Universal Semi-Supervised Domain Adaptation by Mitigating Common-Class Bias

Wenyu Zhang[1], Qingmu Liu[2]*, Felix Ong Wei Cong[2]*, Mohamed Ragab[1,3], Chuan-Sheng Foo[1,3]
[1]Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR)
[2]National University of Singapore (NUS)
[3]Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR)

Abstract

Domain adaptation is a critical task in machine learning that aims to improve model performance on a target domain by leveraging knowledge from a related source domain. In this work, we introduce Universal Semi-Supervised Domain Adaptation (UniSSDA), a practical yet challenging setting where the target domain is partially labeled, and the source and target label space may not strictly match. UniSSDA is at the intersection of Universal Domain Adaptation (UniDA) and Semi-Supervised Domain Adaptation (SSDA): the UniDA setting does not allow for fine-grained categorization of target private classes not represented in the source domain, while SSDA focuses on the restricted closed-set setting where source and target label spaces match exactly. Existing UniDA and SSDA methods are susceptible to common-class bias in UniSSDA settings, where models overfit to data distributions of classes common to both domains at the expense of private classes. We propose a new prior-guided pseudo-label refinement strategy to reduce the reinforcement of common-class bias due to pseudo-labeling, a common label propagation strategy in domain adaptation. We demonstrate the effectiveness of the proposed strategy on benchmark datasets Office-Home, DomainNet, and VisDA. The proposed strategy attains the best performance across UniSSDA adaptation settings and establishes a new baseline for UniSSDA.

1. Introduction

Domain adaptation (DA) is a critical task in machine learning, where models are adapted from a source domain to perform well on a different target domain. Conventionally, DA works focus on the closed-set adaptation setting, where the source and target label space match exactly, to address covariate shift and potential label distribution shift [11, 38, 40, 47]. More recently, the generalized problem in unsupervised DA termed 'Universal DA' (UniDA) [4, 32, 46, 49] aims to bridge the gap between labeled source data and unlabeled target data, and considers challenging settings where there may be label space shift between the two domains. This means that, under UniDA, there may be source private and target private classes besides common classes shared across the two domains, as in the open-set, partial-set, and open-partial settings. The universal setup handles practical scenarios where different object classes are present in different environments, or when new task objectives require collecting new target domain object classes not present in the source domain. However, since the UniDA setting assumes target data is unlabeled, all target private classes are categorized under a single 'unknown' class. Unsupervised domain adaptation is also known to be prone to negative transfer as target classes can be mapped to incorrect source classes [17, 48].

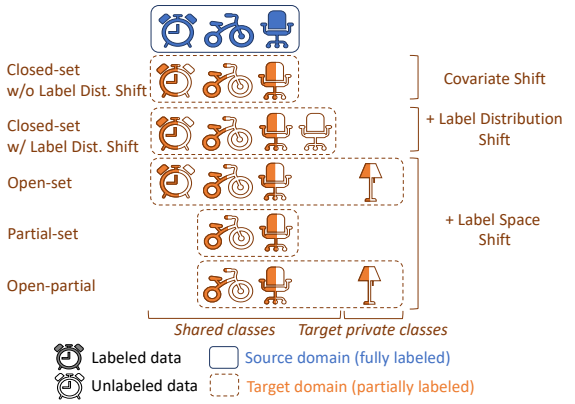
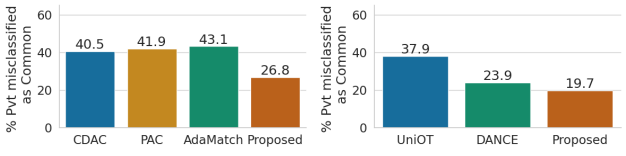


Figure 1. Adaptation settings in Universal SSDA. Existing SSDA works conventionally focus on closed-set settings.

In this work, we introduce a new setting to include a small number of labeled target samples in UniDA such that methods can provide fine-grained classification of target private class samples and better leverage target information that may be available. While the related semi-supervised domain adaptation (SSDA) setting also allows

*Contributed to this work while interning at A*STAR.



(a) Comparison with SSDA (b) Comparison with UniDA

Figure 2. (a) and (b) show the percentage of samples in target private classes ($\mathcal{Y}_T \cap \mathcal{Y}_S^c$) misclassified as common classes ($\mathcal{Y}_T \cap \mathcal{Y}_S$) under open-partial setting, demonstrating the effect of common-class bias on existing SSDA and UniDA methods. (a) is implemented on DomainNet-126 $C \rightarrow P$ with ResNet-34 backbone. (b) is implemented on DomainNet-345 $C \rightarrow P$ with frozen ViT foundation model encoder and learnable classifier.

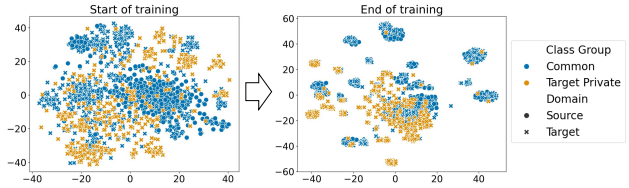


Figure 3. T-SNE visualization of common versus target private class samples shows negative transfer due to common-class bias. Incorrect mapping between common and target private classes can persist and be reinforced by naive pseudo-labeling. Example taken from DomainNet-126 $C \rightarrow P$ in open-partial setting, with visualization on 15 shared and 15 target private classes.

for partially labeled target data and offers a practical balance between data annotation and performance, works have been restricted to the closed-set setting [21, 24, 35]. We thus propose the new generalized setting at the intersection of UniDA and SSDA as 'Universal SSDA' (UniSSDA). In UniSSDA, methods need to improve adaptation performance regardless of the source and target label space as illustrated in Figure 1. To the best of our knowledge, our work is the first study on UniSSDA.

A key challenge in UniSSDA is that due to the abundance of labeled source data, model learning is naturally biased towards fitting the source distribution. This overfitting is especially detrimental to the learning of target private classes in UniSSDA settings, since these classes are not represented at all in the source domain. We refer to this bias as the 'common-class bias', where model learning focuses on classes common to both domains at the expense of private classes. From Figure 2, we observe that existing SSDA and UniDA methods are vulnerable to the common-class bias in a UniSSDA open-partial setting on DomainNet, and misclassify as much as 40% target private samples as belonging to common classes.

We hypothesize that the propagation of biased label information to the unlabeled target samples resulted in the eventual negative transfer observed in target private classes. To gain further insight into this issue, we study the pseudo-labeling component commonly used in DA methods [21, 22, 41, 42, 45], as naive pseudo-labeling tends to reinforce pre-existing biases as the model iteratively fits to incorrect pseudo-labels during training.

Through t-SNE visualizations of the feature space, Figure 3 illustrates the effects of the common-class bias when training utilizes a naive pseudo-labeling strategy where the class with the highest confidence score is assigned as the pseudo-label, and pseudo-labeled samples are selected by a confidence threshold. Target private classes are incorrectly mapped to common classes, and this incorrect mapping is reinforced during training. In this work, we propose a pseudo-label refinement strategy to mitigate common-class bias. We introduce prior-guidance to directly reweigh and refine the predictions on unlabeled target samples. The strategy is simple to implement and effective, and the refined pseudo-labels can be readily incorporated into existing adaptation algorithms.

Our key contributions in this work are:

- We are the first to study Universal SSDA (UniSSDA), a new setup at the intersection of UniDA and SSDA. Unlike UniDA, UniSSDA allows fine-grained classification of target private classes and is able to leverage available target label information. UniSSDA can also be viewed as a practical generalization of SSDA that removes the restriction for source and target label space to strictly match.
- Experimental evaluations find that existing SSDA and UniDA methods do not consistently perform well in UniSSDA settings and are susceptible to common-class bias. This highlights the need to develop new approaches to address UniSSDA.
- We propose a prior-guided pseudo-label refinement strategy to reduce the reinforcement of common-class bias due to target pseudo-labeling, a common label propagation strategy in DA problems.
- We demonstrate the performance of the proposed strategy on 3 domain adaptation image classification datasets. The proposed strategy establishes a new UniSSDA baseline for future research work in this area.

2. Related Works

2.1. Semi-Supervised Domain Adaptation

Existing works in SSDA aim to learn domain invariant representations and to mine intrinsic target domain structures. Methods focused on domain invariance align features [5, 15, 18, 22, 35, 41, 44, 45], label distributions [1] or hypotheses [6, 14, 16, 18, 25] between domains to transfer knowledge from source to target domain. [5] directly maps source to target data manifold by learning a linear transformation. Following works in unsupervised domain adaptation [11, 38, 40], many SSDA methods reduce feature distribution mismatch by adversarial learning or minimizing a domain discrepancy measure [15, 18, 22, 35, 41, 44, 45]. CLDA [35] uses contrastive learning and ECACL [22] uses

a triplet loss to simultaneously pull same-class samples together and push different-class samples apart.

Since both labeled and unlabeled target samples are available during training, methods employ strategies including those from semi-supervised learning [43] and self-supervised learning [26] to mine structures and information from target data. PAC [24] uses rotation prediction as a pre-training task, and [19] adds meta-learning to an existing SSDA method to find better model initializations. Several methods apply clustering objectives on the target domain to learn class-discriminative target features [8, 21, 33, 41, 44, 45]. Sample similarity for clustering can be determined based on pseudo-label [45], inter-sample distance [44], or a given similarity graph [8]. Some methods encourage prediction consistency of samples under different augmentations so as to find a smooth data manifold and to learn compact target clusters [1, 21, 22, 24, 35, 41].

To propagate label information to unlabeled samples, methods typically use pseudo-labeling. BiAT [13] and S^3D [45] generate intermediate styles to bridge the domain gap between source and target domains to facilitate label propagation. Some methods use uncertainty measures, such as confidence and entropy, for data selection to improve pseudo-labeling accuracy [21, 22, 41, 42, 45]. However, pseudo-labeling risks reinforcing pre-existing source-induced bias, and we propose pseudo-label refinement strategies to mitigate such bias using duo classifiers. DST [3], a semi-supervised learning method, also utilizes an additional classifier, but DST cannot be applied to frozen foundation model feature extractors, and DST's main classifier may not fully represent the data distribution as it is not trained on any unlabeled samples.

2.2. Universal Domain Adaptation

UniDA comprises settings where the source and/or target domain can have private classes, and assumes all target samples to be unlabeled. Similar to SSDA methods in Section 2.1, UniDA methods exploit intrinsic structures in target data by discovering clusters [2, 20, 34]. UniDA methods also aim to learn domain invariant representations through domain alignment. MATHS [4] uses contrastive learning between mutual nearest neighbor samples for domain alignment, and UniAM [49] achieves domain-wise and category-wise alignment by feature and attention matching between domains in vision transformers. A key challenge in UniDA is to distinguish between common and private class samples such that alignment is only performed on the former, so as to avoid negative transfer. OVANet [32] trains a one-vs-all classifier for each class to estimate the inter-class distance as a threshold to distinguish between the two class groups. Other methods use uncertainty measures such as entropy, confidence and consistency [2, 10, 34, 36, 46], sample similarity between domains [36, 46], or outlier detection on the logits [4] to detect target private class samples. A common assumption in UniDA is that the target private class samples are predicted with higher uncertainty because there are no labeled data from target private classes for supervised training. The assumption is not applicable in UniSSDA. A recent study finds that the supervised baseline is competitive with or outperforms existing UniDA methods on foundation models [7], hence we include foundation models in our experiments for a more comprehensive evaluation.

3. UniSSDA Preliminaries

We denote the input and output space as $\mathcal{X}$ and $\mathcal{Y}$. Distribution shift occurs when the source and target distribution differ (i.e. $p_S(x,y) \neq p_T(x,y)$), for instance in covariate shift (i.e. $p_S(x) \neq p_T(x)$) and label shift (i.e. $p_S(y) \neq p_T(y)$). Same as other domain adaptation tasks [11, 21, 46], UniSSDA assumes the presence of covariate shift. In existing SSDA works, methods and evaluations are focused on the closed-set setting where source and target domain have the same label space (i.e. $\mathcal{Y}_\mathcal{S} = \mathcal{Y}_\mathcal{T}$), and label shift is restricted to label distribution shift where $\mathcal{Y}_\mathcal{S} = \mathcal{Y}_\mathcal{T}$ and $\exists y \in \mathcal{Y}_\mathcal{S}\ p_S(y) \neq p_T(y)$.

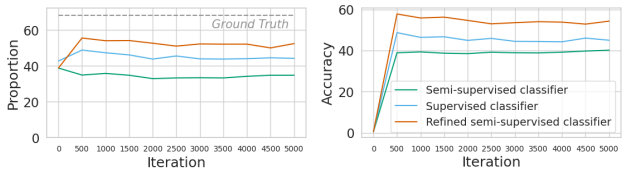
Label space shift involves a change in label set (i.e. $\mathcal{Y}_\mathcal{S} \neq \mathcal{Y}_\mathcal{T}$) [46], and UniSSDA methods need to be effective in these more challenging types of domain shift. To the best of our knowledge, we are the first to study UniSSDA. We consider the adaptation settings in Figure 1:

- Closed-set with no label distribution shift (i.e. $\mathcal{Y}_\mathcal{S} = \mathcal{Y}_\mathcal{T}$, $\forall y \in \mathcal{Y}_\mathcal{S}\ p_S(y) = p_T(y)$);
- Closed-set with label distribution shift (i.e. $\mathcal{Y}_\mathcal{S} = \mathcal{Y}_\mathcal{T}$, $\exists y \in \mathcal{Y}_\mathcal{S}\ p_S(y) \neq p_T(y)$);
- Open-set: Source label space is a proper subset of target label space (i.e. $\mathcal{Y}_S \subset \mathcal{Y}_T$);
- Partial-set: Target label space is a proper subset of source label space (i.e. $\mathcal{Y}_T \subset \mathcal{Y}_S$);
- Open-partial: Source and target label space intersect but neither is a proper subset of the other (i.e. $\mathcal{Y}_S \cap \mathcal{Y}_T \neq \emptyset$ and $\mathcal{Y}_S \cap \mathcal{Y}_T^c \neq \emptyset$ and $\mathcal{Y}_T \cap \mathcal{Y}_S^c \neq \emptyset$).

Classes can be categorized into 3 groups based on domain label space membership: common ($\mathcal{Y}_T \cap \mathcal{Y}_S$), source private ($\mathcal{Y}_T^c \cap \mathcal{Y}_S$), and target private ($\mathcal{Y}_T \cap \mathcal{Y}_S^c$).

4. Methodology

We denote the input, feature, logit and output space as $\mathcal{X}, \mathcal{Z}, \mathcal{G}$ and $\mathcal{Y}$, respectively. With a neural network model $h \circ f$, the feature extractor f parameterized by Θ learns the mapping $f : \mathcal{X} \to \mathcal{Z}$, and classifier h parameterized by Ψ learns the mapping $h : \mathcal{Z} \to \mathcal{G}$. For a logit $g \in \mathcal{G}$, we obtain the predictive probability $p = \sigma(g)$ using softmax function σ, and $\hat{y} = \arg\max_i(p[i]) \in \mathcal{Y}$ as the predicted label. Let ℓ denote the labeled source and target, and u denote the unlabeled target.



(a) Proportion of predictions in target private class

(b) Target private class prediction accuracy

Figure 4. The semi-supervised classifier trained with naive pseudo-labels is more vulnerable to common-class bias than the supervised classifier is. Using supervised classifier outputs as priors to refine the pseudo-labels significantly improves the performance of the resulting semi-supervised classifier. Example taken from DomainNet $C \rightarrow P$ in open-partial setting.

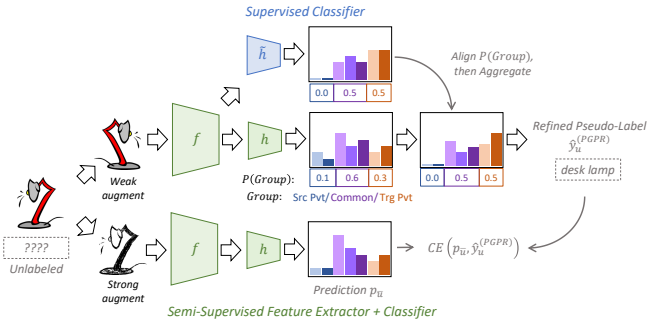


Figure 5. Proposed Prior-Guided Pseudo-label Refinement (PGPR) strategy: During adaptation, we add a supervised classification head $\tilde{h}$ trained only on labeled samples to provide prior distribution estimates. We align the per-instance class group distribution output by $h \circ f$ to the estimate by $\tilde{h} \circ f$ and then aggregate the two classifier decisions. The class with the highest resulting probability is taken as the refined pseudo-label for training. The supervised classfier $\tilde{h}$ is discarded at the end of training.

4.1. Supervised vs. Semi-Supervised Classifier

Due to the abundance of labeled source data, a vanilla model naturally overfits to the source data distribution. This overfitting is especially detrimental to the learning of target private classes in UniSSDA since these classes are not represented at all in the source domain. Consequently, the model focuses learning on classes common to both domains, at the expense of target private classes. We refer to this bias as the 'common-class bias'. Naive pseudo-labeling assigns the class with the highest confidence score as the pseudo-label for an unlabeled sample. Using naive pseudo-labels to propagate label information to the unlabeled target samples risks reinforcing pre-existing bias. In the example on open-partial setting in Figure 4, the semi-supervised classifier trained with naive pseudo-labels severely underestimates the proportion of target private class samples throughout the training process. In comparison, by fitting a supervised classification head on top of the same feature extractor, we observe that the supervised classifier is less susceptible to common-class bias and has higher accuracy on target private classes. By refining the pseudo-labels with the supervised classier via our proposed strategy and using them to train the semi-supervised classifier, the refined classifier achieves the highest accuracy.

4.2. Prior-Guided Pseudo-Label Refinement

From Figure 4, we observe that the supervised classifier is less susceptible to common-class bias than the naive semi-supervised classifier is. Motivated by this observation, we propose to refine the predictive probabilities output by $h \circ f$ guided by per-instance priors estimated using a supervised classifier. On top of the feature extractor f, we add a new linear classification head $\tilde{h} : \mathcal{Z} \rightarrow \mathcal{G}$ parameterized by $\tilde{\Phi}$ learned only on labeled source and target samples. Depending on the domain of the input sample, we apply domain-specific classifier masks to mask logits of classes absent from the domain. The classifier $\tilde{h}$ is trained alongside $h \circ f$ using cross-entropy loss:

$$\tilde{L}_\ell(\tilde{\Phi}) = -y_\ell \cdot \log(\tilde{p}_\ell) \tag{1}$$

where $\tilde{p}$ is the predictive probability output by $\tilde{h} \circ f$ and y is the ground-truth label. Gradients are blocked in the feature extractor f.

The goal of the proposed Prior-Guided Pseudo-label Refinement (PGPR) strategy is to reduce the effect of source-induced bias on target pseudo-labels. On each unlabeled sample, we adjust the predictive probability p_u originally estimated by $h \circ f$ using the prior distribution $\tilde{p}_u$ estimated by the supervised classifier $\tilde{h} \circ f$. We apply a group reweighted refinement step followed by a classifier aggregated refinement step. Group reweighted refinement aligns the class group distribution. The classes are grouped based on the domain(s) they are present in i.e. source private ($\mathcal{Y}_T^c \cap \mathcal{Y}_S$), target private ($\mathcal{Y}_T \cap \mathcal{Y}_S^c$), and common ($\mathcal{Y}_T \cap \mathcal{Y}_S$) classes. We treat $\tilde{p}_u$ as the prior to compute the probability of each class group grp, and reweigh p_u output by $h \circ f$ such that $\sum_{j \in grp} p_u^{(reweighted)}[j] = \sum_{j \in grp} \tilde{p}_u[j]$, by:

$$p_u^{(reweighted)}[i] = p_u[i] \cdot \frac{\sum_{j \in grp} \tilde{p}_u[j]}{\sum_{j \in grp} p_u[j]} \tag{2}$$

for $i \in grp$. That is, for each class group, the group distribution from $p_u^{(reweighted)}$ is instance-wise aligned to that from $\tilde{p}_u$. Class aggregated refinement further adjusts individual class probabilities by aggregating the decisions from the two classifiers to give the final predictive probability:

$$p_u^{(PGPR)} = \left(p_u^{(reweighted)} + \tilde{p}_u \right) / 2. \tag{3}$$

The class with the highest probability in $p_u^{(PGPR)}$ is taken as the pseudo-label $\hat{y}_u^{(PGPR)}$.

Taken together, group reweighted refinement can be viewed as a strict and coarse-grained regularization on the

estimated class group distribution, and classifier aggregated refinement can be viewed as a soft and fine-grained regularization on the estimated class distribution.

4.3. Training Objectives

The overall training objective is:

$$L(\Theta, \Psi) = L_\ell(\Theta, \Psi) + \mu(t) L_u(\Theta, \Psi) \tag{4}$$

where $L_\ell(\Theta, \Psi)$ and $L_u(\Theta, \Psi)$ are the respective loss functions on the labeled and unlabeled data, and $\mu(t) = \frac{1}{2} - \frac{1}{2} cos(min(\pi, \frac{\pi t}{T}))$ is a warmup function weighing $L_u(\Theta, \Psi)$ at iteration t with total T warmup steps. After training, the supervised classifier $\tilde{h}$ is discarded. Only the model $h \circ f$ is retained for inference.

We apply cross-entropy loss on the labeled data:

$$L_\ell(\Theta, \Psi) = -y_\ell \cdot \log\left(p_\ell\right). \tag{5}$$

Existing robust training techniques can be incorporated. For instance, AdaMatch [1] interpolates between logit g_ℓ and another copy g'_ℓ output with augmented batch normalization statistics to compute $p_\ell = \sigma(\lambda \cdot g_\ell + (1-\lambda) \cdot g'_\ell)$, $\lambda \sim Unif(0, 1)$, to improve robustness to covariate shift.

The unlabeled data is trained with refined pseudo-labels:

$$\begin{aligned} &L_u(\Theta, \Psi) \\ &= -\hat{y}_u^{(PGPR)} \cdot \log(p_{\bar{u}}) \cdot \mathbb{1}[\max_j p_u^{(PGPR)}[j] \geq c_\tau] - \\ &\quad \frac{1}{2} \hat{y}_u^{(PGPR)} \cdot \log(p_{\bar{u}}) \cdot \mathbb{1}[\max_j p_u^{(PGPR)}[j] < c_\tau] \end{aligned} \tag{6}$$

where pseudo-labels with confidence below threshold $c_\tau = \tau \cdot \mathbb{E}(\max_j p_\ell[j])$ have a down-weighted loss. We apply weak and strong augmentations for consistency regularization. In Equation 6, pseudo-labels are estimated on weakly augmented images denoted by u, and the loss is applied on strongly augmented images denoted by $\bar{u}$.

5. Experiments and Results

5.1. Experimental Setups

Methods. We compare the proposed strategy with the supervised baseline S+T and existing SSDA and UniDA methods modified for UniSSDA. **S+T** trains only on labeled source and target samples with cross-entropy loss.

For existing SSDA methods, we evaluate CDAC, PAC and AdaMatch. **CDAC** [21] clusters target data and selects highly-confident target samples for pseudo-labeling and consistency regularization. **PAC** [24] pre-trains with rotation prediction task and encourages label consistency during adaptation. **AdaMatch** [1] builds on FixMatch [37] by augmenting data with weak and strong augmentations, applies random logit interpolation for logit alignment and aligns the overall class distribution between domains.

For existing UniDA methods, we evaluate DANCE and UniOT. **DANCE** [34] uses self-supervision to cluster target samples and uses an entropy threshold to identify samples in common classes for alignment. **UniOT** [2] uses optimal transport to cluster target samples around prototypes. It detects samples in common classes for alignment based on statistical information of class assignment estimates, without the need for prefined threshold values.

For all methods except AdaMatch, we apply domain-specific classifier masks to mask logits of classes absent from the domain. We exclude AdaMatch as it interpolates logits across domains during training, but apply the masks during inference to constrain predictions to classes present in the domain. To add target supervision to UniDA methods, we add cross-entropy loss on the labeled target samples to the original training objectives.

Datasets. We evaluate on 3 popular benchmark datasets for domain adaptation. **Office-Home** [39] has 65 categories of everyday objects in 4 domains: Art (A), Clipart (C), Product (P) and Real World (R). **DomainNet** [30] has a total of 6 domains and 345 classes. Following [21, 24], we evaluate on 4 domains: Clipart (C), Painting (P), Real (R) and Sketch (S). We denote the version with a subset of 126 classes as DomainNet-126 for comparison with SSDA methods [21, 24], and denote the full version as DomainNet-345 for comparison with UniDA methods [7]. Each of Office-Home and DomainNet has a total of 12 source-target domain pairs. **VisDA** [29] has 12 classes and is used to evaluate synthetic-to-real transfer.

For each domain, we randomly split the samples into 50% training, 20% validation, and 30% testing. Following existing practice [21, 24], we assume k-shot target annotation for training and validation, with k set to 3 in main experiments. Instead of pre-specifying a single selection of labeled target samples as in existing works [21, 24], we randomly draw target samples for labeling from the larger collection of training and validation data in each run, in order to take into account the variability of target annotation.

Adaptation settings. We evaluate on the settings listed in Figure 1. For 'Closed-set without label distribution shift', we sample the datasets such that all domains share the same sample size and class distribution, to study covariate shift in isolation. All classes are included in closed-set settings. For the open-set, partial-set and open-partial adaptation settings, we include a subset of classes in the source and/or target domain, as specified in Table 1.

Implementation details. For comparison with SSDA methods, we use ResNet-34 backbone plus a linear classifer following [21, 24]. Models are trained using SGD optimizer with momentum 0.9 and weight decay 0.0005 for 5000 iterations using batch size 24. We set the learning rate

Adaptation Setting	Office-Home		DomainNet-126		DomainNet-345		VisDA	
	Source	Target	Source	Target	Source	Target	Source	Target
Open-set	1-40	1-65	1-80	1-126	1-150	1-345	1-6	1-12
Partial-set	1-65	41-65	1-126	81-126	1-345	1-150	1-12	1-6
Open-partial	1-40	1-20, 41-65	1-80	1-40, 81-126	1-200	1-150, 201-345	1-9	1-6, 10-12

Table 1. Source and target classes in open-set, partial-set and open-partial setting, for Office-Home, DomainNet-126, DomainNet-345 and VisDA. All classes are used in closed-set setting.

Method	Covariate Shift	Covariate + Label Shift				Overall
	Closed-set	Closed-set	Open-set	Partial-set	Open-partial	
Office-Home						
S + T	67.9 ±0.5	65.1 ±0.5	63.2 ±0.3	72.9 ±0.9	64.2 ±0.7	66.7
CDAC	69.7 ±0.4	67.1 ±1.4	60.3 ±0.3	68.7 ±1.7	56.3 ±1.5	64.4
PAC	67.3 ±0.5	65.1 ±0.5	60.6 ±0.4	69.7 ±1.2	61.3 ±0.6	64.8
AdaMatch	69.7 ±0.4	66.6 ±0.5	63.1 ±0.6	74.4 ±0.3	64.1 ±0.3	67.6
Proposed	**72.3 ±1.3**	**69.4 ±0.8**	**66.9 ±1.2**	**77.4 ±1.8**	**67.4 ±1.8**	**70.7**
DomainNet-126						
S + T	63.9 ±0.4	58.8 ±0.2	54.1 ±0.1	72.6 ±0.8	54.4 ±0.7	60.8
CDAC	69.7 ±0.1	65.3 ±0.3	52.1 ±0.9	75.3 ±0.4	43.7 ±1.4	61.2
PAC	69.1 ±0.4	64.6 ±0.2	51.6 ±0.2	77.9 ±0.3	51.2 ±0.8	62.9
AdaMatch	66.7 ±0.1	61.3 ±0.4	53.1 ±0.5	76.3 ±0.5	53.7 ±1.0	62.2
Proposed	**71.8 ±0.7**	**67.3 ±0.6**	**61.2 ±0.8**	**80.3 ±0.4**	**62.5 ±1.3**	**68.6**

Table 2. Comparison with SSDA methods: Target accuracy averaged across 12 domain pairs for each dataset, trained with ResNet-34 backbone.

0.001 for the feature extractor and 0.01 for the classifier, and temperature scaling 0.05. We standardize the set of image augmentations used to random horizontal flips and crops to 224×224 for weak augmentations, with the addition of RandAugment for strong augmentations, following [21].

For comparison with UniDA methods, we follow the training setup in [7] for adapting vision transformer [9] foundation models: dinov2_vitl14 in DINOv2 trained with self-supervision [27] and ViT-L/14@336px in CLIP trained with image-text pairs [31]. The foundation model encoder is frozen, and the classifier is trained for 10000 iterations using batch size 32, with initial learning rate 0.01 and a cosine scheduler with 50 warmup interations. No data augmentation is applied following [31].

Experiments are run on NVIDIA container for PyTorch, release 23.02, on NVIDIA GeForce RTX 3090. We set confidence threshold hyperparameter $\tau = 0.9$ and warmup step count $T = 500$ for our proposed method. Hyperparameters for other methods follow the defaults in [1, 7, 21, 24]. Experiments are run over 3 seeds, and we report the average ± standard deviation of the target domain accuracy.

5.2. Results

5.2.1 Comparison with SSDA Methods

Table 2 shows the target accuracy averaged across all domain pairs for each dataset. The existing SSDA methods evaluated generally improve over the supervised baseline S+T in the closed-set settings. In the more challenging label space shift settings, CDAC and PAC do not consistently improve performance in the partial-set setting, and degrade performance in the open-set and open-partial settings. Moreover, CDAC learning can be unstable, leading to large performance drops (above 7%) in open-partial setting. AdaMatch improves performance in the partial-set setting for both datasets, but still marginally degrades performance in the open-set and open-partial setting as a result of common-class bias.

Method	Closed-set	Open-set	Partial-set	Open-partial	Overall
DomainNet-345					
S + T	73.6 ±0.2	68.2 ±0.4	80.9 ±0.4	70.3 ±0.5	73.2
DANCE	73.8 ±0.3	67.1 ±0.5	81.4 ±0.5	69.3 ±0.5	72.9
UniOT	72.8 ±0.3	62.3 ±0.5	76.2 ±0.5	64.3 ±0.4	68.9
Proposed	**74.4 ±0.3**	**69.0 ±0.4**	**82.3 ±0.3**	**71.5 ±0.5**	**74.3**
VisDA					
S + T	78.0 ±0.6	73.2 ±1.5	91.0 ±0.3	79.9 ±1.2	80.5
DANCE	75.1 ±0.8	67.0 ±1.2	93.6 ±0.3	78.9 ±2.9	78.6
UniOT	77.9 ±1.3	68.7 ±2.2	87.6 ±0.7	78.4 ±0.7	78.2
Proposed	**79.8 ±0.8**	**76.5 ±1.5**	**93.7 ±0.7**	**82.6 ±0.7**	**83.2**

(a) DINOv2 encoder dinov2_vitl14

Method	Closed-set	Open-set	Partial-set	Open-partial	Overall
DomainNet-345					
S + T	77.3 ±0.4	68.9 ±0.5	84.1 ±0.4	71.1 ±0.6	75.4
DANCE	77.0 ±0.3	67.5 ±0.6	84.1 ±0.3	69.6 ±0.5	74.6
UniOT	77.4 ±0.3	61.8 ±0.7	81.6 ±0.4	64.7 ±0.5	71.4
Proposed	**77.5 ±0.3**	**71.1 ±0.7**	**84.4 ±0.3**	**73.7 ±0.6**	**76.7**
VisDA					
S + T	87.5 ±0.4	78.8 ±1.1	95.0 ±0.1	84.2 ±0.8	86.4
DANCE	86.2 ±0.8	74.7 ±1.3	**96.3 ±0.4**	82.5 ±1.9	84.9
UniOT	87.6 ±1.1	79.7 ±0.8	92.1 ±1.1	84.3 ±0.8	85.9
Proposed	**88.0 ±0.1**	**83.3 ±1.5**	96.2 ±0.4	**84.6 ±0.6**	**88.0**

(b) CLIP encoder ViT-L/14@336px

Table 3. Comparison with UniDA methods: Target accuracy averaged across 12 domain pairs for DomainNet-345 and synthetic-to-real transfer accuracy for VisDA, in covariate + label shift settings. Training is performed with frozen foundation model encoder and learnable classifier.

Amongst the SSDA methods evaluated, overall, AdaMatch shows the least performance degradation in non-closed settings. Consequently, we adopt random logit interpolation from AdaMatch for robust training of the labeled samples in Equation 5. We simulate covariate shifts by passing the input images though the feature extractor f with different batch normalization statistics i.e. statistics from both labeled and unlabeled data and from labeled data alone. We randomly interpolate between the two versions of logits before computing the predictive probability and training with cross-entropy loss.

The proposed method attains the highest overall accuracy, and outperforms the second-best method by 3.1% and 5.7% on Office-Home and DomainNet-126, respectively. It consistently outperforms the supervised baseline S+T, and increases accuracy by over 7% in the challenging open-set and open-partial setting for DomainNet-126.

5.2.2 Comparison with UniDA Methods

Table 3 shows the target accuracy evaluated on two ViT encoders. The existing UniDA methods evaluated performs similarly or underperforms the supervised baseline S+T in most cases. Existing criteria (e.g. high entropy) to identify target private class samples in UniDA become less suitable in UniSSDA, as a small number of target private class samples are now available for supervision. We observe that performances of all methods are generally higher on the CLIP encoder than on the DINOv2 encoder, as the latter is trained only with self-supervised objectives. The proposed method attains the highest overall accuracy on both datasets and foundation model encoders. On DomainNet-345, the peformance gain over the second-best method is 1.1% and 1.3% with DINOv2 and CLIP, respectively. On VisDA, the performance gain is 2.7% and 1.6% with DINOv2 and CLIP, repectively. We note that no special robust training is implemented on the labeled samples to learn robust features as the encoders are frozen. However due to the stronger and more robust feature extraction capability of foundation models, the frozen encoders can more adequately address covariate shift compared to the ResNet-34 in Section 5.2.1.

6. Further Analysis

The proposed method improves private class accuracy without having to sacrifice common class accuracy. We study adaptation accuracy on samples in common and target private classes separately in open-set and open-partial settings on DomainNet-345 in Table **??**. We observe that the UniDA method UniOT achieves the highest common class accuracy in 5 out of 8 cases, but at the expense of private class accuracy. Comparing methods with similar common class accuracy, the proposed method improves private class accuracy in most cases. We include comparisons with SSDA methods in the Appendix.

The proposed method is effective in scenarios where not all target classes are labeled. We study these scenarios on DomainNet-345 in Table 5. In 'Unlabeled Private', 'Unlabeled Common' and 'Unlabeled Mixed', we set 45 target private classes (class 301-345), 50 common classes (class 101-150) and a mixture of 95 classes (class 301-345, 101-150) as entirely unlabeled, respectively. Following UniDA practice, we treat unlabeled private classes as a single 'unknown' class. For all non-UniDA methods, we adopt the one-vs-all classifier from OVANet [32] to detect 'unknown' samples. Pre-training method PAC is excluded since the encoder is frozen. We include recent methods for long-tailed learning (DRw [28]), SSDA (ProML [12]) and UniDA (OVANet [32]). Our method attains the best performance in these challenging scenarios.

Method	Open-set (Common / Pvt)	Open-partial (Common / Pvt)	Open-set (Common / Pvt)	Open-partial (Common / Pvt)
	DomainNet-345 (CLIP)		**VisDA (CLIP)**	
S + T	81.5 / 60.0	81.9 / 61.4	92.7 / 65.0	92.7 / **67.2**
DANCE	81.1 / 57.8	81.2 / 59.0	91.0 / 58.4	92.7 / 62.2
UniOT	**83.0** / 46.7	**82.3** / 48.7	**94.9** / 64.5	93.5 / 66.1
Proposed	81.2 / **64.0**	**82.3** / **66.0**	94.1 / **72.4**	**93.9** / 65.8

Table 4. Average target accuracy on common and target private (Ptv) classes.

Method	Open-partial	Unl. Private	Unl. Common	Unl. Mixed
S + T	71.1 ±0.6	65.5 ±0.7	70.5 ±0.6	64.1 ±0.8
DRw	69.2 ±0.7	65.0 ±0.8	62.1 ±0.5	59.0 ±0.9
CDAC	69.3 ±0.6	64.6 ±0.6	67.3 ±0.5	63.0 ±0.6
AdaMatch	68.6 ±0.6	65.1 ±0.7	68.1 ±0.6	64.1 ±0.7
ProML	71.1 ±0.6	66.1 ±0.7	70.5 ±0.5	64.9 ±0.8
DANCE	69.6 ±0.5	66.0 ±0.8	71.5 ±0.7	64.6 ±0.8
UniOT	64.7 ±0.5	58.8 ±0.4	64.9 ±0.6	58.9 ±0.5
OVANet	71.2 ±0.6	66.1 ±0.7	70.5 ±0.5	64.9 ±0.8
Proposed	**73.7** ±**0.6**	**67.1** ±**0.6**	**72.5** ±**0.5**	**66.4** ±**0.6**

Table 5. Comparison with long-tailed SSL, SSDA and UniDA methods: Accuracy averaged across 12 domain pairs for DomainNet-345 with CLIP encoder. In Unlabeled (Unl.) Private / Common / Mixed scenarios, several target private / common / mixture of private and common classes are entirely unlabeled.

Each pseudo-label refinement step is effective in improving pseudo-label quality. We perform the ablation study on DomainNet-126 $C \rightarrow P$ across all UniSSDA settings. Note that the group reweighted refinement step does not affect adaptation in closed-set settings since all classes belong to the common class group. From Table 6, overall, each refinement step helps to improve target accuracy.

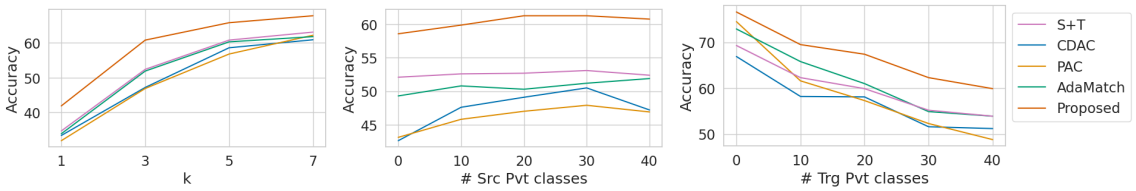
The refined pseudo-labels can be readily incorporated into existing SSDA methods to expand their adaptation capabilities to non-closed-set settings. In existing SSDA methods, we add a supervised classification head on top of the feature extractor to estimate prior distributions for pseduo-label refinement. We replace the original pseudo-labels used with our refined pseudo-labels while retaining all other components of the algorithms. From Table 7, we see up to 3.4%, 3.9% and 7.8% improvement in open-set, partial-set and open-partial setting, respectively.

The proposed method is effective with varying amounts of target annotation. We vary $k \in \{1, 3, 5, 7\}$ for k-shot target annotation in Figure 6a. We focus on the most challenging open-partial setting. On DomainNet-126 $C \rightarrow P$, the proposed method outperforms SSDA methods by more than 4% on all values of k.

The proposed method is effective under varying degrees of 'partialness' and 'openness'. We experiment with different number of source or target private classes on DomainNet-126 $C \rightarrow P$ to vary the degree of 'partialness'

Group reweighted	Classifier aggregated	Covariate Shift	Covariate + Label Shift				Overall
		Closed-set	Closed-set	Open-set	Partial-set	Open-partial	
✗	✗	66.1	61.6	50.4	**78.7**	51.5	61.7
✓	✗	66.1	61.6	56.8	**78.7**	59.2	64.5
✓	✓	**66.9**	**63.7**	**57.4**	78.5	**60.8**	**65.5**

Table 6. Ablation study on effectivess of pseudo-label refinement steps, evaluated on DomainNet-126 $C \rightarrow P$.



(a) Vary annotation (b) Vary 'partialness' (c) Vary 'openness'

Figure 6. Further experiments on DomainNet-126 $C \rightarrow P$ in open-partial setting. (a) plots target accuracy with different k for k-shot target annotation. (b) and (c) plot target accuracy under varying degrees of 'partialness' and 'openness', respectively.

Method	Open-set	Partial-set	Open-partial
CDAC	50.0	74.8	47.2
+ PGPR	49.9 (↓ 0.1)	74.9 (↑ 0.1)	55.0 (↑ 7.8)
PAC	46.8	75.2	46.9
+ PGPR	49.2 (↑ 2.4)	77.7 (↑ 2.5)	53.5 (↑ 6.6)
AdaMatch	50.2	73.9	51.9
+ PGPR	53.6 (↑ 3.4)	77.8 (↑ 3.9)	58.5 (↑ 6.6)

Table 7. Proposed prior-guided pseudo-label refinement can be incorporated into existing SSDA methods to expand their capabilities to non-closed-set settings. Values in green/red denote increase/decrease over performance of the original method, evaluated on DomainNet-126 $C \rightarrow P$.

or 'openness' in the open-partial setting. In Figure 6b, we start with 40 common classes (class 1-40), 46 target private classes (class 81-126) and no source private class, and increase the number of source private classes from 0 to 40 by increments of 10 (class 41-80). The proposed method outperforms SSDA methods by more than 6%. We observe that target accuracy tends to slightly increase when source private classes are initially added till 30 classes and then decrease. Source private class samples may help feature learning, but since these classes are irrelevant to the target domain, an abundance of these samples may distract the model from learning target-relevant features.

In Figure 6c, we start with 40 common classes (class 1-40), 40 source private classes (class 41-80) and no target private class, and increase the number of target private classes from 0 to 40 by increments of 10 (class 81-120). Target accuracy expectedly decreases as the number of target classes increases. The advantage of the proposed method is more evident under higher degree of 'openness'. The proposed method outperforms all other methods by 2.1% at no target private classes, and 6% at 40 target private classes.

The proposed method can be applied effectively on different model backbones. We additionally compare against SSDA methods on other backbone architectures including

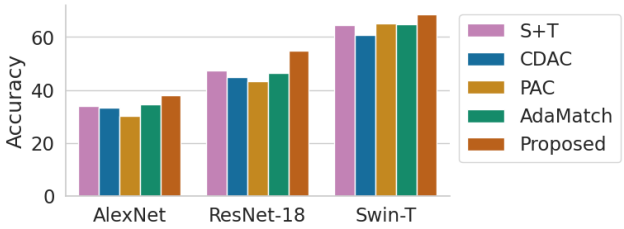


Figure 7. Further comparison with existing SSDA methods using different model backbones, evaluated on DomainNet-126 $C \rightarrow P$ in open-partial setting.

AlexNet, ResNet-18 and Swin-T (Tiny version of Swin Transformer [23]). From Figure 7, in open-partial setting on DomainNet-126 $C \rightarrow P$, the proposed method outperforms the second-best method by 3.9%, 7.5% and 3.2% on AlexNet, ResNet-18 and Swin-T, respectively.

7. Conclusion

This work introduces Universal SSDA, a generalized semi-supervised domain adaptation problem covering diverse and practical types of covariate and label shifts. We find that existing SSDA and UniDA methods are susceptible to common-class bias and do not consistently perform well in UniSSDA settings. We propose a new prior-guided pseudo-label refinement strategy to address the reinforcement of common-class bias during label propagation. The proposed strategy is simple to implement and effective, as demonstrated through evaluations on multiple datasets and models. We propose the approach as a baseline for further research and application in this area.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

References

[1] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *ICLR*, 2022. 2, 3, 5, 6

[2] Wanxing Chang, Ye Shi, Hoang Duong Tuan, and Jingya Wang. Unified optimal transport framework for universal domain adaptation. In *NeurIPS*, 2022. 3, 5

[3] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. In *NeurIPS*, 2022. 3

[4] Liang Chen, Qianjin Du, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Mutual nearest neighbor contrast and hybrid prototype self-training for universal domain adaptation. In *AAAI*, 2022. 1, 3

[5] Li Cheng and Sinno Jialin Pan. Semi-supervised domain adaptation on manifolds. *IEEE Transactions on Neural Networks and Learning Systems*, 25(12):2240–2249, 2014. 2

[6] Hal Daumé III, Abhishek Kumar, and Avishek Saha. Frustratingly easy semi-supervised domain adaptation. In *Workshop on Domain Adaptation for Natural Language Processing*, 2010. 2

[7] Bin Deng and Kui Jia. Universal domain adaptation from foundation models. *arXiv*, 2023. 3, 5, 6

[8] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, 2013. 3

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6

[10] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *ECCV*, 2020. 3

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35, 2016. 1, 2, 3

[12] Xinyang Huang, Chuang Zhu, and Wenkai Chen. Semi-supervised domain adaptation via prototype-based multi-level learning. In *IJCAI*, 2023. 7

[13] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, 2020. 3

[14] Ju Hyun Kim, Ba Hung Ngo, Jae Hyeon Park, Jung Eun Kwon, Ho Sub Lee, and Sung In Cho. Distilling and refining domain-specific knowledge for semi-supervised domain adaptation. In *BMVC*, 2022. 2

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

[16] Abhishek Kumar, Avishek Saha, and Hal Daume. Co-regularization based semi-supervised domain adaptation. In *NIPS*, 2010. 2

[17] Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking distributional matching based domain adaptation. *arXiv*, 2020. 1

[18] Bo Li, Yezhen Wang, Shanghang Zhang, Dongsheng Li, Kurt Keutzer, Trevor Darrell, and Han Zhao. Learning invariant representations and risks for semi-supervised domain adaptation. In *CVPR*, 2021. 2

[19] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *ECCV*, 2020. 3

[20] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *CVPR*, 2021. 3

[21] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *CVPR*, 2021. 2, 3, 5, 6

[22] Kai Li, Chang Liu, Handong Zhao, Yulun Zhang, and Yun Fu. ECACL: A holistic framework for semi-supervised domain adaptation. In *ICCV*, 2021. 2, 3

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 8

[24] Samarth Mishra, Kate Saenko, and Venkatesh Saligrama. Surprisingly simple semi-supervised domain adaptation with pretraining and consistency. *BMVC*, 2021. 2, 3, 5, 6

[25] Ba Hung Ngo, Ju Hyun Kim, Yeon Jeong Chae, and Sung In Cho. Multi-view collaborative learning for semi-supervised domain adaptation. *IEEE Access*, 9:166488–166501, 2021. 2

[26] Kriti Ohri and Mukesh Kumar. Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems*, 224:107090, 2021. 3

[27] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv*, 2023. 6

[28] Hanyu Peng, Weiguo Pian, Mingming Sun, and Ping Li. Dynamic re-weighting for long-tailed semi-supervised learning. In *WACV*, 2023. 7

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

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

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6

[32] Kuniaki Saito and Kate Saenko. OVANet: One-vs-all network for universal domain adaptation. In *ICCV*, 2021. 1, 3, 7

[33] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. *ICCV*, 2019. 3

[34] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *NeurIPS*, 2020. 3, 5

[35] Ankit Singh. CLDA: Contrastive learning for semi-supervised domain adaptation. *NeurIPS*, 2021. 2, 3

[36] Anurag Singh, Naren Doraiswamy, Sawa Takamuku, Megh Bhalerao, Titir Dutta, Soma Biswas, Aditya Chepuri, Balasubramanian Vengatesan, and Naotake Natori. Improving semi-supervised domain adaptation using effective target selection and semantics. In *CVPRW*, 2021. 3

[37] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 5

[38] Baocheng Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. *ECCV Workshops*, 2016. 1, 2

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

[40] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 1, 2

[41] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. In *IJCAI*, 2022. 2, 3

[42] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q. Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. *ICCV*, 2020. 2, 3

[43] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2022. 3

[44] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015. 2, 3

[45] Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *WACV*, 2022. 2, 3

[46] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *CVPR*, 2019. 1, 3

[47] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *NeurIPS*, 2018. 1

[48] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019. 1

[49] Didi Zhu, Yinchuan Li, Junkun Yuan, Zexi Li, Kun Kuang, and Chao Wu. Universal domain adaptation via compressive attention matching. In *ICCV*, 2023. 1, 3