Universal Semi-supervised Model Adaptation via Collaborative Consistency Training

Zizheng Yan[1,2*] Yushuang Wu[1,2*] Yipeng Qin[3] Xiaoguang Han[1,2]
Shuguang Cui[1,2] Guanbin Li[4,5†]

[1]FNii, CUHKSZ [2]SSE, CUHKSZ [3]Cardiff University [4]Sun Yat-sen University
[5]Research Institute, Sun Yat-sen University, Shenzhen

Abstract

In this paper, we introduce a realistic and challenging domain adaptation problem called Universal Semi-supervised Model Adaptation (USMA), which i) requires only a pre-trained source model, ii) allows the source and target domain to have different label sets, i.e., they share a common label set and hold their own private label set, and iii) requires only a few labeled samples in each class of the target domain. To address USMA, we propose a collaborative consistency training framework that regularizes the prediction consistency between two models, i.e., a pre-trained source model and its variant pre-trained with target data only, and combines their complementary strengths to learn a more powerful model. The rationale of our framework stems from the observation that the source model performs better on common categories than the target-only model, while on target-private categories, the target-only model performs better. We also propose a two-perspective, i.e., sample-wise and class-wise, consistency regularization to improve the training. Experimental results demonstrate the effectiveness of our method on several benchmark datasets.

1. Introduction

Deep neural networks have achieved remarkable progress in various tasks, such as image recognition [14], machine translation [2], biomedical imaging [40], etc. However, training a good neural network model remains challenging as it requires huge amounts of labeled data that are expensive to annotate. To this end, unsupervised domain adaptation (UDA) methods [13, 48, 55] were proposed to train neural network models without annotated data by transferring knowledge learned from a label-rich source domain to the unlabeled target domain. However, due to the gap between the source and target domains, it is still challenging for UDA methods to achieve good performance on the target domain. To bridge such a domain gap, Semi-supervised Domain Adaptation (SSDA) [25, 43] incorporates a few labeled target samples into the training and thus significantly outperforms UDA, showing great potential for applications.

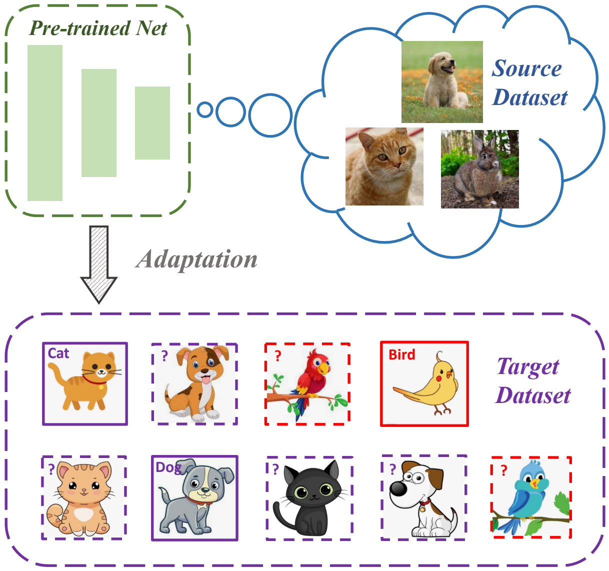


Figure 1. Illustration of Universal Semi-supervised Model Adaptation (USMA). In this example, we begin with a source model that has been pre-trained on a real-world domain dataset containing categories ”Cat”, ”Dog”, and ”Rabbit”. The target domain consists of a cartoon dataset with different categories, ”Cat”, ”Dog”, and ”Bird”, each containing both labeled (Solid boxes) and unlabeled samples (Dashed boxes).

Although promising, existing SSDA methods usually assume that the source data is available during training, which is impractical in many real-world scenarios where restrictions apply, *e.g.*, data privacy and limited storage [28]. To

[*] Equal contributions.
[†] Corresponding Author.

meet such new demands, a new research topic, namely the model adaptation [28, 32, 33, 50, 57], has recently been proposed with the aim of transferring knowledge from a pre-trained source model rather than the source data. To simplify the problem, most of these works assume that the source and target domain share the same label set.

However, the above simplification significantly limits the practical application of those methods due to the prevalence of category gaps between the pre-trained source models and the target data. For example, when adapting a product retrieval model pre-trained on a web dataset to real-world ones collected from shelves of different supermarkets, there are likely different subsets of common categories between the pre-trained source model and each real-world dataset. This issue is addressed by universal domain adaptation [60].

In this paper, we propose a new problem, called *Universal Semi-supervised Model Adaptation* (USMA), which covers all aforementioned problems. USMA presents a more realistic and, consequently, more challenging scenario. An Illustration of USMA is shown in Fig. 1. A naive solution for USMA is to apply a semi-supervised learning method to fine-tune the pre-trained source model directly with the target data. However, we observed that such a naive solution is ineffective as the source model's extensive knowledge in the source domain impedes its learning in the target domain: it works better for common categories shared by both the source and target domains, than the target-private categories (*i.e.*, categories in the target but not in the source domain). This is justified by the opposite performance of a reference model, *i.e.*, a model with the same architecture of the source model but pre-trained using only the target data in a self-supervised way (Fig. 2). Despite such a disappointment, the complementary strengths of the two models caught our attention: can we integrate the appeal of the two to train a more powerful model that performs well on both common and target-private categories?

To answer the above question, we propose a collaborative consistency training (CCT) framework which extends the vanilla consistency regularization applicable to a single model [47, 53] to our double-model case. Specifically, we propose to add additional regularization across the two models, thereby allowing each model to exploit the strengths of the other model to resolve its shortcomings. To take full advantage of the proposed CCT framework, we further propose to incorporate consistency regularization from two perspectives, *i.e.*, sample-wise and class-wise. Sample-wise, we first augment each training sample into two views and employ pseudo labeling to enforce consistency of high-confidence predictions between not only the two views of a single model but also those across two different models. Class-wise, we propose a single loss function that incorporates both the *class consistency prior* (*i.e.*, same prediction across views) and the *class sparsity prior* (*i.e.*, sparsity of prediction vectors) into training.

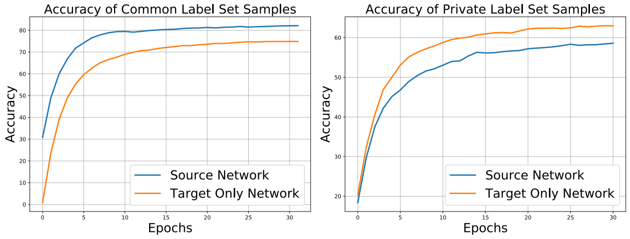


Figure 2. Left: The classification accuracy of samples from common label set. Right: The classification accuracy of the samples from private label set. The blue curve shows the accuracy of the network fine-tuning from the source pre-trained network while the orange curve shows the accuracy of the network fine-tuning from a network pre-trained with only target data through self-supervised learning.

Interestingly, we observed that the performance of the two models converge to a similar point after training. Therefore, without loss of generality, we choose the adapted source model as the final model for reference. Our contributions include:

- We define Universal Semi-supervised Model Adaptation (USMA), a more realistic and challenging domain adaptation problem to be solved.
- We propose a collaborative consistency training (CCT) framework that leverages the complementary strengths of a source-pretrained and target-pretrained model to produce a more powerful one.
- We propose a two-perspective approach (sample-wise and class-wise) for CCT.
- Extensive experimental results demonstrate the effectiveness of the proposed methods.

2. Related Work

Semi-supervised Domain Adaptation. Semi-supervised Domain Adaptation (SSDA) assumes few labeled samples per category in the target domain [4, 15, 18, 22, 23, 25, 26, 29, 38, 43, 51, 54, 56], which yields cost-effective applications that require little labeling efforts. Most of those works address the SSDA problem from two perspectives: (i) domain alignment and (ii) self-training on target domain, *i.e.*, entropy minimization, pseudo labeling. Saito *et al.* [43] first introduce the SSDA problem and propose to minimize entropy for feature extractor and align the source and target domain by maximizing entropy for the classifier. UODA [38] proposes to minimize the target entropy so that the target feature can be compact, and maximize the source entropy so that the source feature can be scattered, making alignment easier. CDAC [25] proposes an adaptive clustering method to perform domain alignment and uses pseudo labeling for self-training. ECACL [26] proposes to reduce the domain gap by matching source and target prototypes.

AdaMatch [4] proposes to match the class distribution between source and target. Compared with the vanilla SSDA, our proposed USMA is more challenging as it has no access to source data, which is infeasible to perform domain alignment, and also needs to adapt models to a target domain that has a different label set than the source domain.

Model Adaptation. Model adaptation (MA) was proposed to implement domain adaptation without access to the source data, thereby addressing the dilemma of data sharing versus data privacy [28] in traditional domain adaptation. Many works have been proposed to address the problem of MA [8, 11, 19–21, 27, 28, 30, 33, 39, 41, 50, 52, 57–59], and the methods can be roughly categorized into two streams: generative [27, 59] and discriminative [28, 52, 57]. Generative methods usually model the generation of labeled images or features. 3C-GAN [27] proposes to synthesize the target-style labeled training images via conditional GAN. SoFA [59] proposes to generate reliable latent features for domain alignment. On the other hand, discriminative methods usually require to fix the source classifier and fine-tune the backbone. SHOT [28, 30] proposes to minimize the target entropy while maximizing the mutual information. AANet [52] proposes to incorporate a trainable classifier with the fixed classifier to jointly perform adaptation. Though achieving good performance in unsupervised or semi-supervised settings, these MA methods work under the assumption that the source and target domains share the same label set, which is not the case in USMA.

Universal Domain Adaptation. Universal DA [6, 10, 12, 24, 31, 34, 44, 45, 60, 61] resolves the class mismatch issue by combining Open-set DA [46] and Partial DA [5]. Open-set DA excludes some target domain categories (a.k.a. target-private categories) not in the source domain, while Partial DA excludes some source domain categories (a.k.a. source-private categories) not in the target domain. Most works focus on designing the criterion to reject target-private samples. UAN [60] proposes to use entropy as the criterion, *e.g.*, reject high entropy samples and exclude them from domain alignment. Fu *et al.* [12] propose the joint use of entropy, confidence and classifier consistency as the criterion. DANCE [44] also uses entropy as the criterion and proposes an entropy separation loss to reject the target-private samples. OVANet [45] trains one-vs-all classifiers to learn the distance between the source positive and nearest negative classes and use such distance as the criterion. Instead of rejecting open-set samples as in universal DA, USMA aims to classify them correctly in a semi-supervised way given few labeled samples in the target-private categories, which is more challenging.

3. Methodology

In this section, we first present the definition of USMA, and then detail the proposed collaborative consistency training framework. Finally, we propose two novel loss functions that regularize collaborative consistency in a sample-wise and class-wise manner respectively. Our framework is illustrated in Fig. 3.

3.1. Problem Definition

Universal Semi-supervised Model Adaptation (USMA) aims to learn a model F that achieves high prediction accuracy on target domain data using only F_s and $\mathcal{D}_t$, where

- F_s denotes a source model pre-trained with the source data $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}, \mathbf{y}_i^s \in \mathcal{C}_s$ sampled from a different distribution than the target domain,
- $\mathcal{D}_t = \mathcal{D}_t^l \cup \mathcal{D}_t^u = \{(\mathbf{x}_i^{tl}, \mathbf{y}_i^{tl})\}_{i=1}^{n_{tl}} \cup \{(\mathbf{x}_j^{tu})\}_{j=1}^{n_{tu}}, \mathbf{y}_i^{tl} \in \mathcal{C}_t$ denotes a target dataset consisting of n_{tl} labeled and n_{tu} unlabeled samples ($n_{tl} \ll n_{tu}$),
- Label sets $\mathcal{C}_s \neq \mathcal{C}_t$.

To facilitate discussion, we denote $\mathcal{C} = \mathcal{C}_s \cap \mathcal{C}_t$ as the common label set, and have $\bar{\mathcal{C}}_s = \mathcal{C}_s \backslash \mathcal{C}$ and $\bar{\mathcal{C}}_t = \mathcal{C}_t \backslash \mathcal{C}$ that denote the label sets of private categories of source and target domains respectively. Following [60], we define the commonness of label sets ξ as $\frac{|\mathcal{C}_s \cap \mathcal{C}_t|}{|\mathcal{C}_s \cup \mathcal{C}_t|}$.

3.2. Collaborative Consistency Training

As Fig. 3 shows, the key insight of our framework is that the final model $F(\cdot|\theta)$ can be derived from the collaborative training of $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$ that enforces consistent predictions in-between, where $F(\cdot|\theta_s) = F_s$ is the model pre-trained with source data (*a.k.a.* source model) and $F(\cdot|\theta_t)$ is the target model pre-trained with $\mathcal{D}_t$ in a self-supervised learning manner (*a.k.a.* target only model). Both $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$ share the same network architecture but have different parameters θ_s and θ_t. Specifically, $F(\cdot|\theta)$ consists of a backbone and a classifier. The backbones of $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$ are pretrained with different data, while the classifiers are both randomly initialized. The output dimension of the classifiers of $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$ is $|\mathcal{C}_t|$. The rationale of our approach stems from the observation (Fig. 2) that: i) for samples with common labels $\mathcal{C}$, despite the domain gaps, $F(\cdot|\theta_s)$ outperforms $F(\cdot|\theta_t)$ in prediction accuracy from its rich prior knowledge in $\mathcal{C}$; ii) while for samples with target private labels $\bar{\mathcal{C}}_t$, $F(\cdot|\theta_t)$ achieves a higher prediction accuracy than $F(\cdot|\theta_s)$. Such an observation implies that $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$ learn complementary information of $\mathcal{C}$ and $\bar{\mathcal{C}}_t$ ($\mathcal{C}_t = \mathcal{C} \cup \bar{\mathcal{C}}_t$) in the target domain respectively.

To harness this, we introduce a Collaborative Consistency Training (CCT) framework. This involves creating dual views of each sample in the unlabeled target data $\mathcal{D}_t^u$ through augmentations. Unlike traditional consistency regularization, which enforces internal consistency within a single model, we introduce *cross consistency*. This aligns predictions between the two models $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$, enabling them to learn more discriminative features for both

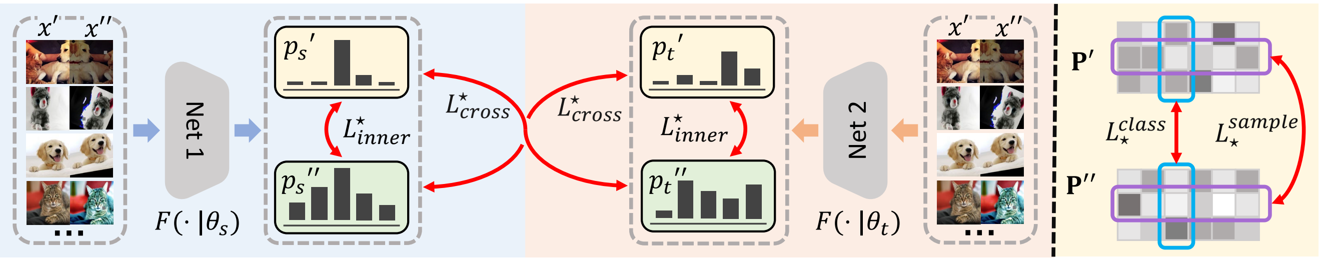


Figure 3. Illustration of the proposed collaborative consistency training (CCT) framework. Left: the overall pipeline of CCT. Right: The detailed consistency loss of CCT. The backbone of $F(\cdot|\theta_s)$ is initialized with the source-pretrained model, while the backbone of $F(\cdot|\theta_t)$ is initialized with the self-supervised pre-trained model using target data. $\mathbf{P}', \mathbf{P}'' \in \mathbb{R}^{N\times C}$ are the batch predictions of two views, where N is the batch size and C is the number of classes.

common and private classes. We further enrich CCT by implementing both inner and cross consistency from sample-wise and class-wise perspectives. Subsequent sections detail these regularization techniques.

3.3. Sample-wise Consistency Regularization

For each unlabeled target sample $\mathbf{x}_i \in \mathcal{D}_t^u$, we apply two different augmentations to it and get its two views $\mathbf{x}'_i$ and $\mathbf{x}''_i$, respectively.

Sample-wise Inner Consistency Regularization. We follow the vanilla consistency regularization in SSL [47, 53] and use the pseudo labels generated by one sample view $\mathbf{x}'_i$ to supervise the training of the other sample view $\mathbf{x}''_i$. For each sample, the loss function can be formulated as:

$$\begin{aligned}\mathcal{L}_{sample}^{inner} = &-\sum_{i=1}^{C} \mathbb{1}(F(\mathbf{x}'|\theta_s)_i \geq \tau)\log(F(\mathbf{x}''|\theta_s)_i)\\ &-\sum_{i=1}^{C} \mathbb{1}(F(\mathbf{x}'|\theta_t)_i \geq \tau)\log(F(\mathbf{x}''|\theta_t)_i),\end{aligned} \tag{1}$$

where $C = |\mathcal{C}_t|$ is the number of classes in the target domain, τ is the threshold. As indicated in [53], $\mathcal{L}_{sample}^{inner}$ regularizes both networks to be invariant to input noises (*i.e.*, different views) and makes them more robust.

Sample-wise Cross Consistency Regularization. While in USMA, the aforementioned inner consistency regularization is not sufficient as it suffers from the poor performance of the source model $F(\cdot|\theta_s)$ on target private categories $\bar{\mathcal{C}}_t$. Our sample-wise cross consistency regularization addresses this issue by collaboratively training $F(\cdot|\theta_s)$ and the target model $F(\cdot|\theta_t)$ together:

$$\begin{aligned}\mathcal{L}_{sample}^{cross} = &-\sum_{i=1}^{C} \mathbb{1}(F(\mathbf{x}'|\theta_s)_i \geq \tau)\log(F(\mathbf{x}''|\theta_t)_i)\\ &-\sum_{i=1}^{C} \mathbb{1}(F(\mathbf{x}'|\theta_t)_i \geq \tau)\log(F(\mathbf{x}''|\theta_s)_i),\end{aligned} \tag{2}$$

where the highly-confident predictions of $F(\mathbf{x}'|\theta_t)$ are used as pseudo labels of $F(\mathbf{x}''|\theta_s)$ and the highly-confident predictions of $F(\mathbf{x}'|\theta_s)$ (mostly samples of $\mathcal{C}$) are used as pseudo labels of $F(\mathbf{x}''|\theta_t)$.

Sample-wise Consistency Loss. Combining the inner and cross consistency loss, we have our final sample-wise consistency loss as:

$$\mathcal{L}_{sample} = \frac{1}{2}(\mathcal{L}_{sample}^{inner} + \mathcal{L}_{sample}^{cross}). \tag{3}$$

Remark. The above objective indicates that each sample receives supervision from *two pseudo labels*. Since the two networks have learned complementary information, we argue that it is highly likely that at least one of them is correct. Furthermore, as indicated by [1] that DNNs tend to fit clean labels before noisy ones, our networks will prioritize the fitting of such correct pseudo labels. This helps our networks learn better representations and provides more reliable pseudo labels for subsequent training steps, thus forming a virtuous circle of training.

3.4. Class-wise Consistency Regularization

Being orthogonal to our sample-wise consistency regularization, our class-wise consistency regularization facilitates the proposed CCT through the incorporation of *class consistency prior* and *class sparsity prior* that takes the relation among samples in a batch into consideration. Similar to the above section, we denote $\mathbf{x}'_i$ and $\mathbf{x}''_i$ as the two views of an unlabeled target sample $\mathbf{x}_i \in \mathcal{D}_t^u$ and have: 1) **Class Consistency Prior.** The predictions of $\mathbf{x}'_i$ and $\mathbf{x}''_i$ should be the same over all classes. 2) **Class Sparsity Prior.** The predictions of $\mathbf{x}'_i$ and $\mathbf{x}''_i$ should be unit vectors with a single 1.0 for one class and 0 for all the other classes.

Class-wise Inner Consistency Regularization. Suppose $\mathbf{P}', \mathbf{P}'' \in \mathbb{R}^{n\times C}$ are the batch predictions of $F(\mathbf{x}'|\theta_s)$ and $F(\mathbf{x}''|\theta_s)$ respectively, where n is the batch size, C is the number of classes, and $\mathbf{P}_{(j,k)}$ represents the softmax confidence of classifying the jth sample into the kth class, we compute the correlation matrix of $\mathbf{P}'$ and $\mathbf{P}''$ as:

$$\mathbf{R} = \mathbf{P}'^{\top}\mathbf{P}''. \tag{4}$$

It can be observed that $\mathbf{R}_{(j,k)}$ represents the inner product similarity between the j-th and k-th column of $\mathbf{P}'$ and $\mathbf{P}''$, respectively. Such a cross-correlation matrix $\mathbf{R}$ allows us to incorporate i) the **class consistency prior** by maximizing the similarity between the same classes (diagonal values) and ii) the **class sparsity prior** by minimizing the similarity between different classes (off-diagonal values). [1]

Since $\mathbf{R}$ is asymmetric and the row summations are different, we convert it to be symmetric and normalize its rows and columns:

$$\hat{\mathbf{R}} = \phi\big((\mathbf{R} + \mathbf{R}^\top)/2\big), \tag{5}$$

where $\phi(\cdot)$ represents a row normalization where each element is divided by the row sum. With $\hat{\mathbf{R}}$, we can formulate our class-wise inner consistency loss as:

$$\mathcal{L}_{class}^{inner} = -\frac{1}{2C}\big(\operatorname{Tr}(\hat{\mathbf{R}}_s) + \operatorname{Tr}(\hat{\mathbf{R}}_t)\big), \tag{6}$$

where $\operatorname{Tr}(\cdot)$ is the trace of a matrix, $\hat{\mathbf{R}}_s$ and $\hat{\mathbf{R}}_t$ are the computed matrices from $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$, respectively. With the normalization in Eq. 5 (*i.e.*, the sum of values in each row or column equals to 1), minimizing the above loss function not only maximizes the diagonal values but also minimizes the off-diagonal values, which incorporates the proposed two priors simultaneously.

Class-wise Cross Consistency Regularization. We extend the above inner consistency regularization to a cross consistency one as follows. Let $\mathbf{P}'_1, \mathbf{P}''_2 \in \mathbb{R}^{n\times C}$ be the batch predictions of $F(\mathbf{x}'|\theta_s)$ and $F(\mathbf{x}''|\theta_t)$ respectively, we compute the cross-correlation matrices as:

$$\mathbf{R}_1 = \mathbf{P}_1'^{\top}\mathbf{P}''_2, \quad \mathbf{R}_2 = \mathbf{P}_2'^{\top}\mathbf{P}''_1. \tag{7}$$

Similar to the sample-wise consistency, we compute the consistency between one network's one view prediction with the other network's the other view prediction, leading to two cross-correlation matrices. As done above, we convert $\mathbf{R}_1$ and $\mathbf{R}_2$ to symmetric ones and normalize its rows and columns:

$$\hat{\mathbf{R}}_i = \phi\big((\mathbf{R}_i + \mathbf{R}_i^\top)/2\big),\ i = 1, 2. \tag{8}$$

Then, we have our class-wise cross consistency loss as,

$$\mathcal{L}_{class}^{cross} = -\frac{1}{2C}\big(\operatorname{Tr}(\hat{\mathbf{R}}_1) + \operatorname{Tr}(\hat{\mathbf{R}}_2)\big), \tag{9}$$

Class-wise Consistency Loss. Similar to the sample-wise one, our final class-wise cross-consistency loss is:

$$\mathcal{L}_{class} = \frac{1}{2}(\mathcal{L}_{class}^{inner} + \mathcal{L}_{class}^{cross}). \tag{10}$$

Analysis. We also devise a simple analysis of class-wise consistency, showing that it integrates all the merits claimed by Mutual Information Maximization [28], Minimum Class Confusion [16], and sample-wise consistency. This is achieved by simultaneously addressing three objectives: i) **consistency regularization**, ensuring that predictions derived from distinct augmented views remain consistent, *i.e.*, $\mathbf{P}' = \mathbf{P}''$; ii) **entropy minimization**, facilitating the sharpening of softmax predictions for individual samples, *i.e.*, the rows of $\mathbf{P}$ are one-hot-like; and iii) **prediction diversification**, penalizing collapsed solutions in which the majority of samples are predicted to a single class, *i.e.*, the rows of $\mathbf{P}$ are distinct. Specifically, without loss of generality, assuming that the batch prediction is a square matrix, it can be verified that the optimal solutions of the class-wise consistency are equal to the ones of the following optimization problem:

$$\min_{\mathbf{P}_1,\mathbf{P}_2} \|\frac{1}{2}(\mathbf{P}_1^\top\mathbf{P}_2 + \mathbf{P}_2^\top\mathbf{P}_1) - \mathbf{I}_n\|_F, \tag{11}$$

$$\text{s.t.} \quad \mathbf{P}_i\mathbf{1}_n = \mathbf{1}_n,\ \mathbf{P}_i \geq 0,\ i = 1, 2. \tag{12}$$

The optimal solution for this convex optimization problem is $\mathbf{P}_1 = \mathbf{P}_2$, which are permutation matrices that achieve the three aforementioned objectives. In addition, our class-wise cross consistency further benefits the co-learning of the two networks.

3.5. Overall Objective Function

Both the proposed $\mathcal{L}_{sample}$ (Eq. 3) and $\mathcal{L}_{class}$ (Eq. 10) are designed for the unlabeled sample $\mathcal{D}_t^u$, while for the labeled samples in $\mathcal{D}_t^l$, we employ the cross entropy loss:

$$\mathcal{L}_{CE} = -\sum_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_t^l} \mathbf{y}[\log\big(F(\mathbf{x}|\theta_s)\big) + \log\big(F(\mathbf{x}|\theta_t)\big)]. \tag{13}$$

Summing up the three loss functions, we have our overall objective:

$$\min_{\theta_s,\theta_t} \mathcal{L}_{CE} + \lambda_1\mathcal{L}_{sample} + \lambda_2\mathcal{L}_{class}, \tag{14}$$

where λ_1 and λ_2 are hyper-parameters control the trade-off between losses.

Choice of Final Model. We observe that the performance of $F(\cdot|\theta_s)$ and $F(\cdot|\theta_t)$ converge to a similar point after training. Therefore, without loss of generality, we choose $F(\cdot|\theta_s)$ as the final model for inference.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our method on several popular benchmark datasets, including *Office* [42], *DomainNet* [37],

[1] This works as when two columns are similar, there will be two nonzero values (two classes) for one sample, which violates the sparsity prior.

Table 1. *H-score* on *DomainNet* under the settings of 3-shot and 5-shot using ResNet34 as backbone networks.

	R → C		P → C		C → S		R → P		S → P		R → S		**Mean**	
Method	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	3-shot	5-shot	**3-shot**	**5-shot**
CE	50.8	58.0	53.0	57.3	47.1	54.6	50.3	57.7	53.5	58.8	43.6	50.6	49.7	56.2
MixMatch	48.0	66.5	50.5	65.2	44.4	57.6	50.1	64.6	50.2	65.3	44.2	58.1	47.9	62.8
FixMatch	52.5	68.6	48.2	66.4	49.8	61.1	52.8	68.5	48.6	68.2	43.0	64.2	49.2	66.2
UMA	42.7	60.4	46.2	58.5	41.7	51.2	42.9	56.8	40.4	51.6	40.2	50.5	42.3	54.8
MME	56.7	66.5	58.9	64.1	45.0	55.2	49.9	62.4	52.7	63.8	42.8	51.2	51.0	60.5
CDAC	59.0	71.4	59.2	72.3	50.8	64.7	62.3	70.1	62.3	70.5	51.9	63.1	57.6	68.7
MCL	62.0	74.8	65.2	73.5	53.4	65.1	60.9	72.5	64.7	70.9	54.9	63.1	60.2	70.0
AdaMatch	59.5	73.3	62.1	71.4	52.0	64.7	60.5	73.1	65.9	73.1	51.6	60.8	58.6	69.4
SHOT++	55.6	66.8	59.4	68.1	51.0	58.5	56.1	63.3	56.7	60.8	45.4	53.6	54.0	61.9
CCT	**69.9**	**77.7**	**69.0**	**77.4**	**58.6**	**66.8**	**66.7**	**75.5**	**67.4**	**75.3**	**56.2**	**66.9**	**64.6**	**73.3**

Table 2. *H-score* on *Office-Home* under the settings of 3-shot and 5-shot using ResNet34 as backbone networks.

Method	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	**Mean**
							3-shot						
CE	51.8	73.9	69.9	49.9	70.7	65.6	57.8	51.1	68.9	55.7	54.0	74.6	62.0
MixMatch	51.3	77.6	73.3	51.8	76.4	69.7	56.8	51.1	72.2	58.3	53.4	78.5	64.2
FixMatch	50.6	76.2	67.5	39.9	76.6	66.9	53.9	56.7	67.4	47.1	51.8	75.4	60.8
UMA	54.6	76.3	71.3	52.1	73.7	63.3	57.2	52.5	68.8	54.0	56.4	76.2	63.0
MME	55.6	77.1	72.3	55.2	75.3	68.8	60.0	53.6	72.5	61.4	57.8	77.3	65.6
CDAC	55.7	76.3	72.0	54.0	75.4	68.9	54.5	58.8	72.3	60.6	57.1	76.9	65.2
MCL	54.2	74.2	71.6	54.6	77.6	66.6	54.5	59.6	73.4	60.0	55.8	77.3	65.0
AdaMatch	52.2	71.4	71.1	55.4	78.6	65.4	55.5	59.7	74.9	59.2	54.0	77.8	64.5
SHOT++	54.9	80.3	74.2	53.3	73.9	70.8	55.9	52.3	76.9	50.2	52.7	79.3	64.6
CCT	**57.8**	**80.6**	**77.3**	**61.0**	**79.4**	**76.9**	**62.6**	**57.8**	**77.1**	**63.4**	**59.2**	**80.2**	**69.4**
							5-shot						
CE	56.5	77.1	73.2	58.7	75.2	69.4	61.4	56.5	73.2	60.6	57.2	78.9	66.5
MixMatch	59.1	81.5	76.7	61.3	80.2	75.3	62.6	59.9	77.3	62.6	58.5	83.6	70.2
FixMatch	59.1	80.8	76.7	58.5	79.9	72.4	60.3	61.6	78.0	61.7	62.5	80.0	69.3
UMA	55.8	77.6	72.3	56.4	76.7	68.5	60.0	59.9	75.3	55.6	52.4	77.7	65.7
MME	62.2	80.5	74.3	63.1	81.6	73.4	62.6	61.9	76.5	64.0	62.4	83.1	70.5
CDAC	64.0	80.6	76.3	61.5	80.2	71.8	60.1	62.5	76.1	61.4	59.7	78.3	69.4
MCL	63.5	79.5	76.9	63.1	80.4	73.5	61.1	64.3	78.2	61.8	59.4	79.7	70.1
AdaMatch	57.6	78.7	76.3	59.1	82.1	70.1	60.3	63.4	78.1	61.1	61.2	80.4	69.0
SHOT++	54.6	80.3	75.2	62.2	75.0	74.1	62.9	55.5	77.5	64.1	50.9	80.6	67.7
CCT	**65.9**	**82.2**	**79.3**	**67.3**	**82.1**	**77.9**	**67.9**	**65.6**	**78.9**	**67.5**	**63.6**	**83.9**	**73.5**

Table 3. *H-score* on *Office* under the settings of 5-shot and 10-shot using ResNet34 as backbone networks.

	W → A		D → A	
Method	3-shot	5-shot	3-shot	5-shot
CE	60.4	61.2	60.9	61.6
Mixmatch	61.2	69.7	62.6	69.4
Fixmatch	62.4	63.4	61.0	67.4
UMA	60.1	61.5	58.7	62.4
MME	64.7	69.0	64.3	69.1
CDAC	64.5	68.5	63.9	67.8
MCL	65.9	68.8	66.0	68.1
AdaMatch	62.8	65.4	64.1	67.4
SHOT++	64.0	68.3	64.5	68.2
CCT	**68.3**	**71.3**	**67.9**	**70.7**

and *Office-Home* [49], with different $\mathcal{C}_s$, $\mathcal{C}_t$ and ξ (see Sec. 3.1). Similar to most recent works [25,56], we conduct 3-shot (3 labeled samples per class in the target domain) and 5-shot experiments on all datasets. **DomainNet** [37] was first introduced as a multi-source domain adaptation benchmark comprising 6 domains with 345 categories. Following [25, 43], we select the *Real*, *Clipart*, *Painting*, *Sketch* domains with 126 categories for evaluation. The first 80 classes are used as $\mathcal{C}_s$ and last 96 classes are used as $\mathcal{C}_t$, hence $|\mathcal{C}| = 50$ and $\xi = 0.4$. **Office-Home** [49] is a popular domain adaptation benchmark, which consists of 4 domains (*Real*, *Clipart*, *Product*, *Art*) and 65 categories. We use the first 43 classes as $\mathcal{C}_s$ and the last 35 classes as $\mathcal{C}_t$, hence $|\mathcal{C}| = 13$ and $\xi = 0.2$. **Office** [42] contains 3 domains (Amazon, Webcam, and DSLR) with 31 classes. Similar to [25,43], we conduct experiments from DSLR to Amazon and Webcom to Amazon to evaluate on the domain with enough examples. The first 25 classes are set as $\mathcal{C}_s$ and the last 25 classes are set as $\mathcal{C}_t$, hence $|\mathcal{C}|$ = 19 and $\xi = 0.6$.

Evaluation Criterion. Following [12], we use *H-score* to evaluate the performance of our method, which is defined as:

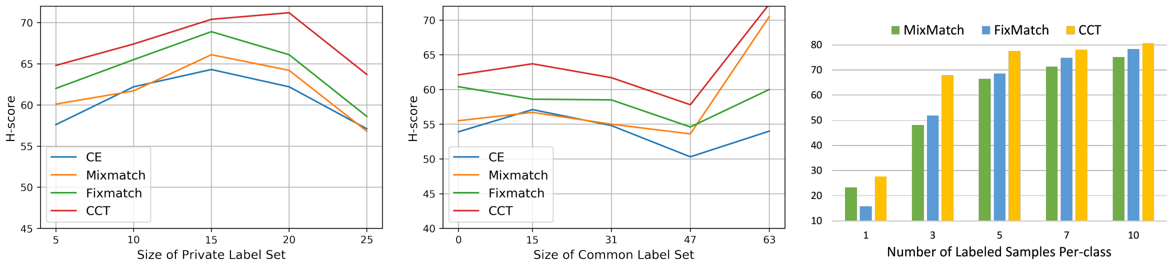
$$H = 2 \cdot \frac{a_{\mathcal{C}} \cdot a_{\bar{\mathcal{C}}_t}}{a_{\mathcal{C}} + a_{\bar{\mathcal{C}}_t}}, \tag{15}$$

where $a_{\mathcal{C}}$ and $a_{\bar{\mathcal{C}}_t}$ represent the accuracy for the common class $\mathcal{C}_t$ and the target private class $\bar{\mathcal{C}}_t$, respectively.

Implementation Details. Similar to [25, 43], we use

Table 4. *H-score* on *Domainnet* under the settings of 5-shot using ResNet34. The first row means only $\mathcal{L}_{CE}$ (Eq. 13) is utilized.

	L_{sample}^{inner}	L_{class}^{inner}	L_{sample}^{cross}	L_{class}^{cross}	R → C	P → C	C → S	R → P	S → P	R → S	**Mean**
(a)					58.0	57.3	54.6	57.7	58.8	50.6	56.2
(b)	✓				68.6	66.4	61.1	65.5	68.2	64.2	65.7
(c)	✓		✓		71.8	67.9	65.2	70.4	70.9	64.7	68.5
(d)	✓	✓			76.2	75.8	66.3	74.4	72.9	64.8	71.7
(e)			✓	✓	77.6	77.2	67.2	73.8	74.8	66.0	72.8
(f)	✓	✓	✓	✓	77.7	77.4	66.8	75.5	75.3	66.8	73.3



(a) *H-score* w.r.t. $|\bar{\mathcal{C}}_t|$.

(b) *H-score* w.r.t. $|\mathcal{C}|$

(c) *H-score* w.r.t. $|\mathcal{D}_t^l|/C$

Figure 4. (a) *H-score* w.r.t. $|\bar{\mathcal{C}}_t|$ in task R → C of *Office-Home* with $\xi = 0.23$. (b) *H-score* w.r.t. $|\mathcal{C}|$ in task R → C of *Office-Home*. (c) *H-score* w.r.t. number of labeled samples per-class in task R → C of *Domainnet*.

Resnet34 [14] as the backbone network where the classifier consists of a L_2 normalization layer that projects feature into a spherical feature space and a linear layer. Following [28], we pre-train the source model with label smoothing [35] for 50 epochs. For the pre-training of the target-only network, we train the network for 50 epochs with supervised contrastive loss [17] for the labeled samples and SimCLR [7] loss for the unlabeled samples. Moreover, we use RandAugment [9] as the augmentation for x'' and standard random resize, flip and crop as the augmentation for x'. For the adaptation phase, the experimental settings (*i.e.*, optimizer, batch size, *etc.*) are the same as MME [43]. We set λ_1 as 1, and λ_2 as 1 for *Domainnet* and 0.5 for *Office* and *Office-Home*. All experiments are implemented with PyTorch [36] on a single NVIDIA 2080Ti.

4.2. Comparison Experiments

We compared our method with i) baseline methods: only train the labeled target samples with cross-entropy loss (CE), ii) semi-supervised domain adaptation (SSDA) methods: MME [43], CDAC [25], MCL [54] and AdaMatch [4], iii) semi-supervised learning (SSL) methods: MixMatch [3] and FixMatch [47], iv) a semi-supervised model adaptation (SSMA) method: SHOT++ [30], and v) a universal model adaptation method: UMA [20]. To make a fair comparison, we implement them with the same $F(\cdot|\theta_s)$ as used in our method. Note that we only report the results using $F(\cdot|\theta_s)$ as it outperforms those using $F(\cdot|\theta_t)$. The implementation details of the compared methods can be found in the supplementary file. The results on the three benchmark datasets are shown in Table 1, 2 and 3, respectively. It can be observed that our method significantly outperforms the state-of-the-art SSL, SSDA, SSMA, and UMA methods in terms of H-score.

Results. As Table 1 shows, our method outperforms all previous methods in all domains on *Domainnet*, achieving 64.6% and 73.3% mean H-scores under 3-shot and 5-shot settings, respectively. Notably, our method surpasses the State-of-the-Art (SOTA) SSDA technique, *i.e.*, AdaMatch [4], by substantial margins, demonstrating the superiority of the proposed method. Furthermore, it is imperative to highlight that certain methods falter even in the 3-shot setting, exhibiting performance inferior to the basic CE method. In contrast, CCT consistently maintains its stability and superiority. As Table 2 and 3 show, CCT still works effectively and outperforms all the previous methods on *Office-Home* and *Office* significantly, which further demonstrates the efficacy of the proposed method.

4.3. Ablation Study

We perform an ablation study on *Domainnet* to verify the efficacy of each component of the proposed method. For clarity, except for the most basic sample-wise inner consistency, either sample/class-wise or inner/cross consistency are jointly used. As Table 4 shows, starting from $\mathcal{L}_{sample}^{inner}$ (b) that represents the case when only the most basic sample-wise inner consistency is used, adding the sample-wise cross consistency loss (c) improves the performance by 2.8% and a further incorporation of the class-wise consistency losses (f) boosts the performance by 4.7%. This demonstrates the efficacy of both the proposed collaborating consistency training framework, especially the cross consistency, and the proposed class-wise consistency regularization. In general, our final method outperforms its variant with only the basic sample-wise inner consistency regularization by 7.6%. Surprisingly, we observed that the cross consistency regularization could work effectively without the inner consistency regularization (e), and performs even better than inner consistency alone (d) by 1.1%. We conjecture that the inner consistency can be *implicitly* regularized by the cross consistency, and they are compatible as the network achieves better performance when they are combined together. Furthermore, the 6% improvement of sample and class-wise inner consistency (d) over sample-wise inner consistency alone (b) demonstrates the efficacy and versatility of the proposed class-wise consistency.

4.4. Analysis Experiments

In this section, we conduct several additional experiments to validate the versatility of our method. Further analysis experiments can be found in the supplementary file.

Varying Sizes of $\bar{\mathcal{C}}_t$ and $\bar{\mathcal{C}}_s$. Similar to [60], with fixed $|\mathcal{C}_s \cup \mathcal{C}_t|$ and ξ, we investigate how the performance of our method changes with various sizes of $\bar{\mathcal{C}}_t$. We conduct the experiments on the task of *Real* → *Clipart* of *Office-Home*. We set ξ as 0.23 and $\mathcal{C} = 15$. Note that $|\bar{\mathcal{C}}_s|$ will change correspondingly with $|\bar{\mathcal{C}}_t|$. As Fig. 4a shows, our method outperforms all the other methods for all $|\bar{\mathcal{C}}_t|$. Furthermore, compared with MixMatch [3] and FixMatch [47], the performance gap increases as the size of $|\bar{\mathcal{C}}_t|$ increases, which implies that our method (CCT) is more suitable for the scenarios with large target-private categories.

Varying Sizes of $\mathcal{C}$. Similarly, we investigate how the performance of our method changes with various sizes of $\mathcal{C}$ but fixed $|\mathcal{C}_s \cup \mathcal{C}_t|$. We conduct the experiments on the task of *Real* → *Clipart* of *Office-Home*. We set $|\mathcal{C}_s \cup \mathcal{C}_t| = 65$ and $|\bar{\mathcal{C}}_t| = |\bar{\mathcal{C}}_s|$. The results are shown in Fig. 4b. Note that when $|\mathcal{C}| = 0$, we set $\bar{\mathcal{C}}_s = \bar{\mathcal{C}}_t = 15$ and report the accuracy of target-private categories samples only. It can be observed that our method (CCT) outperforms all the other methods for all $|\mathcal{C}|$. It is worth noting that our CCT can still achieve decent performance when there is no intersection between the source and target label sets.

Varying Sizes of $\mathcal{D}_t^l$. Although we already show the superior performance of our method in the 3-shot and 5-shot settings, it is still interesting to see how the performance of our method changes under various number of labeled samples. We conduct the experiments on the task of *Real* → *Clipart* of *Domainnet*. As Fig. 4c shows, our method (CCT) consistently outperforms MixMatch [3] and FixMatch [47] in all cases. Specifically, when the number of labeled samples is small, CCT surpasses the other methods.

Results on SSMA Benchmark. In addition to Universal Semi-supervised Model Adaptation (USMA), we also implement CCT on Semi-supervised Model Adaptation (SSMA) benchmark. Following [30], we conduct experiments on *Office-Home* 1-shot setting using VGG16 as the backbone. The mean accuracy over 12 domain pairs is shown in Table 5. It can be observe that CCT outperforms the current SOTA: SHOT++ [30], which further demonstrates the versatility of CCT.

Table 5. SSMA Results on *Office-Home* 1-shot setting using VGG16 as the backbone.

Method	CE	MME	SHOT-IM++	SHOT++	CCT
Accuracy	57.4	62.7	65.2	66.1	68.3

5. Conclusions

In this paper, we introduce a more realistic and challenging domain adaptation problem called Universal Semi-supervised Model Adaptation (USMA), where i) instead of the source data, only the pre-trained source model is available; ii) the source and target domain do not share the same label set; iii) and there are only a few labeled samples in each class of the target domain. We show that USMA cannot be resolved by naively fine-tune the pre-trained source model using semi-supervised learning or semi-supervised model adaptation methods. Thus, we propose a collaborative consistency training framework that addresses USMA by utilizing the complementary strengths of both the pre-trained source model and its variant pre-trained on target data only. Moreover, we propose a two-perspective (*i.e.*, sample-wise and class-wise) consistency regularization that helps to make full advantage of our framework, leading to better performance. Experimental results show that our method surpass existing semi-supervised learning, semi-supervised domain adaptation and semi-supervised model adaptation methods on various benchmark datasets.

Acknowledgment. The work was supported in part by NSFC with Grant No. 62293482, the Basic Research Project No. HZQBKCZYZ-2021067 of Hetao ShenzhenHK S&T Cooperation Zone, the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund 202002, by Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), and by Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. ZDSYS201707251409055). It was also partially supported by NSFC62172348, Outstanding Yound Fund of Guangdong Province with No. 2023B1515020055 and Shenzhen General Project with No. JCYJ20220530143604010.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017. 4

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 7, 8

[4] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022. 2, 3, 7

[5] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018. 3

[6] Liang Chen, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Geometric anchor correspondence mining with uncertainty modeling for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16134–16143, June 2022. 3

[7] 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. 7

[8] Haoang Chi, Feng Liu, Wenjing Yang, Long Lan, Tongliang Liu, Bo Han, William Cheung, and James Kwok. Tohan: A one-step approach towards few-shot hypothesis adaptation. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 7

[10] Bin Deng, Yabin Zhang, Hui Tang, Changxing Ding, and Kui Jia. On universal black-box domain adaptation, 2021. 3

[11] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. Source-free domain adaptation via distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7212–7222, 2022. 3

[12] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*, pages 567–583. Springer, 2020. 3, 6

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

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 1, 7

[15] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pages 934–940, 2020. 2

[16] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020. 5

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 7

[18] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 591–607. Springer, 2020. 2

[19] Jogendra Nath Kundu, Suvaansh Bhambri, Akshay Kulkarni, Hiran Sarkar, Varun Jampani, and R Venkatesh Babu. Concurrent subsidiary supervision for unsupervised source-free domain adaptation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 177–194. Springer, 2022. 3

[20] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020. 3, 7

[21] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625, 2021. 3

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1104–1113, 2021. 2

[23] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer, 2020. 2

[24] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9757–9766, 2021. 3

[25] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2505–2514, 2021. 1, 2, 6, 7

[26] Kai Li, Chang Liu, Handong Zhao, Yulun Zhang, and Yun Fu. Ecacl: A holistic framework for semi-supervised domain

adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8578–8587, 2021. 2

[27] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 3

[28] 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. 1, 2, 3, 5, 7

[29] 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. 2

[30] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3, 7, 8

[31] Omri Lifshitz and Lior Wolf. A sample selection approach for universal domain adaptation. *arXiv preprint arXiv:2001.05071*, 2020. 3

[32] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2021. 2

[33] Ning Ma, Jiajun Bu, Lixian Lu, Jun Wen, Zhen Zhang, Sheng Zhou, and Xifeng Yan. Semi-supervised hypothesis transfer for source-free domain adaptation. *arXiv preprint arXiv:2107.06735*, 2021. 2, 3

[34] Xinhong Ma, Junyu Gao, and Changsheng Xu. Active universal domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8968–8977, October 2021. 3

[35] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019. 7

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 7

[37] 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, 6

[38] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Contradictory structure learning for semi-supervised domain adaptation. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 576–584. SIAM, 2021. 2

[39] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 165–182. Springer, 2022. 3

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[41] Subhankar Roy, Martin Trapp, Andrea Pilzer, Juho Kannala, Nicu Sebe, Elisa Ricci, and Arno Solin. Uncertainty-guided source-free domain adaptation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 537–555. Springer, 2022. 3

[42] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 5, 6

[43] 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. 1, 2, 6, 7

[44] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*, 2020. 3

[45] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. *arXiv preprint arXiv:2104.03344*, 2021. 3

[46] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–168, 2018. 3

[47] 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. *arXiv preprint arXiv:2001.07685*, 2020. 2, 4, 7, 8

[48] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 1

[49] 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. 6

[50] Xiaodong Wang, Junbao Zhuo, Shuhao Cui, and Shuhui Wang. Learning invariant representation with consistency and diversity for semi-supervised source hypothesis transfer. *arXiv preprint arXiv:2107.03008*, 2021. 2, 3

[51] Yushuang Wu, Zizheng Yan, Ce Chen, Lai Wei, Xiao Li, Guanbin Li, Yihao Li, Shuguang Cui, and Xiaoguang Han. Scoda: Domain adaptive shape completion for real scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17630–17641, 2023. 2

[52] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9010–9019, 2021. 3

[53] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 2, 4

[54] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. *arXiv preprint arXiv:2205.04066*, 2022. 2, 7

[55] Zizheng Yan, Xianggang Yu, Yipeng Qin, Yushuang Wu, Xiaoguang Han, and Shuguang Cui. Pixel-level intra-domain adaptation for semantic segmentation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 404–413, 2021. 1

[56] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Mico: Mixup co-training for semi-supervised domain adaptation. *arXiv e-prints*, pages arXiv–2007, 2020. 2, 6

[57] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020. 2, 3

[58] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 3

[59] Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 474–483, 2021. 3

[60] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2720–2729, 2019. 2, 3, 8

[61] Qing Yu, Atsushi Hashimoto, and Yoshitaka Ushiku. Divergence optimization for noisy universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2515–2524, 2021. 3