Class Relationship Embedded Learning for Source-Free Unsupervised Domain Adaptation

Yixin Zhang[1,2] Zilei Wang*[2] Weinan He[2]
[1] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[2] University of Science and Technology of China
{zhyx12, zlwang}@ustc.edu.cn, hwn2018@mail.ustc.edu.cn

Abstract

This work focuses on a practical knowledge transfer task defined as Source-Free Unsupervised Domain Adaptation (SFUDA), where only a well-trained source model and unlabeled target data are available. To fully utilize source knowledge, we propose to transfer the class relationship, which is domain-invariant but still under-explored in previous works. To this end, we first regard the classifier weights of the source model as class prototypes to compute class relationship, and then propose a novel probability-based similarity between target-domain samples by embedding the source-domain class relationship, resulting in Class Relationship embedded Similarity (CRS). Here the inter-class term is particularly considered in order to more accurately represent the similarity between two samples, in which the source prior of class relationship is utilized by weighting. Finally, we propose to embed CRS into contrastive learning in a unified form. Here both class-aware and instance discrimination contrastive losses are employed, which are complementary to each other. We combine the proposed method with existing representative methods to evaluate its efficacy in multiple SFUDA settings. Extensive experimental results reveal that our method can achieve state-of-the-art performance due to the transfer of domain-invariant class relationship. [1]

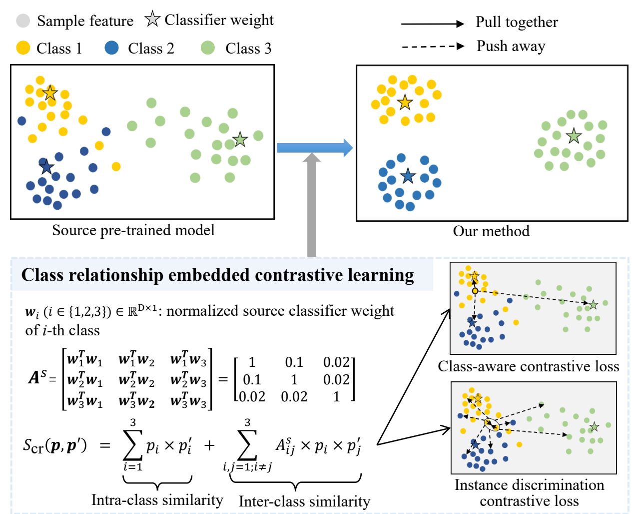

Figure 1. Illustration of our proposed method. The upper left represents the target-domain featrue distribution by source pre-trained model. The upper right is the feature distribution obtained by our method. The bottom is the process of class relationship embedded contrastive learning. Best viewed in color.

1. Introduction

Benefiting from the large amount of labeled training data, deep neural networks have achieved promising results in many computer vision tasks [7, 15, 19, 95]. To reduce the annotation cost, Unsupervised Domain Adaptation (UDA) has been devised by transferring knowledge from a label-rich source domain to a label-scarce target domain. Currently, many UDA methods have been proposed that jointly learn on the source and target data. But they would be unapplicable for some real-world scenarios involving privacy (*e.g.*, medical images, surveillance videos) because the source-domain data cannot be accessed. Thus, more recent methods [36, 42, 71, 79, 87, 88] focus on Source-Free Unsupervised Domain Adaptation (SFUDA). Under this setting, the labeled source data are not accessible any more when training the target model, but the pre-trained model in the source domain is provided. Then a natural question arises, *i.e.*, what knowledge should we transfer to facilitate the learning of unlabeled target-domain data?

Some methods [21, 36, 42] assume that the source hypothesis (*i.e.*, classifier [42] or whole model [21, 36]) contains sufficient knowledge for target domain. Then they transfer source knowledge by directly aligning features with the fixed source classifier [42], resorting to historical mod-

*Corresponding author
[1]Code is available at https://github.com/zhyx12/CRCo

els [21], or weight regularization [36]. Another line of works [87, 89, 90] assume the source model already forms some semantic structure, and then utilize the local information (*i.e.*, neighborhood) of feature space to enforce the similarity in the output space. Despite these progress, what knowledge to be transferred remains an open question.

In this work, we propose to transfer the class relationship represented by the similarity of classifier weights. Actually, the class relationship is domain-invariant [14], since the same class in the source and target domains essentially represents the same semantic in spite of domain discrepancy. For example, the computers are always more similar with the TV than the scissors. Thus, it is reasonable to guide the target domain learning using class relationship prior. Unlike previous methods that learn class distribution by pseudo labeling [21, 42] and local aggregation of neighborhood predictions [79, 87, 89, 90], here we explicitly model the source-domain class relationship. To be specific, we regard each weight of classifier as the class prototype [90], and then compute a class relationship matrix $\boldsymbol{A}^s$ by cosine similarity, as shown in Figure 1.

Before explaining how to use this matrix, we illustrate the purpose of representation learning in the target domain using Figure 1, where three classes are adopted for clarity. The top left shows the target-domain feature distribution together with the class weights learned from the source domain. It can be seen that such a situation makes it difficult to perform correct classification. In fact, it is expected that the learned features are discriminative and compact around the corresponding class weight, as shown on the top right. To this end, an intuitive way is to make the relationship of learned class prototypes in the target domain consistent with that in the source domain. However, it has a very limited effect on training the target-domain model as such a prototype-level constraint is too weak for optimization.

In this work, we propose to embed the source-domain class relationship in contrastive learning, which has been shown to be outstanding in representation learning [9, 18]. Here we design a novel sample similarity by taking into account $\boldsymbol{A}^s$. Specifically, we compute the similarity between two target domain samples in the output space, represented by the prediction probabilities (*i.e.*, $\boldsymbol{p}$ and $\boldsymbol{p}'$), as shown in Figure 1. In particular, we consider the inter-class term (*i.e.* $\sum_{i\neq j} p_i p'_j$) in addition to the traditional intra-class term (*i.e.* $\sum_i p_i p'_i$). Note that the sum of these two terms equals one, in which the intra-class term measures the similarity that two samples come from the same class, while the inter-class term measures that from different classes. Considering the relationship between classes, we weight the inter-class term by the non-diagonal elements in $\boldsymbol{A}^s$. In particular, if the class i is closer to j than k (*i.e.*, $A^s_{ij} > A^s_{ik}$), $p_i p'_j$ would be more important in calculating the similarity of $\boldsymbol{p}$ and $\boldsymbol{p}'$. By the above design, our proposed similarity can more accurately express the relationship of two samples based on output space. For example, among three classes $1, 2, 3$ in Figure 1, the classes 1 and 2 are closer. Given three samples x_1, x_2, x_3 belonging to them with the probabilities $[0.9, 0.05, 0.05]$, $[0.05, 0.9, 0.05]$, and $[0.05, 0.05, 0.9]$. If we only use the intra-class term, all three samples have the same similarity. But for our designed similarity, x_1 is closer to x_2 than x_3, which is more reasonable.

On the basis of the proposed similarity, we further propose to perform contrastive learning in a unified form. Inspired by recent success in semi-supervised learning [67, 85, 91] and unsupervised learning [9, 18], we propose two types of contrastive losses. As shown in Figure 1, the first one is **C**lass **R**elationship embedded **C**lass-**A**ware **Co**ntrastive (CR-CACo) loss where the high-confident samples are enforced to be close to the corresponding prototype and away from other prototypes. Due to embedding the prior class relationship, our CR-CACo loss is more robust to label noise caused by domain shift. The second one is **C**lass **R**elationship embedded **I**nstance **D**iscrimination **Co**ntrastive (CR-IDCo) loss where two views of the same sample are encouraged to be close and away from all other samples. Benefited from our designed accurate similarity, the CR-IDCo loss would more effectively learn discriminative features [9, 18, 75]. Actually, these two losses are complementary to each other, and their combination can achieve better performance.

Our contributions are summarized as follows:

- We propose to explicitly transfer class relationship for SFUDA which is more domain-invariant. And we propose to embed the class relationship into contrastive learning in order to effectively perform representation learning in the target domain.
- We propose a novel class relationship embedded similarity which can more accurately express the sample relationship in the output space. Furthermore, we propose two contrastive losses (*i.e.*, CR-CACo and CR-IDCo) exploiting our designed similarity.
- We conduct extensive experiments to evaluate the proposed method, and the results validate the effectiveness of our method, which achieves the state-of-the-art performance on multiple SFUDA benchmarks.

2. Related Work

2.1. Unsupervised Domain Adaptation

The conventional approaches in UDA [2, 3] are to learn domain-invariant representations, and can be classified into two coarse types. The first one decreases some distribution discrepancy metrics [49, 57, 68, 72, 84]. Another common line is adversarial training [11, 12, 45, 48, 64, 77, 96]. To increase discriminability, more recent DA methods attempt to investigate the target domain structure. Self-training as a typical approach generates target domain pseudo labels [16, 17, 27, 28, 40, 46, 51, 54, 94, 101]. Another category is to construct prototypes [8, 34, 55, 80, 81, 93, 97, 99] or cluster centers [13, 26, 47, 70] across domains and then perform class-wise alignment. Among these methods, most of them rely on source-labeled data during target training, while pseudo label based methods can be directly applied in SFUDA. In our CR-CACo loss, pseudo label is also used and our method can alleviate the negative impact of label noise due to the usage of source class relationship.

2.2. Source-free Domain Adaptation

Early SFUDA methods [31, 32, 36] resort to synthesize extra training samples to get compact decision boundaries. Some recent methods [10, 21, 33, 42, 61] use pseudo-label based self-training. SHOT [42] proposes to freeze the source classifier and it clusters target features by IM loss [20] along with clustering based pseudo labeling. HCL [21] adopts feature space Instance Discrimination [78] from current and historical models, together with pseudo label learning conditioned on historical consistency. D-MCD [10] proposes denoised MCD [64] to mitigate the impact of sample selection bias and label noise. CoWA-JMD [33] uses the joint model-data structure score as sample-wise weights, along with weight mixup to exploit more target knowledge. BMD [61] proposes a class-balanced multicentric dynamic prototype strategy to obtain more accurate pseudo labels. Another line of methods (*i.e.* NRC [87], G-SFDA [89] and AaD [90]) propose neighborhood clustering which enforces prediction consistency between local neighbors.

In addition to the above methods, others solve SFUDA from different aspects. A^2Net [79] proposes to learn an additional target classifier and uses a contrastive category-wise matching module to learn compact features. DIPE [74] and VMP [25] explore transferability of source model parameters. Sub-sup [29] proposes novel subsidiary pretext tasks to assist domain adaptation. U-SFAN [62] uses the uncertainty quantified by the predictions to guide the target adaptation. Feat-mix [30] proposes image-level and feature-level mixup to enhances the discriminability-transferability tradeoff. Among these methods, few of them explicitly consider the usage of class relationship.

3. Method

3.1. Preliminaries

For source-free unsupervised domain adaptation (SFUDA), we are given source pre-trained model and an unlabeled target domain with N_t samples as $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$. Target domain have same C classes as source domain (for closed-set setting). The goal of SFUDA is to adapt the model to target domain without source data. We divide the model into two parts: the feature extractor f, and the classifier g. The output of classifier is denoted as $\boldsymbol{p} = \delta(g(f(x))) \in \mathbb{R}^C$ where δ is the softmax function.

We adopt a teacher-student framework following [18] as shown in Figure 2. The teacher feature extractor (*i.e.* $\tilde{f}$) is continuously updated by exponential moving average of the student feature extractor (*i.e.* f), and the classifiers (*i.e.* $g, \tilde{g}$) are shared and frozen as in SHOT [42]. We use sample probabilities as input for contrastive loss, the query probability is obtained from the student model, and the positive and negative probabilities are obtained through the teacher model. During inference, we directly use the student model. In the following, we first introduce class relationship embedded similarity, and then elaborate on how it is used in the proposed contrastive learning process.

3.2. Class Relationship Embedded Similarity

The inherent semantic and visual relationships among different classes are consistent across domains, regardless of distribution discrepancy. Inspired by this, we propose to explicitly transfer class similarities. Here we regard the normalized classifier weights $\boldsymbol{W}^s = [\boldsymbol{w}_1^s, \boldsymbol{w}_2^s, ...\boldsymbol{w}_C^s]^T \in \mathbb{R}^{C\times D}$ as class prototypes where D is the dimension of the last feature. Thus, the class similarity matrix $\boldsymbol{A}^s$ can be obtained by $\boldsymbol{A}^s = \boldsymbol{W}^s(\boldsymbol{W}^s)^T \in \mathbb{R}^{C\times C}$.

An natural way to utilize $\boldsymbol{A}^s$ is to obtain target class similarity matrix $\boldsymbol{A}^t$ by target prototypes and then enforce consistency between $\boldsymbol{A}^s$ and $\boldsymbol{A}^t$. However, such a prototype-level constraint is coarse-grained and not strong enough to benefit the training. Instead of prototype-level, we propose a probability based sample-level similarity that considers not only the traditional intra-class term, but also the inter-class term. The latter contains probability products of all different classes, thus we can embed the class relationship matrix $\boldsymbol{A}^s$ into it as coefficients. Specifically, given the probabilities $\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{R}^{C\times 1}$ of two samples, our **C**lass **R**elationship embedded **S**imilarity (CRS) can be presented as follows:

$$\begin{aligned} s_{intra}(\boldsymbol{p}, \boldsymbol{p}') &= \sum_{i=1}^{C} p_i \times p'_i \\ s_{inter}(\boldsymbol{p}, \boldsymbol{p}') &= \sum_{i=1}^{C} \sum_{j=1, j\neq i}^{C} A^s_{i,j} \times p_i \times p'_j \\ s_{cr}(\boldsymbol{p}, \boldsymbol{p}') &= s_{intra}(\boldsymbol{p}, \boldsymbol{p}') + \lambda_{inter} s_{inter}(\boldsymbol{p}, \boldsymbol{p}') \end{aligned} \tag{1}$$

where s_{intra} is the traditional intra-class similarity, and s_{inter} is the inter-class similarity, λ_{inter} is a trade-off parameter between these two terms.

Given $\lambda_{inter} = 1$, our similarity can be presented in a simpler version as follows:

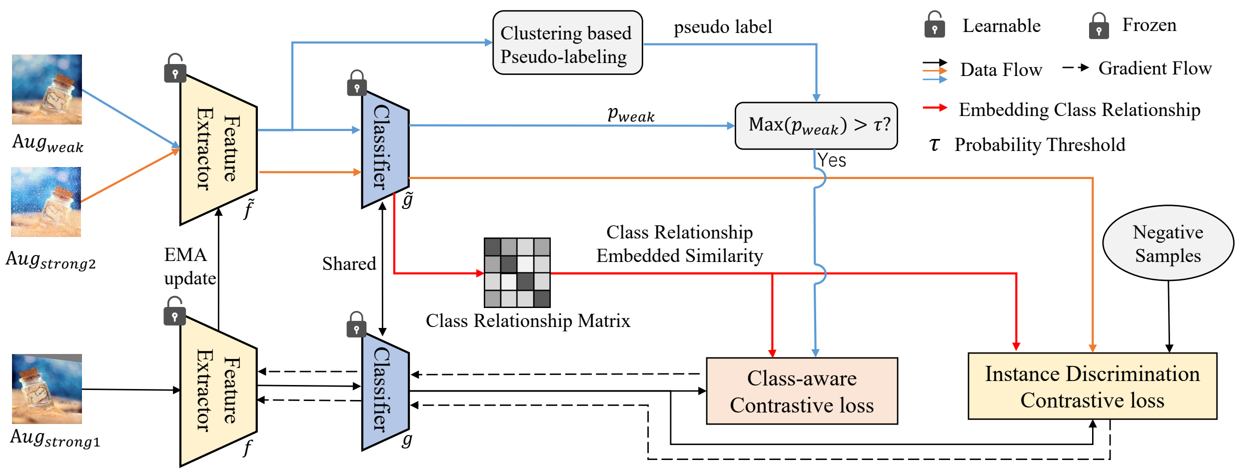

Figure 2. The framework of our proposed method. We use a teacher-student architecture to conduct contrastive learning. Class relationship matrix is extracted from fixed classifier and embedded in two types of contrastive loss. Best viewed in color.

$$
\begin{aligned}
s_{cr}(\boldsymbol{p}, \boldsymbol{p}') &= \sum_{i=1}^{C}\sum_{j=1}^{C} A_{i,j} \times p_i \times p'_j \\
&= \sum_{i=1}^{C}\sum_{j=1}^{C} (p_i \boldsymbol{w}_i^s)^T (p'_j \boldsymbol{w}_j^s) = (\boldsymbol{p}^T \boldsymbol{W}^s)(\boldsymbol{p}'^T \boldsymbol{W}^s)^T
\end{aligned} \tag{2}
$$

where $\boldsymbol{p}^T\boldsymbol{W}^s \in \mathbb{R}^{1\times D}$ can be viewed as a new feature which is probability weighted sum of different class weights.

3.3. Class Relationship Embedded Contrastive Loss

Inspired by the success of contrastive learning in different tasks [9, 18, 21, 26, 85], we propose to use our CRS for contrastive loss. Following previous work [9, 18], the contrastive loss can be presented as follows:

$$
\begin{aligned}
h(\boldsymbol{p}, \boldsymbol{p}') &= \exp(s_{cr}(\boldsymbol{p}, \boldsymbol{p}')/T_{co}), \\
\ell_{co} &= -\log \frac{h(\boldsymbol{p}, \boldsymbol{p}_+)}{h(\boldsymbol{p}, \boldsymbol{p}_+) + \sum_{\boldsymbol{p}_- \in M} h(\boldsymbol{p}, \boldsymbol{p}_-)},
\end{aligned} \tag{3}
$$

where T_{co} is a temperature parameter, and h denotes the exponent of scaled similarity. M is the memory bank [18] storing probabilities of negative samples. Since our CRS is built on probability, here we use probabilities as inputs instead of original features [18] or projected features [9].

With this unified form, we further instantiate it to two losses: class-aware contrastive loss and instance discrimination contrastive loss. The framework of our proposed method is shown in Figure 2.

Class-Aware Contrastive Loss To achieve class-aware learning, it is important to select proper samples and obtain more accurate pseudo labels for them. Inspired by previous efforts in SSL [67, 85] and UDA [41, 93], we consider constructing positive pairs from strongly and weakly augmented views of the same image. Specifically, given probabilities $\boldsymbol{p}_s$ and $\tilde{\boldsymbol{p}}_w$ (tilde superscript denotes output by the teacher model) of strongly and weakly augmented views, we first compare $Max(\tilde{\boldsymbol{p}}_w)$ with a predefined threshold τ to select high-confident samples. Due to domain shift, instead of directly using $argmax(\boldsymbol{p}_w)$ as pseudo label, we adopt the clustering-based pseudo-labeling method in SHOT [42]. Based on pseudo label, we can get the positive samples $\boldsymbol{p}^*$ with the one-hot form. For the negative samples $\boldsymbol{p}_-$, we use one-hot probabilities of all classes except the pseudo label class. Therefore, both positive and negative samples (*i.e.* $\boldsymbol{p}^*$ and $\boldsymbol{p}_-$) are generated from the pseudo label of weak target sample through teacher model. Finally, the **C**lass **R**elationship embedded **C**lass-**A**ware **Co**ntrastive (CR-CACo) loss is presented as

$$
\ell_{cr\_caco} = -\log \frac{h(\boldsymbol{p}_s, \boldsymbol{p}^*)}{h(\boldsymbol{p}_s, \boldsymbol{p}^*) + \sum_{\boldsymbol{p}_- \in M_{caco}} h(\boldsymbol{p}_s, \boldsymbol{p}_-)}, \tag{4}
$$

where h follows the definition in Equation (3), M_{caco} stores negative samples and is different for each sample according to its pseudo label.

Regardless of pseudo label generation, using only s_{intra} and replacing $\boldsymbol{p}_s$ with features and $\boldsymbol{p}*, \boldsymbol{p}_-$ with classifier weights, the above Equation (4) is actually the FixMatch loss [67]. Although FixMatch can improve the performance, our CR-CACo loss can greatly outperform it.

Our CR-CACo loss is more robust to label noise due to the class relationship embedded inter-class similarity. We compare our CRS with the traditional semi-supervised FixMatch loss. The results are shown in Figure 3. It can be seen that the simple IM loss [42] has lower entropy since it has an entropy minimization term. When FixMatch loss is used, the entropy will increase since the class relationship is learned through high-confidence target samples. Our CRS

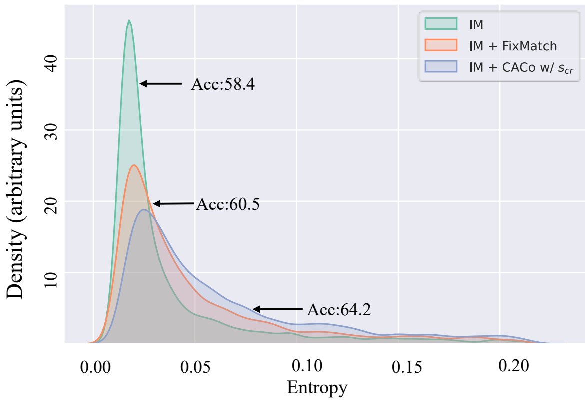

Figure 3. Entropy density plots under SUDA setting Office-Home Rw→Cl. Our class relationship embedded similarity s_{cr} has higher entropy but better accuracy. Best viewed in color.

has larger entropy since we explicitly consider the class relationship and can alleviate the negative impact of label noise.

Instance Discrimination Contrastive Loss The instance discrimination contrastive (IDCo) loss has shown remarkable success in self-supervised learning [9, 18]. It can learn discriminative features by regrading the two views of the same sample as positive pairs and all other samples as negative samples. The traditional IDCo loss uses features as input, but it only makes the feature discriminative and cannot guarantee that the learned feature will be close to corresponding prototypes (*i.e*. classifier weights). Conducting IDCo loss in the output space (*i.e*. computing similarity using s_{intra}) can alleviate this problem [35], but the more informative relationship between samples in the original feature space can not be fully expressed. This can be addressed by our CRS since we embed the source prior class similarity and can reflect more accurate relationship of samples.

Here we use the probabilities $\boldsymbol{p}_{s1}$ and $\tilde{\boldsymbol{p}}'_{s2}$ of two strong augmented views from the same image as positive pairs, and probabilities of all other samples in the same batch as negative samples. Finally, the **C**lass **R**elationship embedded **I**nstance **D**iscrimination **Co**ntrastive (CR-IDCo) loss is presented as follows:

$$\ell_{cr\_idco} = -\log \frac{h(\boldsymbol{p}_{s1}, \tilde{\boldsymbol{p}}'_{s2})}{h(\boldsymbol{p}_{s1}, \tilde{\boldsymbol{p}}'_{s2}) + \sum\limits_{\tilde{\boldsymbol{p}}_{-} \in M_{idco}} h(\boldsymbol{p}_{s1}, \tilde{\boldsymbol{p}}_{-})}, \tag{5}$$

where h follows the definition in Equation (3), M_{idco} contains the probabilities of other samples within the same batch.

3.4. Overall Training Objective

The proposed two losses are complementary to each other and can be trained end-to-end. Since our CR-CACo uses a high prob-threshold (*i.e*. 0.95) to select confident samples, it should be combined with other methods that can generate confident predictions. Otherwise, it will be dominated by noise labels in the early training stage. Although it performs normally when combined with our CR-IDCo loss, both of them focus on transferring source class relationship, and other methods that encourage confident predictions are still needed and complementary to ours. As a result, we combine our method with existing representative methods (*i.e*. SHOT-IM [42], AaD [90]) to validate the effectiveness. As a result, the overall loss is presented as:

$$\ell_{ours} = \ell_{baseline} + \lambda_{caco}\ell_{cr\_caco} + \lambda_{idco}\ell_{cr\_idco} \tag{6}$$

where $\lambda_{caco}, \lambda_{idco}$ are the trade-off hyperparameters. $l_{baseline}$ is the loss of baseline (*i.e.*, IM or AaD). It contains two terms: discriminability and diversity. The former produces high-confident outputs, and the latter avoids collapse of predictions. IM uses entropy of single sample and average of all samples, and AaD uses neighbor information.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate the effectiveness of our approach on four standard DA benchmarks. **Office-31** [63] benchmark consists of three domains in office environments: Amazon (**A**), DSLR (**D**), and Webcam (**W**), each with 31 categories. **Office-Home** [73] is a more challenging dataset. It comprises of images of commonplace objects divided into four domains: Artistic (**Ar**), Clipart (**Cl**), Product (**Pr**), and Real-World (**Rw**), each with 65 classes. **VisDA** [58] is a large-scale dataset for synthetic-to-real domain adaptation. The source domain has 152,397 synthetic images, while the target domain has 55,388 real-world images. **DomainNet** [57] is the most challenging dataset involving 6 domains: Clipart (**C**), Real (**R**), Infograph (**I**), Painting (**P**), Sketch (**S**) and Quickdraw (**Q**) with 345 classes each.

Implementation details. To ensure fair comparison with related methods, we adopt the backbone of a ResNet-50 [19] for Office-31, Office-Home, and DomainNet, and a ResNet-101 for VisDA. We use the same network architecture as SHOT [42] where the final part of the network is changed to suit the SFUDA task. For the classifier, we find that fixing it or not have little impact on the final performance, and we choose to fix it in all experiments. We adopt SGD with momentum 0.9 and batch size 64 for all datasets where each image has three views (*i.e*. one weak and two strong augmentations).

For the hyperparameters, we empirically set the probability threshold τ to 0.95 following FixMatch [67]. We find that $\tau = 0.95$ works well across all settings and tasks. For the temperature T_{co} in both contrastive losses,

Table 1. **Single-Source Unsupervised DA (SUDA)** on Office-Home.

Method	SF	**Office-Home**												
		Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
SENTRY (ICCV'21) [59]	✗	61.8	77.4	80.1	66.3	71.6	74.7	66.8	**63.0**	80.9	74.0	**66.3**	84.1	72.2
FixBi (CVPR'21) [52]	✗	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	**76.4**	62.9	86.7	72.7
SCDA (ICCV'21) [38]	✗	60.7	76.4	82.8	69.8	77.5	78.4	68.9	59.0	82.7	74.9	61.8	84.5	73.1
A^{2}Net (ICCV'21) [79]	✓	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
NRC (NeurIPS'21) [87]	✓	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	85.6	72.2
D-MCD (AAAI'22) [10]	✓	59.4	78.9	80.2	67.2	79.3	78.6	65.3	55.6	82.2	73.3	62.8	83.9	72.2
DIPE (CVPR'22) [74]	✓	56.5	79.2	80.7	70.1	79.8	78.8	67.9	55.1	83.5	74.1	59.3	84.8	72.5
Sub-Sup (ECCV'22) [29]	✓	61.0	80.4	82.5	69.1	79.9	79.5	69.1	57.8	82.7	74.5	65.1	86.4	74.0
BMD (ECCV'22) [61]	✓	58.1	79.7	82.6	69.3	81.0	80.7	**70.8**	57.6	83.6	74.0	60.0	85.9	73.6
U-SFAN (ECCV'22) [62]	✓	57.8	77.8	81.6	67.9	77.3	79.2	67.2	54.7	81.2	73.3	60.3	83.9	71.9
CoWA-JMDS (ICML'22) [33]	✓	56.9	78.4	81.0	69.1	80.0	79.9	67.7	57.2	82.4	72.8	60.5	84.5	72.5
Feat-Mixup (ICML'22) [30]	✓	61.8	81.2	83.0	68.5	80.6	79.4	67.8	61.5	**85.1**	73.7	64.1	86.5	74.5
VMP (NeurIPS'22) [25]	✓	57.9	77.6	82.5	68.6	79.4	80.6	68.4	55.6	83.1	75.2	59.6	84.7	72.8
DaC (NeurIPS'22) [98]	✓	59.1	79.5	81.2	69.3	78.9	79.2	67.4	56.4	82.4	74.0	61.4	84.4	72.8
SHOT-IM (ICML'20) [42]	✓	55.4	76.6	80.4	66.9	74.3	75.4	65.6	54.8	80.7	73.7	58.4	83.4	70.5
Ours+ SHOT-IM	✓	62.8	82.0	84.3	70.9	80.8	**82.6**	70.0	61.1	83.6	76.2	65.1	87.0	75.5 (+5.0)
AaD (NeurIPS'22) [90]	✓	59.3	79.3	82.1	68.9	79.8	79.5	67.2	57.4	83.1	72.1	58.5	85.4	72.7
Ours+ AaD	✓	**63.5**	**82.1**	**85.0**	**73.0**	**82.7**	82.4	69.5	62.9	82.6	74.2	65.7	**87.3**	**75.9** (+3.2)

Table 2. **Single-Source Unsupervised Domain Adaptation (SUDA)** on Office-31 and VisDA benchmarks.

Method	SF	**Office-31**							**VisDA**
		A→D	A→W	D→W	W→D	D→A	W→A	Avg	S → R
BCDM (AAAI'20) [37]	✗	93.8	95.4	98.6	100.0	73.1	73.0	89.0	83.4
MCC (ECCV'20) [24]	✗	94.4	95.5	98.6	100.0	72.9	74.9	89.4	78.8
FixBi (CVPR'21) [52]	✗	95.0	96.1	**99.3**	100.0	78.7	**79.4**	**91.4**	87.2
RADA (ICCV'21) [23]	✗	96.1	**96.2**	**99.3**	100.0	77.5	77.4	91.1	76.3
FAA (ICCV'21) [22]	✗	94.4	92.3	99.2	99.7	**80.5**	78.7	90.8	82.7
SCDA (ICCV'21) [38]	✗	95.4	95.3	99.0	100.0	77.2	75.9	90.5	-
SHOT (ICML'20) [42]	✓	94.0	90.1	98.4	99.9	74.7	74.3	88.6	82.9
CPGA (IJCAI'21) [60]	✓	94.4	94.1	98.4	99.8	76.0	76.6	89.9	84.1
VDM-DA (TCSVT'21) [71]	✓	93.2	94.1	98.0	100.0	75.8	77.1	89.7	85.1
A^{2}Net (ICCV'21) [79]	✓	94.5	94.0	99.2	100.0	76.7	76.1	90.1	84.3
HCL (NeurIPS'21) [21]	✓	90.8	91.3	98.2	100.0	72.7	72.7	87.6	83.5
NRC (NeurIPS'21) [87]	✓	96.0	90.8	99.0	100.0	75.3	75.0	89.4	85.9
SHOT++ (TPAMI'21) [43]	✓	94.3	90.4	98.7	99.9	76.2	75.8	89.2	87.3
D-MCD (AAAI'22) [10]	✓	94.1	93.5	98.8	100.0	76.4	76.4	89.9	87.5
DIPE (CVPR'22) [74]	✓	**96.6**	93.1	98.4	99.6	75.5	77.2	90.1	83.1
Sub-Sup (ECCV'22) [29]	✓	95.6	94.6	99.2	99.8	77.0	77.7	90.7	88.2
BMD (ECCV'22) [61]	✓	96.2	94.2	98.0	100.0	76.0	76.0	90.1	88.7
CoWA-JMDS (ICML'22) [33]	✓	94.4	95.2	98.5	100.0	76.2	77.6	90.3	86.9
Feat-Mixup (ICML'22) [30]	✓	94.6	93.2	98.9	100.0	78.3	78.9	90.7	87.8
VMP (NeurIPS'22) [25]	✓	93.3	**96.2**	98.6	100.0	75.4	76.9	90.0	-
SHOT-IM (ICML'20) [42]	✓	90.6	91.2	98.3	99.9	72.5	71.4	87.3	80.4
Ours + SHOT-IM	✓	95.8	95.1	99.0	**100.0**	76.6	78.3	90.8**(+3.5)**	89.1**(+8.7)**
AaD (NeurIPS'22) [90]	✓	96.4	92.1	99.1	100.0	75.0	76.5	89.9	88.0
Ours+ AaD	✓	**96.6**	95.5	99.1	**100.0**	76.9	78.3	91.1**(+1.2)**	**89.6(+1.6)**

we set it to 0.07 following previous contrastive learning methods [18, 21, 85]. For the λ_{inter} in our similarity, we use 1.0 for simplicity. For the trade-off hyperparameters $\lambda_{caco}, \lambda_{idco}$, we directly set them to 1.0. More training details can be found in the supplementary material.

4.2. Results

The results are shown in Table 1, Table 2, Table 3 and Table 4. In each table, SF indicates *source-free*, * indicates results we reproduced from the released code. **(+x.x)** indicates gains over SHOT-IM [42] and AaD [90] respectively. *Ours* means combination of our proposed two losses.

Single Source Domain Adaptation Table 1 and Table 2 show the classification accuracy for single-source DA on each dataset: Office-Home, Office-31, and VisDA. Based on the simple SHOT-IM and stronger AaD, our method can achieve state-of-the-art performance. UDA methods BCDM [37] and MCC [24] also explore the inter-class relationship by directly suppressing the inter-class term to obtain determined outputs. Our method surpasses these methods even without source domain supervision.

The two proposed contrastive losses exploit different views of the same image. Thus our method is complementary to those not utilizing information of different views. Both baselines only consider weak views, and so our method can boost them. AaD surpasses IM by regarding local neighbors as positive pairs. In our CR-IDCo loss, the strong views of the same image can be regarded as neighbors since they are very close in feature space, which has overlapped effect with AaD. When our method is combined with AaD, therefore, the improvement is relatively smaller.

Multi Source Domain Adaptation Table 3 shows the results for multi-source DA on Office-Home and the large-scale DomainNet benchmark. We adopt two baselines without using domain labels, that is, all data from different source domains is treated as belonging to a single domain and only one source model is trained and transferred. We observe improvements of 4.1% and 3.4% in DomainNet, 4.5% and 3.0% in Office-Home. CAiDA [14] considers class relationship by enforcing consistency of different source model outputs. It achieves 76.2% on Office-Home by combining IM loss, a novel confident-anchor-induced pseudo label method and the class-relationship-aware consistency loss. Based on the same IM loss, ours achieves 78.1% and exceeds it by 1.9%.

Table 3. **Multi-Source Unsupervised Domain Adaptation (MUDA)** on DomainNet and Office-Home.

Method	SF	w/o Domain Labels	DomainNet →C	→I	→P	→Q	→R	→S	Avg	Office-Home →Ar	→Cl	→Pr	→Rw	Avg
MCC (ECCV'20) [24]	✗	✗	65.5	26.0	56.6	16.5	68.0	52.7	47.6	-	-	-	-	-
CMSDA (BMVC'21) [66]	✗	✗	70.9	26.5	57.5	21.3	68.1	59.4	50.4	71.5	**67.7**	84.1	82.9	76.6
DRT (CVPR'21) [39]	✗	✗	71.0	**31.6**	61.0	12.3	71.4	**60.7**	51.3	-	-	-	-	-
STEM (ICCV'21) [53]	✗	✗	72.0	28.2	**61.5**	**25.7**	**72.6**	60.2	**53.4**	-	-	-	-	-
Source-combine	✗	✓	57.0	23.4	54.1	14.6	67.2	50.3	44.4	58.0	57.3	74.2	77.9	66.9
DECISION (CVPR'21) [1]	✓	✗	61.5	21.6	54.6	18.9	67.5	51.0	45.9	74.5	59.4	84.4	83.6	75.5
CAiDA (NeurIPS'21) [14]	✓	✗	-	-	-	-	-	-	-	75.2	60.5	84.7	84.2	76.2
Sub-Sup (ECCV'22) [29]	✓	✓	70.3	25.7	57.3	17.1	69.9	57.1	49.6	75.1	64.1	**86.6**	84.4	77.6
Feat-Mixup (ICML'22) [30]	✓	✓	**75.4**	24.6	57.8	23.6	65.8	58.5	51.0	72.6	67.4	85.9	83.6	77.4
SHOT-IM* (ICML'20) [42]	✓	✓	64.5	24.2	55.2	15.5	67.0	53.2	46.6	71.5	58.6	81.9	82.3	73.6
Ours + SHOT-IM	✓	✓	69.8	27.0	59.0	21.2	70.2	57.5	50.7(+4.1)	74.9	66.4	85.1	85.8	78.1 (+4.5)
AaD* (NeurIPS'22) [90]	✓	✓	65.7	25.5	56.4	17.0	68.1	54.6	47.9	72.1	60.6	85.3	84.4	75.6
Ours +AaD	✓	✓	71.2	27.8	59.3	22.5	69.3	57.8	51.3(+3.4)	**75.4**	66.9	86.4	**85.9**	**78.6** (+3.0)

Table 4. **Partial-set** and **open-set Domain Adaptation** (**PDA** and **ODA**) on Office-Home.

Partial-set DA	SF	Avg.	Open-set DA	SF	Avg.
ResNet-50 [19]	✗	61.3	ResNet [19]	✗	65.3
IWAN (CVPR'18) [92]	✗	63.6	OpenMax (CVPR'16) [4]	✗	66.7
SAN (CVPR'18) [5]	✗	65.3	ATI-λ (ICCV'17) [56]	✗	66.1
ETN (CVPR'19) [6]	✗	70.5	OSBP (ECCV'18) [65]	✗	65.7
SAFN (ICCV'19) [82]	✗	71.8	STA (CVPR'19) [44]	✗	69.5
Source model only	✓	62.8	Source model only	✓	46.6
SHOT (ICML'20) [42]	✓	79.3	SHOT (ICML'20) [42]	✓	72.8
SHOT+HCL (NeurIPS'21) [21]	✓	80.1	SHOT+HCL (NeurIPS'21) [21]	✓	73.2
CoWA-JMDS (ICML'22) [33]	✓	**83.2**	CoWA-JMDS (ICML'22) [33]	✓	73.2
SHOT-IM (ICML'20) [42]	✓	76.8	SHOT-IM (ICML'20) [42]	✓	71.5
Ours +SHOT-IM	✓	80.6 (+3.8)	*Ours +SHOT-IM*	✓	73.2(+1.7)
AaD* (NeurIPS'22) [90]	✓	79.7	AaD* (NeurIPS'22) [90]	✓	71.8
Ours +AaD	✓	82.4 (+2.7)	*Ours +AaD*	✓	**73.3**(+1.5)

Domain Adaptation Beyond Vanilla Closed-set We provide additional results under source-free partial-set and open-set DA (PDA and ODA) setting on Office-Home. For open-set detection in ODA , we follow the same protocol to detect unseen categories as SHOT [42]. Results are shown in Table 4. It can be seen that our method can achieve performance comparable to that of the state-of-the-arts.

4.3. Analysis

Component-wise ablations Here we validate the effectiveness of different proposed losses. As described in Sec. 3.4, we do not use CR-CACo loss alone. The results are shown in Table 5. It can be seen that when using our method alone (line #2) can achieve comparable performance. When combined with SHOT-IM, our both losses are complementary, and the best performance can be achieved by combining baseline and our losses. Note that although CR-IDCo (line #1) outperforms IM (line #0), it is worse than IM when combined with CR-CACo (*i.e.* line #2 $<$ line #3). This is caused by the fact that CR-IDCo cannot produce enough confident samples as IM for the learning of CR-CACo loss.

Effect of pseudo labels in CR-CACo In our proposed CR-CACo loss, the pseudo label is generated from clustering-based method in SHOT [42] instead of predictions of weakly augmented images. The results of the comparison are shown in Table 6. It can be seen that our

Table 5. Component-wise ablation studies of the proposed methods under the SUDA setting.

#	IM	CR-CACo	CR-IDCo	Office-31	Office-Home	VisDA
0	✓			87.3	70.5	80.4
1			✓	89.2	72.5	83.0
2		✓	✓	89.6	74.0	85.3
3	✓	✓		89.8	74.8	87.0
4	✓		✓	90.2	74.3	85.8
5	✓	✓	✓	90.8	75.5	89.1

CR-CACo can bring significant improvement when using pseudo label from model prediction. The clustering based pseudo-label can bring slight but consistent improvements.

Table 6. Comparison of different pseudo label methods.

#	Baseline	Pseudo-label in CR-CACo Prediction	Clutering	Office-31	Office-Home	VisDA
0	IM			87.3	70.5	80.4
1		✓		89.6(+2.3)	74.5(+4.0)	86.6(+6.2)
2			✓	89.8(+2.5)	74.8(+4.3)	87.0(+6.6)
3	AaD			89.9	72.7	88.0
4		✓		90.6(+0.7)	75.0(+2.3)	88.9(+0.9)
5			✓	90.7(+0.8)	75.2(+2.5)	89.1(+1.1)

Different similarities and disentangled improvements. To validate the superiority of our class relationship embedded similarity in contrastive loss, we compare it with other two choices. The first one uses only s_{intra} based on features, and the second one uses only s_{intra} based on probabilities which equals $\lambda_{inter} = 0$ in our case. The results are shown in Table 7. Using features (*i.e. i.e.* s_{intra} w/ feature) rather than probabilities is a more natural way. Indeed, using features in CACo Loss would result in FixMatch loss, and using features in IDCo loss would fall in the traditional self-supervised way (*e.g.*, MoCo). Only using the probability based intra-class term (*i.e.* s_{intra} w/ probability) is still special for similarity in contrastive losses. And it can bring consistent improvements. In both CACo and IDCo losses, our s_{cr} can achieve the best result, thus validating the effectiveness of adopting source prior class similarities.

Table 7. Comparison with different similarities.

#	Loss	Similarity	Office-31	Office-Home	VisDA
0	IM	—	87.3	70.5	80.4
1	IM+CACo	s_{intra} w/ feature	88.9	72.5	83.5
2		s_{intra} w/ probability	89.1	72.9	84.0
3		s_{cr} w/ probability	89.8	74.8	87.0
4	IM+IDCo	s_{intra} w/ feature	87.5	70.9	80.8
5		s_{intra} w/ probability	88.2	71.4	81.5
6		s_{cr} w/ probability	90.2	74.3	85.8

Comparison with fixed coefficient within s_{inter} In the definition of inter-class term in Equation (1), we use the source class similarity matrix $\boldsymbol{A}^s$ as coefficient. Instead of this prior knowledge, we can also set the coefficient to a small fixed value. Considering the positive pairs in contrastive loss, the negative coefficient (*i.e.* -0.1) works to suppress the inter-class term, thus obtaining more determined predictions as in [24, 37] while the positive value (*i.e.* 0.1) is more like a smooth regularization as KLD loss in [100]. The results are shown in Figure 4. It can be seen that both positive and negative values can bring some improvements. However, our CR-CACo loss can achieve better performance, which validates the importance of source prior class relationship.

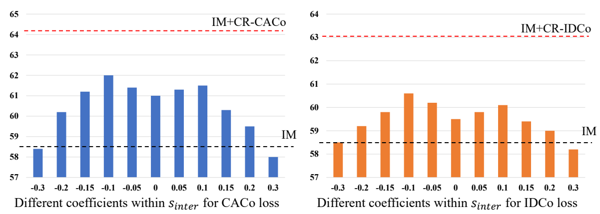

Figure 4. Using fixed coefficients in s_{inter} in both CACo and IDCo losses under SUDA setting Office-Home Rw→Cl. Black dotted line shows the performance of baseline IM. Red dotted line shows the performance of our losses.

Parameter sensitivity Here we analyze the hyperparameter sensitivity under the setting of SUDA Office-Home with SHOT-IM baseline. We consider four hyperparameters, as shown in Figure 5. It is evident that they are stable within specific ranges.

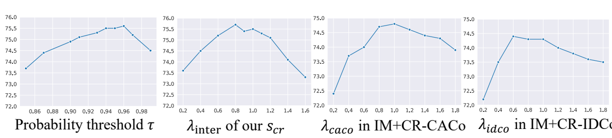

Figure 5. Sensitivity analysis of different hyperparameters. The first two use our two losses. The last two use corresponding loss.

Visualization of features We visualize the target features and classifier weights in Figure 6. It can be seen that our method has less misclassified samples for classes 0, 1, 7 compared with IM. Thus our method can learn more discriminative features.

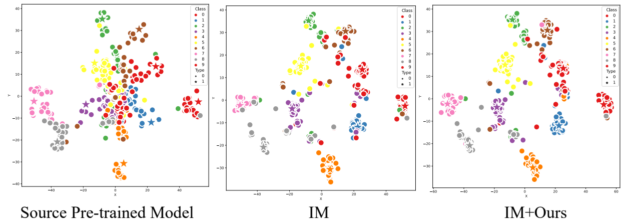

Figure 6. The t-SNE visualization of classifier weights (*i.e.* star) and target domain features (*i.e.* dot) of different methods under the setting of SUDA Office-Home Rw→Cl. We directly use the first 10 classes (*i.e.* different colors), and the coordinates of classifier weights are the same across different methods since source classifier is frozen. Best viewed in color.

Transformer backbone We further apply our method with Transformer-based backbone under the SFUDA setting, as shown in Table 8. We choose the same network (*i.e.* ViT-B [15]) following SSRT [69]. For the training of the source model, we follow SHOT [42] which obtains better performance than original ViT-B (78.2% *vs*. 75.5%). Adding IM loss achieves 84.0%. Based on IM loss, ours achieves 88.3% which significantly outperforms SSRT.

Table 8. **Single-Source Unsupervised DA (SUDA)** on Office-Home with ViT-B backbone.

Method	SF	Avg.	Method	SF	Avg.
ViT-B	✗	75.5	CDTrans (ICLR'22) [83]	✗	80.5
TVT (WACV'23) [86]	✗	83.6	DOT-B (ACMMM'22) [50]	✗	84.3
SSRT (CVPR'22) [69]	✗	85.4	BCAT-DTF (Arxiv'22) [76]	✗	86.6
Source model	✓	78.2	SHOT-IM	✓	84.0
SHOT-IM+CR-CACo	✓	87.0(+3.0)	SHOT-IM+Ours	✓	**88.3**(+4.3)

5. Conclusion

In this paper, we propose to explicitly transfer the class relationship for SFUDA which is more domain-invariant. We propose a novel class relationship embedded similarity that can more accurately express the sample relationship in the output space. Furthermore, we propose two contrastive losses (*i.e.*, CR-CACo and CR-IDCo) that exploit our designed similarity. These two losses are complementary, and their combination can better explore the target distribution. We combine our method with existing representative baselines in multiple SFUDA settings. Extensive experiments show the effectiveness of the proposed method, which achieves state-of-the-art performance.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 62176246 and Grant 61836008. This work is also supported by Anhui Provincial Natural Science Foundation 2208085UD17 and the Fundamental Research Funds for the Central Universities (WK3490000006).

References

[1] Sk. Miraj Ahmed, Dripta S. Raychaudhuri, S. Paul, Samet Oymak, and Amit K. Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *CVPR*, 2021. 7

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

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NeurIPS*, 2006. 2

[4] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016. 7

[5] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *CVPR*, 2018. 7

[6] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *CVPR*, 2019. 7

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[8] Chaoqi Chen, Weiping Xie, Tingyang Xu, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, 2019. 3

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 4, 5

[10] Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, and Lixin Duan. Denoised maximum classifier discrepancy for sourcefree unsupervised domain adaptation. In *AAAI*, 2022. 3, 6

[11] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *ICCV*, 2019. 2

[12] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020. 2

[13] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, 2019. 3

[14] Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. In *NeurIPS*, 2021. 2, 6, 7

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 8

[16] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018. 3

[17] Yuan Gao, Zilei Wang, Jiafan Zhuang, Yixin Zhang, and Junjie Li. Exploit domain-robust optical flow in domain adaptive video semantic segmentation. In *AAAI*, 2023. 3

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3, 4, 5, 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5, 7

[20] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *ICML*, 2017. 3

[21] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *NeurIPS*, 2021. 1, 2, 3, 4, 6, 7

[22] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. RDA: Robust domain adaptation via fourier adversarial attacking. In *ICCV*, 2021. 6

[23] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Re-energizing domain discriminator with sample relabeling for adversarial domain adaptation. In *ICCV*, 2021. 6

[24] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *ECCV*, 2020. 6, 7, 8

[25] Mengmeng Jing, Xiantong Zhen, Jingjing Li, and Cees GM Snoek. Variational model perturbation for source-free domain adaptation. In *NeurIPS*, 2022. 3, 6

[26] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019. 3, 4

[27] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *CVPR*, 2019. 3

[28] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *ICML*, 2020. 3

[29] 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 *ECCV*, 2022. 3, 6, 7

[30] Jogendra Nath Kundu, Akshay R Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Anand Kulkarni, Varun Jampani, and Venkatesh Babu Radhakrishnan. Balancing discriminability and transferability for source-free domain adaptation. In *ICML*, 2022. 3, 6, 7

[31] Jogendra Nath Kundu, Naveen Venkat, and R Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, 2020. 3

[32] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, Rahul M V, and R. Venkatesh Babu. Towards inheritable models for open-set domain adaptation. In *CVPR*, 2020. 3

[33] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *ICML*, 2022. 3, 6, 7

[34] Junjie Li, Zilei Wang, Yuan Gao, and Xiaoming Hu. Exploring high-quality target domain information for unsupervised domain adaptive semantic segmentation. In *ACMMM*, 2022. 3

[35] Junjie Li, Yixin Zhang, Zilei Wang, and Keyu Tu. Semantic-aware representation learning via probability contrastive loss. *arXiv preprint arXiv:2111.06021*, 2021. 5

[36] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 1, 2, 3

[37] Shuang Li, Fangrui Lv, Binhui Xie, Chi Harold Liu, Jian Liang, and Chen Qin. Bi-classifier determinacy maximization for unsupervised domain adaptation. In *AAAI*, 2021. 6, 8

[38] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *ICCV*, 2021. 6

[39] Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and N. Vasconcelos. Dynamic transfer for multi-source domain adaptation. In *CVPR*, 2021. 7

[40] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019. 3

[41] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *CVPR*, 2022. 4

[42] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[43] 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. 6

[44] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, 2019. 7

[45] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, 2019. 2

[46] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In *NeurIPS*, 2021. 3

[47] Qinying Liu and Zilei Wang. Collaborating domain-shared and target-specific feature clustering for cross-domain 3d action recognition. In *ECCV*, 2022. 3

[48] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. 2

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

[50] Wenxuan Ma, Jinming Zhang, Shuang Li, Chi Harold Liu, Yulin Wang, and Wei Li. Making the best of both worlds: A domain-oriented transformer for unsupervised domain adaptation. In *ACMMM*, 2022. 8

[51] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020. 3

[52] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. FixBi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021. 6

[53] Van-Anh Nguyen, Tuan Nguyen, Trung Le, Quan Hung Tran, and Dinh Phung. STEM: An approach to multi-source domain adaptation with guarantees. In *ICCV*, 2021. 7

[54] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, 2020. 3

[55] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, 2019. 3

[56] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV*, 2017. 7

[57] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2, 5

[58] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. VisDA: The visual domain adaptation challenge. In *CVPRW*, 2018. 5

[59] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. SENTRY: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *ICCV*, 2021. 6

[60] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. In *IJCAI*, 2021. 6

[61] 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 *ECCV*, 2022. 3, 6

[62] Subhankar Roy, Martin Trapp, Andrea Pilzer, Juho Kannala, Nicu Sebe, Elisa Ricci, and Arno Solin. Uncertainty-guided source-free domain adaptation. In *ECCV*, 2022. 3, 6

[63] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 5

[64] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 2, 3

[65] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *ECCV*, 2018. 7

[66] Marin Scalbert, Maria Vakalopoulou, and Florent Couzini'e-Devy. Multi-source domain adaptation via supervised contrastive learning and confident consistency regularization. In *BMVC*, 2021. 7

[67] 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. 2, 4, 5

[68] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016. 2

[69] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *CVPR*, 2022. 8

[70] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 2020. 3

[71] Jiayi Tian, Jing Zhang, Wen Li, and Dong Xu. VDM-DA: Virtual domain modeling for source data-free domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1, 6

[72] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

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

[74] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *CVPR*, 2022. 3, 6

[75] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021. 2

[76] Xiyu Wang, Pengxin Guo, and Yu Zhang. Domain adaptation via bidirectional cross-attention transformer. *arXiv preprint arXiv:2201.05887*, 2022. 8

[77] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI*, 2019. 2

[78] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3

[79] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, 2021. 1, 2, 3, 6

[80] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, 2018. 3

[81] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *CVPR*, 2020. 3

[82] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, 2019. 7

[83] Tongkun Xu, Weihua Chen, WANG Pichao, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *International Conference on Learning Representations*, 2021. 8

[84] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, 2017. 2

[85] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-aware contrastive semi-supervised learning. In *CVPR*, 2022. 2, 4, 6

[86] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation. In *WACV*, 2023. 8

[87] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *NeurIPS*, 2021. 1, 2, 3, 6

[88] 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. 1

[89] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *ICCV*, 2021. 2, 3

[90] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, and Joost van de Weijer. Attracting and dispersing: A simple approach for source-free domain adaptation. In *NeurIPS*, 2022. 2, 3, 5, 6, 7

[91] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS*, 2021. 2

[92] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, 2018. 7

[93] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021. 3, 4

[94] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, 2018. 3

[95] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, 2018. 1

[96] Yixin Zhang and Zilei Wang. Joint adversarial learning for domain adaptation in semantic segmentation. In *AAAI*, 2020. 2

[97] Yixin Zhang, Zilei Wang, and Yushi Mao. Rpn prototype alignment for domain adaptive object detector. In *CVPR*, 2021. 3

[98] Ziyi Zhang, Weikai Chen, Hui Cheng, Zhen Li, Siyuan Li, Liang Lin, and Guanbin Li. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. In *NeurIPS*, 2022. 6

[99] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *CVPR*, 2020. 3

[100] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. 8

[101] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 3