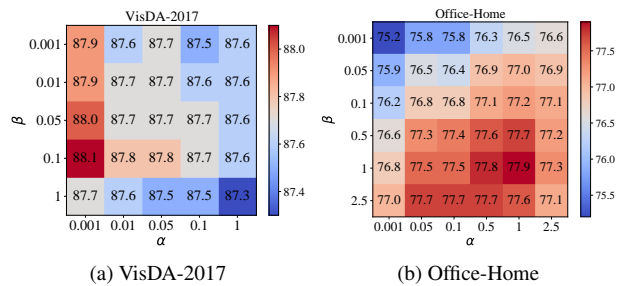
Method	Backbone	Acc.	Method	Backbone	Acc.
UniMoS (ours)	ResNet50	**77.9**	CLIP		82.4
CLIP		87.0	DAPrompt		84.4
DAPrompt	ViT-L	88.7	ADCLIP	ViT-B	86.1
ADCLIP		90.5	PADCLIP		86.7
UniMoS (ours)		**90.7**	**UniMoS (ours)**		**86.9**

Table 7. Computation analysis on VisDA-2017. Best results are marked in bold font. Methods with '*' are based on CLIP.

Method	Bkb.	Param.	Throughput (imges/s)	Train time	FLOPs	Acc.
DAPrompt*		1.2M	244	4.3H	11.3G	86.9
FixBi		86.1M	102	5.5H	15.73G	87.2
CAN	ResNet101	42.5M	31	10.5H	7.9G	87.2
PADCLIP*		-	-	23.5H*	-	**88.5**
UniMoS* (ours)		**0.79M**	**2667**	**0.5H**	**<0.01G**	88.1

(a) VisDA-2017

(b) Office-Home

Figure 5. Parameter sensitivity analysis on α and β of UniMoS.

in Eq. (7), β in Eq. (15), and γ in Eq. (17). We empirically discover that alternating γ has little impact within each dataset, so we focus on exploring the effects of α and β. As illustrated in Fig. 5a, the performance on VisDA is notably influenced by α, with smaller values leading to improved accuracy. This outcome is attributed to the fact, as discussed in Sec. 4.2, that CLIP struggles to adequately identify source synthetic images from VisDA. Consequently, down-weighting source supervision on LAC proves beneficial. Conversely, source supervision from both modalities of Office-Home are important. They positively and equally contributes to the adaptation process, so $\alpha = 1$ and $\beta = 1$ brings the best result, as shown in Fig. 5b.

5. Conclusions

Inspired by the theory of modality gap, in this paper we propose a Unified Modality Separation framework for unsupervised domain adaptation. The CLIP-extracted vision features are explicitly disentangled into vision-associated and language-associated components, which are trained differently according to their modality strengths and further aligned by a modality discriminator. A modality-ensemble training paradigm unifies both components to leverage modality-specific information while preserving modality-shared contexts, contributing to successful classification. This work is hope to inspire further analysis and exploitation of the multimodal features in pretrained VLMs.

with CLIP-extracted text features (red) of naive prompts. The text features distribute distantly with vision features, proving the existence of modality gap. Besides, CLIP-extracted vision features form poor class discriminability and locality structures. Fig. 4b showcases the separated LAC and VAC of our method. A clear boundary can be observed between the features of both modalities, indicating the effectiveness of modality separation. Additionally, the text features distribute closer to LAC, proving that the separated LAC is indeed more relevant to the text modality. Following feature alignment and VAC training, the bottleneck features f_b from both domains display a compact class-level locality structure, as demonstrated in Fig. 4c. This compact structure contributes significantly to the accuracy of the final classification results.

Hyperparameter sensitivity. In the calibration of Uni-MoS, we encounter three hyperparameters to determine: α

---

*Cited from PADCLIP [17]. PADCLIP conducts the experiments on an NVIDIA Tesla V100 GPU. Results on other metrics are unavailable since the authors have not released the code implementations yet.

Split to Merge: Unifying Separated Modalities for Unsupervised Domain Adaptation

Supplementary Material

Algorithm 1 Training Algorithm of UniMoS

Input: Labeled source data $\{x^s, y^s\}$, unlabeled target data x^t, maximum epoch number max_epoch, pretrained CLIP text encoder g_{txt} and vision encoder g_{vis}.

Output: Trained modality separator G_{txt}, G_{vis}, and trained linear layers Φ_1, Φ_2.

1: Let $epoch = 0$.
2: **while** $epoch < max\_epoch$ **do**
3:     Obtain teacher output y_{lac}^t via Eq. (18).
4:     **if** $epoch \mod 2 == 0$ **then**
5:         Take mixed outputs as target vision pseudo label via Eq. (20).
6:     **else**
7:         Obtain target vision pseudo label y_{vac}^t via Eq. (19).
8:     **end if**
9:     Obtain CLIP-extracted vision features $\{f_v^s, f_v^t\} = g_{vis}(x^s, x^t)$ and text features $\mu_i = g_{txt}(t_i)$.
10:    Obtain LAC outputs $\hat{y}_{lac}$ by Eq. (21).
11:    Obtain VAC outputs $\hat{y}_{vac}$ by Eq. (22).
12:    Obtain ensemble outputs $\hat{y}_{ens}$ by Eq. (24).
13:    Obtain modality discrimination outputs y_{dis} by Eq. (23).
14:    Update network parameters via Eq. (25).
15:    Let $epoch = epoch + 1$
16: **end while**
17: **return** solution

---

6. Training algorithm

For clarity, we give full training algorithm of UniMoS. We first obtain teacher output from CLIP's zero-shot inference results by Eq. (18):

$$y_{lac}^t = (l_1 - \bar{l}, l_2 - \bar{l}, \cdots, l_k - \bar{l}), \quad l_i = \cos(\mu_i, f_v^t)/T. \tag{18}$$

Then obtain pseudo label by Eq. (19):

$$\hat{y}_{vac}^t = \arg\max_k \cos(f_b^t, \phi_k). \tag{19}$$

To avoid potential error accumulations [**?**] in pseudo labels, we directly apply mixed outputs from both modalities as pseudo labels in certain epochs:

$$\hat{y}_{vac}^t = \lambda \cdot \hat{y}_{vac}^t + (1 - \lambda) \cdot \tilde{y}_{lac}^t, \tag{20}$$

where λ is a fixed mixup ratio 0.3 that combines outputs from both modalities for inference. LAC and VAC outputs

are obtained via Eq. (21) and Eq. (22), respectively.

$$\hat{y}_{lac} = (\hat{l}_1, \hat{l}_2, \cdots \hat{l}_k), \quad \hat{l}_i = \cos(\mu_i, f_{lac})/T. \tag{21}$$

$$\hat{y}_{vac} = \Phi_2(f_b), \quad f_b = \Phi_1(f_{vac}). \tag{22}$$

We further train a modality discriminator to align modality features from both domains via Eq. (23):

$$\mathcal{L}_{bce} = -[y_{dis} \log \hat{y}_{dis} + (1 - y_{dis}) \log(1 - \hat{y}_{dis})], \tag{23}$$

We assemble LAC and VAC during training to facilitate modality information exchanges by Eq. (24):

$$\hat{y}_{ens}^t = w \cdot \hat{y}_{vac}^t + (1 - w) \cdot \tilde{y}_{lac}^t. \tag{24}$$

Note that the w in Eq. (24) is trainable, which dynamically changes to suit different training phases. While the λ in Eq. (20) is fixed and only used for retrieving pseudo labels and inference, thus does not affect training. Based on the training objectives:

$$\theta_{G_{txt}} = \arg\min_{\theta_{G_{txt}}} \mathcal{L}_{lac} + \gamma\mathcal{L}_{ortho} + \gamma\mathcal{L}_{bce}, \tag{25}$$

$$\theta_{G_{vis}} = \arg\min_{\theta_{G_{vis}}} \mathcal{L}_{vac} + \gamma\mathcal{L}_{ortho} + \gamma\mathcal{L}_{bce},$$

$$\theta_W, \theta_{\Phi_1}, \theta_{\Phi_2} = \arg\min_{\theta_W, \theta_{\Phi_1}, \theta_{\Phi_2}} \mathcal{L}_{vac},$$

$$\theta_D = \arg\min_{\theta_D} \gamma\mathcal{L}_{bce},$$

we update the trainable parameters. Training algorithm is given in Algorithm 1. Please refer to main paper for full descriptions of the equations above.

All trainable modules in UniMoS are fully-connected linear layers, which brings very low computation costs. Specifically, the modality separators G_{txt}, G_{vis} are two separate linear layers of shape (d_v, d_v). The bottleneck feature dimension d_b is 256, thus Φ_1 is of shape $(d_v, 256)$ with batch normalization, and Φ_2 is of shape $(256, K)$. The modality discriminator D consists of two linear layers $(d_v, 256)$ and $(256, 1)$ with a ReLU activation layer. The weight generator W consists of two linear layers $(d_v, 256)$ and $(256, 1)$ with a Sigmoid activation layer.

7. Additional experiments

In this section we provide full results of our experiments.
Mini-DomainNet. We present full results on Mini-DomainNet [25, 40, 58] in Tab. 8 using two backbones. We set $\alpha = 0.1$, $\beta = 1$ and $\gamma = 0.01$ for all tasks. We

Table 8. Results on Mini-DomainNet. Best results are marked in bold font. Methods with '*' are based on CLIP.

Method	Backbone	clp→pnt	clp→rel	clp→skt	pnt→clp	pnt→rel	pnt→skt	rel→clp	rel→pnt	rel→skt	skt→clp	skt→pnt	skt→rel	Avg.
CLIP* [36]	ResNet50	67.9	84.8	62.9	69.1	84.8	62.9	69.2	67.9	62.9	69.1	67.9	84.8	71.2
DAPrompt* [10]		72.4	87.6	65.9	72.7	87.6	65.6	73.2	72.4	66.2	73.8	72.9	87.8	74.8
ADCLIP* [41]		71.7	88.1	66.0	73.2	86.9	65.2	73.6	73.0	68.4	72.3	74.2	**89.3**	75.2
UniMoS* (ours)		**76.0**	**88.9**	**72.1**	**75.5**	**89.2**	**71.1**	**75.1**	**75.9**	**70.5**	**76.4**	**76.3**	88.9	**78.0**
CLIP* [36]	ViT-B	80.3	90.5	77.8	82.7	90.5	77.8	82.7	80.3	77.8	82.7	80.3	90.5	82.8
DAPrompt* [10]		83.3	92.4	81.1	86.4	92.1	81.0	86.7	83.3	80.8	86.8	83.5	91.9	85.8
ADCLIP* [41]		84.3	**93.7**	82.4	**87.5**	**93.5**	82.4	**87.3**	84.5	81.6	**87.9**	84.8	93.0	86.9
UniMoS* (ours)		**86.2**	93.2	**83.2**	86.9	93.2	**83.2**	86.8	**86.0**	**82.8**	87.0	**86.2**	**93.3**	**87.3**

Table 9. Results on Office-Home. Best results are marked in bold font. Methods with '*' are based on CLIP.

Method	Backbone	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
SourceOnly [11]	ResNet50	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
ParetoDA [22]		56.8	75.9	80.5	64.4	73.5	73.7	65.6	55.2	81.3	75.2	61.1	83.9	70.6
CLIP* [36]		51.7	81.5	82.3	71.7	81.5	82.3	71.7	51.7	82.3	71.7	51.7	81.5	71.8
SDAT [37]		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 [50]		58.7	76.9	78.9	70.1	76.2	76.6	69.0	57.2	82.3	74.9	62.7	84.5	72.3
Fixbi [32]		58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
CST [26]		59.0	79.6	83.4	68.4	77.1	76.7	68.9	56.4	83.0	75.3	62.2	85.1	72.9
ATDOC [23]		60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	83.1	74.8	61.5	87.2	73.2
KUDA [42]		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
DAPrompt* [10]		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
EIDCo [60]		**63.8**	80.8	82.6	71.5	80.1	80.9	72.1	61.3	84.5	**78.6**	65.8	87.1	75.8
ICON [55]		63.3	81.3	84.5	70.3	82.1	81.0	70.3	**61.8**	83.7	75.6	**68.6**	87.3	75.8
ADCLIP* [41]		55.4	85.2	85.6	76.1	85.8	86.2	**76.7**	56.1	85.4	76.8	56.1	85.5	75.9
PADCLIP* [17]		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
UniMoS* (ours)		59.5	**89.4**	**86.9**	75.2	**89.6**	**86.8**	75.4	58.4	**87.2**	76.9	59.5	**89.7**	**77.9**
SourceOnly [6]	ViT-B	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
CLIP* [36]		67.8	89.0	89.8	82.9	89.0	89.8	82.9	67.8	89.8	82.9	67.8	89.0	82.4
TVT [52]		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
DAPrompt* [10]		70.6	90.2	91.0	84.9	89.2	90.9	84.8	70.5	90.6	84.8	70.1	90.8	84.0
SSRT [43]		75.2	89.0	91.1	85.1	88.3	89.9	85.0	74.2	91.2	85.7	78.6	91.8	85.4
ADCLIP* [41]		70.9	92.5	92.1	85.4	92.4	**92.5**	**86.7**	74.3	**93.0**	**86.9**	72.6	93.8	86.1
PADCLIP* [17]		**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
UniMoS* (ours)		74.9	**94.0**	**92.5**	86.4	**94.3**	**92.5**	86.0	73.9	**93.0**	86.4	74.2	**94.5**	**86.9**
CLIP* [36]	ViT-L	74.2	93.1	93.3	87.3	93.1	93.3	87.3	74.2	93.3	87.3	74.2	93.1	87.0
DAPrompt* [10]		77.3	94.6	94.3	88.6	94.6	94.0	88.8	76.8	94.0	89.0	77.8	94.4	88.7
ADCLIP* [41]		80.3	95.4	**95.7**	**90.9**	95.5	**95.2**	**90.1**	79.6	95.1	**90.8**	81.1	95.9	90.5
UniMoS* (ours)		**80.9**	**96.2**	95.1	90.1	**96.1**	95.1	90.0	**81.4**	**95.2**	89.9	**81.6**	**96.3**	**90.7**

can observe significant improvements over current SOTA ADCLIP [41] by 2.8% in average with ResNet50 [11], and 0.4% improvement using ViT [6]. We are unable to compare with PADCLIP [17] since no public code implementation is available yet.

Office-Home. We provide full results on Office-Home [46] in Tab. 9 using ResNet50 and two variants of ViT as backbone. Utilizing the rich pretrain knowledge in CLIP, CLIP-based methods achieve higher accuracies than single-modality UDA methods. Our method further consistently outperforms all competing methods on all backbones, demonstrating the efficacy of adapting both modalities.

Computation analysis. We provide comprehensive computation analysis in Tab. 10. The GPU memory consumption is computed using consistent batch size of 32. The re-

sults on DomainNet is computed on task 'clp→inf'. Note our method requires only one forward through CLIP to obtain pre-extracted features, and we have included the time for pre-extraction in the 'Train time' columns. According to Sec. 6, only a few linear layers is trained in our method, which brings significant computation efficiency, allowing agile applications of our method. We can observe that the trainable parameters of DAPrompt differs on different datasets. The reason is that DAPrompt trains the prompt embedding for each class respectively, therefore the trainable parameters and GPU consumption increase drastically as class number increases from 12 (VisDA) to 345 (DomainNet).

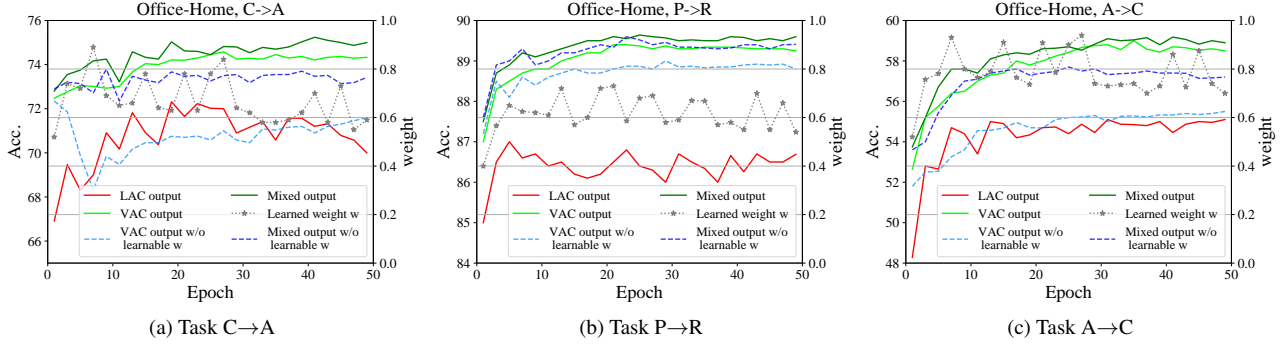**Learnable ensemble weight w.** Fig. 6 provides additional examples on the effects of dynamic ensemble weight on

Figure 6. Examples on the effects of ensemble weight w.

Table 10. Computation analysis. Best results are marked in bold font. Methods with '*' are based on CLIP.

Method	Dataset	Backbone	Param.	Throughput (imges/s)	Train time	FLOPs	GPU mem.	Acc.
DAPrompt*			1.2M	244	4.3H	11.3G	6.9G	86.9
FixBi			86.1M	102	5.5H	15.73G	17.0G	87.2
CAN	VisDA	ResNet101	42.5M	31	10.5H	7.9G	11.3G	87.2
PADCLIP*			-	-	23.5H	-	-	**88.5**
UniMoS* (ours)			**0.79M**	**2667**	**0.5H**	**<0.01G**	**1.8G**	88.1
PMTrans		Swin	89.5M	46	30H	15.2G	19.3G	52.4
DAPrompt*	DomainNet	ViT-B	34.5M	31	7.9H	73.9G	22.5G	59.8
UniMoS* (ours)			**0.79M**	**2827**	**0.4H**	**<0.01G**	**2.9G**	**63.6**

three tasks from Office-Home dataset. We can draw the conclusion that the accuracy of VAC drops drastically without learning ensemble weight w, which further affects the overall mixed output accuracy. We can observe in Fig. 6b that although accuracy of 'VAC output w/o learnable w' is lower than the full design, the final 'Mixed output w/o learnable w' achieves comparable accuracy with the full design. The reason is that the task P→R is relatively easier, thus complementary knowledge in LAC is able to support VAC for classification.

References

[1] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10103–10112, 2021. 4

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

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016. 4

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 4

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[6] 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. In *International Conference on Learning Representations*, 2020. 2, 7, 10

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 1, 2

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 5

[9] 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. *International Journal of Computer Vision*, pages 1–15, 2023. 2

[10] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2023. 1, 2, 3, 6, 7, 10

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 5, 6, 10

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[13] 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 *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 1, 2

[14] Qian Jiang, Changyou Chen, Han Zhao, Liqun Chen, Qing Ping, Son Dinh Tran, Yi Xu, Belinda Zeng, and Trishul Chilimbi. Understanding and constructing latent modality structures in multi-modal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7661–7671, 2023. 1, 2

[15] 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. 2, 6

[16] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 2

[17] 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. 1, 3, 4, 5, 6, 7, 8, 10

[18] 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

[19] Jingjing Li, Zhekai Du, Lei Zhu, Zhengming Ding, Ke Lu, and Heng Tao Shen. Divergence-agnostic unsupervised domain adaptation by adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8196–8211, 2021. 1

[20] Xinyao Li, Zhekai Du, Jingjing Li, Lei Zhu, and Ke Lu. Source-free active domain adaptation via energy-based locality preserving transfer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5802–5810, 2022. 2, 4

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020. 4

[22] Jian Liang, Kaixiong Gong, Shuang Li, Chi Harold Liu, Han Li, Di Liu, Guoren Wang, et al. Pareto domain adaptation. *Advances in Neural Information Processing Systems*, 34:12917–12929, 2021. 6, 10

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16632–16642, 2021. 6, 10

[24] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022. 1, 2, 3

[25] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7640–7650, 2023. 5, 9

[26] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021. 6, 10

[27] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016. 1

[30] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 1, 2

[31] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pretraining. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022. 2

[32] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1094–1103, 2021. 6, 10

[33] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. *arXiv preprint arXiv:2302.12066*, 2023. 2

[34] 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

[35] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 5

[36] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6, 7, 10

[37] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *International Conference on Machine Learning*, pages 18378–18399. PMLR, 2022. 6, 10

[38] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. 2

[39] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 1, 2

[40] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019. 5, 9

[41] Mainak Singha, Harsh Pal, Ankit Jha, and Biplab Banerjee. Ad-clip: Adapting domains in prompt space using clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4355–4364, 2023. 1, 3, 6, 10

[42] Tao Sun, Cheng Lu, and Haibin Ling. Prior knowledge guided unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 639–655. Springer, 2022. 6, 10

[43] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7191–7200, 2022. 7, 10

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2, 7

[45] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 7, 8

[46] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 5, 10

[47] Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. Ufo: A unified transformer for vision-language representation learning. *arXiv preprint arXiv:2111.10023*, 2021. 2

[48] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 2

[49] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14647–14657, 2022. 4

[50] Haifeng Xia, Taotao Jing, and Zhengming Ding. Maximum structural generation discrepancy for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3434–3445, 2022. 6, 10

[51] Tongkun Xu, Weihua Chen, WANG Pichao, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *International Conference on Learning Representations*, 2021. 2, 7

[52] 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, 10

[53] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in neural information processing systems*, 34:29393–29405, 2021. 2

[54] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, Shangling Jui, and Jian Yang. Trust your good friends: Source-free domain adaptation by reciprocal neighborhood clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[55] Zhongqi Yue, Hanwang Zhang, and Qianru Sun. Make the u in uda matter: Invariant consistency learning for unsupervised domain adaptation. *arXiv preprint arXiv:2309.12742*, 2023. 2, 6, 10

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[57] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2

[58] Wenyu Zhang, Li Shen, and Chuan-Sheng Foo. Rethinking the role of pre-trained networks in source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18841–18851, 2023. 5, 9

[59] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International conference on machine learning*, pages 7404–7413. PMLR, 2019. 2

[60] Yixin Zhang, Zilei Wang, Junjie Li, Jiafan Zhuang, and Zihan Lin. Towards effective instance discrimination contrastive loss for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11388–11399, 2023. 2, 6, 10

[61] 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. 2

[62] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2

[63] 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. 2, 7