Alleviating Imbalanced Pseudo-label Distribution: Self-Supervised Multi-Source Domain Adaptation with Label-specific Confidence

Shuai Lü[1,2,3] , **Meng Kang**[1,2] and **Ximing Li**[1,2,3,*]
[1]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, China
[2]College of Computer Science and Technology, Jilin University, China
[3]College of Software, Jilin University, China
lus@jlu.edu.cn, kangmeng20@mails.jlu.edu.cn, liximing86@gmail.com

Abstract

The existing self-supervised Multi-Source Domain Adaptation (MSDA) methods often suffer an imbalanced characteristic among the distribution of pseudo-labels. Such imbalanced characteristic results in many labels with too many or too few pseudo-labeled samples on the target domain, referred to as easy-to-learn label and hard-to-learn label, respectively. Both of these labels hurt the generalization performance on the target domain. To alleviate this problem, in this paper we propose a novel multi-source domain adaptation method, namely Self-Supervised multi-Source Domain Adaptation with Label-specific Confidence (S^{3}DA-LC). Specifically, we estimate the label-specific confidences, *i.e.,* the learning difficulties of labels, and adopt them to generate the pseudo-labels for target samples, enabling to simultaneously constrain and enrich the pseudo supervised signals for easy-to-learn and hard-to-learn labels. We evaluate S^{3}DA-LC on several benchmark datasets, indicating its superior performance compared with the existing MSDA baselines.

1 Introduction

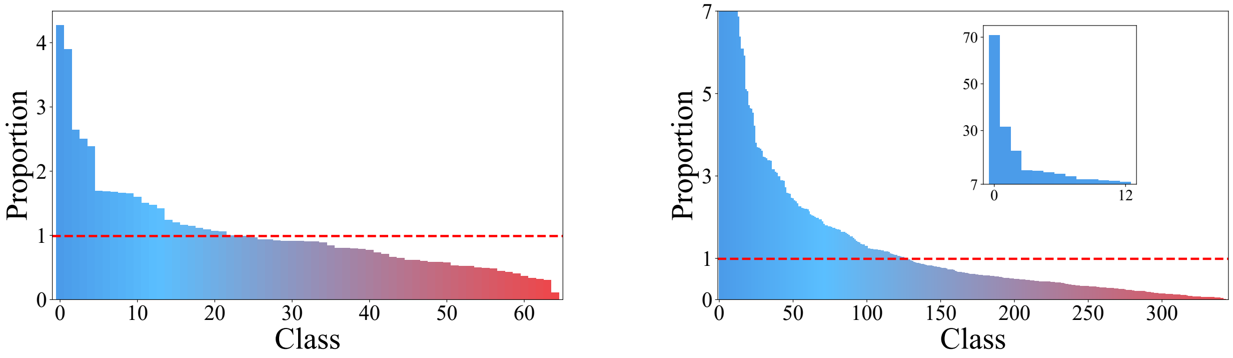
Unsupervised **D**omain **A**daptation (**UDA**) refers to the goal of transferring the task knowledge from the labeled source data to the unlabeled target data under a domain-shift, enabling to reduce the demand for costly labeled samples in the target domain [Tan *et al.*, 2018; Wang and Deng, 2018]. Commonly, the previous studies mainly focus on the **S**ingle-**S**ource **D**omain **A**daptation (**SSDA**), which consists of only one source domain and one target domain [Zhao *et al.*, 2020c]. However, the situation is often violated in real-world scenarios, since the available labeled samples are often from different domains. Accordingly, only applying one specific source domain may result in sub-optimal solutions [Zhao *et al.*, 2018]. Motivated by handling the problem, increasingly more attention has been recently paid to the topic of **M**ulti-**S**ource **D**omain **A**daptation (**MSDA**) within the machine learning community [Zhao *et al.*, 2020a].

To our knowledge, the common framework of the existing MSDA methods consists of two key components: learning the models on the source domains and incorporating target samples to fine-tune the models with various alignment strategies. Generally speaking, the alignment strategies are key to improving the generalization performance of the MSDA methods, and nowadays the representatives mainly include minimizing the statistical discrepancy between source and target domains [Wen *et al.*, 2020; Guo *et al.*, 2020; Feng *et al.*, 2021], adversarial learning with a domain discriminator [Zhao *et al.*, 2018; Xu *et al.*, 2018; Zhao *et al.*, 2020b], and incorporating auxiliary self-supervised learning tasks [Venkat *et al.*, 2020; Kang *et al.*, 2022; Ahmed *et al.*, 2021]. Among them, the self-supervised methods are flexible, straightforward, and easy-to-implement, thus receiving increasing concern. For instance, a common way is to generate pseudo-labels for target samples by using ensemble classification confidences, *e.g.,* Deep CockTail Network (DCTN) [Xu *et al.*, 2018] or nearest neighbor assignments, *e.g.,* Data frEe multi-sourCe unsupervISed domain adaptatION (DECISION) [Ahmed *et al.*, 2021] and Muti-Source Contrastive Adaptation Network (MSCAN) [Kang *et al.*, 2022]; and Self-supervised Implicit Alignment (SImpAl) [Venkat *et al.*, 2020] generates pseudo-labels of target samples by the agreements of a set of shared classifiers.

Unfortunately for self-supervised MSDA methods, the distribution of pseudo-labels often has an imbalanced characteristic, where, specifically, on the target domain there are many labels with too many or too few pseudo-labeled samples, formally referred to as **easy-to-learn labels** and **hard-to-learn labels**. As depicted in Fig.1, we illustrate several examples of pseudo-label distributions learned by SImpAl [Venkat *et al.*, 2020]. Technically, the label whose proportion is above one can be considered as an easy-to-learn label, while the label whose proportion is below one can be considered as a hard-to-learn label. Both of these labels hurt the generalization performance on the target domain, especially the leftmost and rightmost labels.

In this paper, we aim to alleviate the aforementioned problem of self-supervised MSDA methods, and then develop a novel straightforward-yet-effective MSDA method, namely **S**elf-**S**upervised multi-**S**ource **D**omain **A**daptation with **L**abel-specific **C**onfidence (**S^{3}DA-LC**). The basic idea of S^{3}DA-LC is to consider the learning difficulty of each la-

[*]Corresponding author.



(a) →*Cl* (*Office-Home*)

(b) →*Inf* (*DomainNet*)

Figure 1: The examples of pseudo-label distributions learned by SImpAl. The vertical axis refers to the proportion between pseudo-label distribution and true label distribution on the target domain, where label distribution is the proportions between the number of samples per-label and all samples.

bel on the target domain when generating pseudo-labels of target samples, so as to simultaneously constrain and enrich the pseudo supervised signals for easy-to-learn labels and hard-to-learn labels. To be specific, our S^3DA-LC consists of two training stages, *i.e.,* **warm-up stage** and **adaptation stage**. In the warm-up stage, we mainly train domain-specific classifiers for each source domain. In the adaptation stage, we employ the pre-trained classifiers to compute the ensemble predictions for target samples. We then compute the **label-specific confidences**, *i.e.,* learning difficulty of each label on the target domain, by measuring the difference between the ensemble prediction distribution and the label distributions of all source domains. With them, we can determine pseudo-labeled target samples and then formulate a self-supervised objective to further fine-tune the classifiers in a unified framework, where we alternatively update label-specific confidences and fine-tune the classifiers. To empirically examine S^3DA-LC, we compare it against the existing MSDA methods on several benchmark datasets.

To sum up, we list the main contributions below:

- We propose a novel straightforward-yet-effective MSDA method named S^3DA-LC.
- We propose label-specific confidences to alleviate the imbalanced characteristic of pseudo-label distribution.
- Empirical results indicate that S^3DA-LC can outperform the existing MSDA methods and achieves the state-of-the-art results.

2 Related Work

Unsupervised domain adaptation. UDA methods aim at learning a domain-agnostic model from the labeled source domain and apply it to the unlabeled target domain. A common way is to minimize a distance measure in a domain-invariant feature space [Long *et al.*, 2015; Long *et al.*, 2017], and other existing methods learn a latent shared feature space across domains by adversarial learning [Ganin and Lempitsky, 2015]. Recent studies jointly learn with the source domain and target domain with pseudo supervised signals in a self-supervised manner to reduce the domain shift [Saito *et al.*, 2020; Liang *et al.*, 2020]. However, the aforementioned methods mainly handle the single-source-single-target adaptation, which may result in sub-optimal solutions [Zhao *et al.*, 2018].

Multi-source domain adaptation. Beyond the typical UDA, MSDA aims to incorporate task knowledge from multiple source domains. To our knowledge, the self-supervised MSDA methods with pseudo-labeling are the representatives in the community [Xu *et al.*, 2018; Venkat *et al.*, 2020; Ahmed *et al.*, 2021]. For example, DCTN [Xu *et al.*, 2018] assigns pseudo-labels to target samples with high confidence with a fixed threshold, and combines multi-way adversarial learning to train the model. SImpAl [Venkat *et al.*, 2020] trains a set of shared classifiers on source domains and generates pseudo-labels of target samples by the prediction agreements of shared classifiers, then alternately trains the model on source domains and the target domain to align the distributions. DECISION [Ahmed *et al.*, 2021] adopts self-supervised clustering with the combinations of features and predictions, and generates pseudo-labels by measuring the distances between target samples and clustering prototypes. However, those self-supervised MSDA methods take no account of the learning difficulties of different labels when generating pseudo-labels. In contrast, our S^3DA-LC alleviates this problem by estimating the label-specific confidences to generate more precise self-supervised signals.

Pseudo-labeling with confidence threshold. The pseudo-labeling is one prevalent methodology to solve learning tasks with limited supervision [Xu *et al.*, 2018; Li *et al.*, 2020; Li and Wang, 2020; Ahmed *et al.*, 2021]. Kundu et al. [Kundu *et al.*, 2020] select top-*k* percentile target instances based on classifier confidence, and enforce the target predictions to match the pseudo-labels for these instances using cross-entropy. Confident-Anchor-induced multisource-free Domain Adaptation (CAiDA) [Dong *et al.*, 2021] develops a confident-anchor-induced pseudo-label generator to mine pseudo-labels for the unlabeled target data, by incorporating with the quantified source transferability contributions. Besides DA, applying a fixed threshold to select unlabeled samples with high confidence has been also studied in semi-supervised learning [Sohn *et al.*, 2020; Xie *et al.*, 2020]. Beyond applying a fixed threshold for all labels, a recent work of [Zhang *et al.*, 2021] proposes to use different thresholds for different labels. For each label, its threshold describes the learning difficulty, and it is estimated by the number of predicted samples. To some extent, it is built on a balanced assumption of labels, which may be violated in many real applications. Our S^{3}DA-LC also estimates different thresholds for labels, *i.e.,* label-specific confidences. But, inspired by ReMixMatch [Berthelot *et al.*, 2020] where it adjusts predict label distributions according to the ratio between the average model predictions on unlabeled data and the empirical ground-truth label distribution, S^{3}DA-LC employs the label distributions of all source domains, which are exactly known, potentially leading to more precise estimations of label-specific thresholds.

3 S^{3}DA-LC

In this section, we introduce the proposed MSDA method named **S^{3}DA-LC** in more details.

Formulation. Generally, let $\mathcal{X}$ and $\mathcal{Y} = \{1, ..., C\}$ denote the sample and label spaces, respectively. We are given by K source domains $\{\mathcal{D}_{\mathcal{S}}^k\}_{k=1}^K$ and one target domain $\mathcal{D}_{\mathcal{T}}$, where each source domain $\mathcal{D}_{\mathcal{S}}^k$ contains N_k labeled samples denoted by $\mathcal{D}_{\mathcal{S}}^k = \{(\mathbf{x}_i^k, \mathbf{y}_i^k) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{N_k}$, and the target domain includes $N_{\mathcal{T}}$ unlabeled samples denoted by $\mathcal{D}_{\mathcal{T}} = \{\mathbf{x}_i^t \in \mathcal{X}\}_{i=1}^{N_{\mathcal{T}}}$. The target domain shares a same label set $\mathcal{Y}$ with the source domains. The objective of MSDA is to learn a predictive model on the target domain by transferring the knowledge from source domains.

Overview. The main idea of S^{3}DA-LC is to estimate **label-specific confidences** to describe the learning difficulty of each label on the target domain, and then use them as adaptive thresholds to generate more precise pseudo-labels for target samples. Specifically, S^{3}DA-LC consists of two training stages, *i.e.,* **warm-up stage** and **adaptation stage**. In the warm-up stage, we jointly train a shared feature extractor $g(\mathbf{x}; \mathbf{\Theta}) : \mathcal{X} \to \mathbb{R}^d$ and K domain-specific classifiers $\{h^k(g(\mathbf{x}); \mathbf{W}^k) : \mathbb{R}^d \to \mathbb{R}^C\}_{k=1}^K$ on source domains. We then initialize the transferability weights $\{w^k\}_{k=1}^K$ of domain-specific classifiers on the target domain and the weighted ensemble predictions $\{p(\mathbf{x}_i^t) \in \Delta^{C-1}\}_{i=1}^{N_{\mathcal{T}}}$ for target samples. In the adaptation stage, we estimate the label-specific confidences $\{\delta_c\}_{c=1}^C$ on the target domain. We generate pseudo-labels $\{\hat{\mathbf{y}}_i^t\}_{i=1}^{N_{\mathcal{T}}}$ for target samples by using $\{p(\mathbf{x}_i^t)\}_{i=1}^{N_{\mathcal{T}}}$ and $\{\delta_c\}_{c=1}^C$, and then continue to train the feature extractor $g(\mathbf{x}; \mathbf{\Theta})$ and domain-specific classifiers $\{h^k(g(\mathbf{x}); \mathbf{W}^k)\}_{k=1}^K$ in a self-supervised paradigm. For each test sample, we predict it by using the ensemble prediction of those classifiers.

3.1 Warm-up

In the warm-up stage, we first train a feature extractor $g(\mathbf{x}; \mathbf{\Theta})$ shared by all source domains and K domain-specific classifiers $\{h^k(g(\mathbf{x}); \mathbf{W}^k)\}_{k=1}^K$ simultaneously. For each domain $\mathcal{D}_{\mathcal{S}}^k$, its classifier objective is formulated below:

$$\mathcal{L}_s^k(\mathbf{\Theta}, \mathbf{W}^k) = \mathbb{E}_{(\mathbf{x}^k, \mathbf{y}^k) \in \mathcal{D}_{\mathcal{S}}^k} \Big[\ell\big(\mathbf{y}^k, \sigma(h^k(g(\mathbf{x}^k)))\big)\Big] \quad (1)$$

where $\ell(\cdot)$ is the cross-entropy loss and $\sigma(\cdot)$ denotes the softmax function.

Combining with the classifiers of all source domains, the overall training objective of the warm-up stage is:

$$\frac{1}{K} \sum\nolimits_{k=1}^{K} \mathcal{L}_s^k(\mathbf{\Theta}, \mathbf{W}^k) \quad (2)$$

Initialization of $\{w^k\}_{k=1}^K$ and $\{p(\mathbf{x}_i^t)\}_{i=1}^{N_{\mathcal{T}}}$. After a warm-up learning with Eq.(2), we are given by pre-trained $g(\mathbf{x}; \mathbf{\Theta})$ and $\{h^k(g(\mathbf{x}); \mathbf{W}^k)\}_{k=1}^K$. With them, we can predict target samples and further use the ensemble predictions to generate pseudo-labels.

To achieve more precise ensemble predictions, we first estimate the transferability weights $\{w^k\}_{k=1}^K$ of domain-specific classifiers, which describe the transfer difficulty of each source-target pair. We take inspiration from the information theory, where, for each domain-specific classifier, its transferability weight can be measured by the entropy of its predictions on the target domain, specifically formulated below:

$$\eta^k = \frac{1}{\mathbb{E}_{\mathbf{x}^t \in \mathcal{D}_{\mathcal{T}}} \mathcal{H}\big(\sigma(h^k(g(\mathbf{x}^t)))\big)} \quad (3)$$

where $\mathcal{H}(\cdot)$ is the Shannon entropy. Shannon entropy can reflect the confidence of a classifier on the target samples where the more certain the classifier is, the lower the entropy. When the classifier of a source domain is more certain about the target samples, this source domain can be seen as more similar to the target domain, and vice versa. Its final transferability weight can be measured by the inverse of Shannon entropy which is as follows:

$$w^k = \frac{\eta^k}{\max \{\eta^i\}_{i=1}^K} \quad (4)$$

Given those transferability weights, we can compute a weighted ensemble prediction for each target sample $\mathbf{x}^t$ as follows:

$$p(\mathbf{x}^t) = \sigma\Big(\frac{1}{K} \sum_{k=1}^{K} \big[w^k \cdot h^k(g(\mathbf{x}^t))\big]\Big) \quad (5)$$

3.2 Adaptation with Pseudo-labeled Samples

In the adaptation stage, we generate pseudo-labels for target samples by using label-specific confidences, and then update the model with them in a self-supervised manner.

Label-specific confidences. The existing self-supervised MSDA methods generate pseudo-labeled target samples by measuring whether their ensemble predictions are greater than a fixed confidence threshold, so as to maintain target samples with high confidences [Xu *et al.*, 2018; Sohn *et al.*, 2020; Xie *et al.*, 2020]. However, such way neglects the learning difficulties of different labels, resulting in many easy-to-learn and hard-to-learn labels as illustrated in Fig.1.

To alleviate this problem, we propose label-specific confidences used as an adaptive pseudo-labeled confidence threshold for each label on the target domain.

During the warm-up stage, we train a unified feature extractor across multiple source domains. This stage aligns the features from the source domains, enabling the extractor to capture domain-invariant feature, which alleviates the distribution shift between the source and target domains. And, since in the scenario of MSDA the source domains and target domain share a same label set $\mathcal{Y}$, the label distribution of all source domains can serve as a guide for generating pseudo-labels in target domain.

The mean of label distributions of source domains $\mathcal{G}$ is:

$$\mathcal{G} = \left\{ \mathbb{E}_k \frac{n_c^k}{N_k},\ c \in \mathcal{Y} \right\} \tag{6}$$

where n_c^k denotes the number of samples belonging to label c in the source domain $\mathcal{D}_{\mathcal{S}}^k$.

In parallel, we can estimate the distribution of ensemble prediction over the target domain as follows:

$$\mathcal{V} = \left\{ \frac{\sum_{\mathbf{x}^t \in \mathcal{D}_\mathcal{T}} \mathbf{1}(\mathrm{argmax}(p(\mathbf{x}^t)) = c)}{N_\mathcal{T}}, c \in \mathcal{Y} \right\} \tag{7}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Accordingly, we can measure the learning difficulty of each label, *i.e.,* label-specific confidence, by the difference between $\mathcal{G}$ and $\mathcal{V}$, specifically formulated below:

$$\delta_c = \min\{1, \frac{\mathcal{V}_c/\mathcal{G}_c}{\mathbf{\Phi}}\} \cdot \tau, c \in \mathcal{Y} \tag{8}$$

where τ is a scaling parameter used to adjust the confidence; and $\mathbf{\Phi} = Q_3 + \lambda \cdot (Q_3 - Q_1)$ is the upper range of Tukey's fences [Tukey, 1977] [1]. We use $\mathbf{\Phi}$ to normalize $\mathcal{V}/\mathcal{G}$ to avoid the potential negative effect of outliers, where Q_1 and Q_3 represent the lower and upper quartile of $\mathcal{V}/\mathcal{G}$ respectively and λ is a coefficient.

From the label-specific confidence δ_c, the labels with lower (or higher) values of $\mathcal{V}_c/\mathcal{G}_c$ correspond to lower (or higher) thresholds, so as to generate pseudo-labeled target samples with high-confidences.

Self-supervised training. Given the label-specific confidences $\{\delta_c\}_{c=1}^C$, we determine whether a target sample $\mathbf{x}^t$ is associated with a pseudo-label with the following formula:

$$\mathcal{A}(\mathbf{x}^t) = \mathbf{1}\Big(\max\big(p(\mathbf{x}^t)\big) \geq \delta_{\hat{\mathbf{y}}^t} \Big) \tag{9}$$

where $\hat{\mathbf{y}}^t = \mathrm{argmax}\big(p(\mathbf{x}^t)\big)$ is treated as the pseudo-label and $\mathcal{A}(\mathbf{x}^t)$ denotes the corresponding pseudo-label indicator.

We then formulate a self-supervised objective with pseudo-labeled target samples as follows:

$$\mathcal{L}_t\big(\mathbf{\Theta}, \{\mathbf{W}^k\}_{k=1}^K\big) = \mathbb{E}_{\mathbf{x}^t \in \mathcal{D}_\mathcal{T}} \Big[\mathcal{A}(\mathbf{x}^t) \cdot \ell\big(\hat{\mathbf{y}}^t, p(\mathbf{x}^t)\big) \Big] \tag{10}$$

By combining Eq.(10) and the cross-entropy losses on source domains, the overall objective of the adaptation stage is given by:

$$\mathcal{L}_t\big(\mathbf{\Theta}, \{\mathbf{W}^k\}_{k=1}^K\big) + \frac{1}{K} \sum_{k=1}^K \mathcal{L}_s^k(\mathbf{\Theta}, \mathbf{W}^k) \tag{11}$$

Algorithm 1 The full training process of S^{3}DA-LC

```
Input: $K$ source domains $\{\mathcal{D}_\mathcal{S}^k\}_{k=1}^K$ and target domain $\mathcal{D}_\mathcal{T}$; scaling parameter $\tau$.
Output: feature extractor $g(\mathbf{x}; \mathbf{\Theta})$ and domain-specific classifiers $\{h^k(g(\mathbf{x}); \mathbf{W}^k)\}_{k=1}^K$.

   // Warm-up stage
1: while not converge do
2:     Update $\mathbf{\Theta}$ and $\{\mathbf{W}^k\}_{k=1}^K$ by optimizing Eq.(2).
3: end while
4: Initialize $\{w^k\}_{k=1}^K$ by using Eqs.(3) and (4), and $\{p(\mathbf{x}_i^t)\}_{i=1}^{N_\mathcal{T}}$ by using Eq.(5).

   // Adaptation stage
5: Initialize $\{\delta_c\}_{c=1}^C$ by using Eq.(8).
6: while not converge do
7:     Update $\mathbf{\Theta}$ and $\{\mathbf{W}^k\}_{k=1}^K$ by optimizing Eq.(11).
8:     Update $\{w^k\}_{k=1}^K$ and $\{\delta_c\}_{c=1}^C$ by using Eq.(12) every epoch.
9: end while
```

Training summary. We now review and summarize the full training process of S^{3}DA-LC where the full training process of S^{3}DA-LC is outlined in Algorithm 1. In the warm-up stage, we train the shared feature extractor $g(\mathbf{x}; \mathbf{\Theta})$ and K domain-specific classifiers $\{h^k(g(\mathbf{x}); \mathbf{W}^k)\}_{k=1}^K$ on the source domains by optimizing Eq.(2) until convergence (**lines 1-3**), and then initialize the transferability weight $\{w^k\}_{k=1}^K$ by using Eqs.(3) and (4) and the weighted ensemble predictions $\{p(\mathbf{x}_i^t)\}_{i=1}^{N_\mathcal{T}}$ by using Eq.(5) (**line 4**). In the adaptation stage, we initialize the label-specific confidences $\{\delta_c\}_{c=1}^C$ by using Eq.(8) (**line 5**). With them, we generate the pseudo-labels of target samples and fine-tune $g(\mathbf{x}; \mathbf{\Theta})$ and $\{h^k(g(\mathbf{x}); \mathbf{W}^k)\}_{k=1}^K$ in a self-supervised manner by optimizing Eq.(11) (**line 7**). At each epoch, we update $\{w^k\}_{k=1}^K$ and $\{\delta_c\}_{c=1}^C$ using a moving average formula (**line 8**):

$$\begin{aligned} w^k &\leftarrow \alpha \cdot w_{old}^k + (1-\alpha) \cdot w^k \\ \delta_c &\leftarrow \alpha \cdot \delta_{c\ old} + (1-\alpha) \cdot \delta_c \end{aligned} \tag{12}$$

where $\alpha = N_\mathcal{P}/N_\mathcal{T}$ is a scaling parameter, and $N_\mathcal{P} = \sum_{\mathbf{x}^t \in \mathcal{D}_\mathcal{T}} \mathcal{A}(\mathbf{x}^t)$.

4 Experiments

Datasets. In the experiments, we evaluate S^{3}DA-LC on 3 benchmark datasets: **(1)** *Office-31* [Saenko *et al.*, 2010] contains 31 classes and 4,652 images unevenly spreading in three visual domains Amazon (*A*), DSLR (*D*), Webcam (*W*); **(2)** *Office-Home* [Venkateswara *et al.*, 2017] contains 65 classes and about 15,500 images from 4 domains: Art (*Ar*), Clipart (*Cl*), Product (*Pr*) and Real-World (*Rw*); **(3)** *DomainNet* [Peng *et al.*, 2019] contains 345 classes and over 600K images from 6 domains: Clipart (*Clp*), Infograph (*Inf*), Painting (*Pnt*), Quickdraw (*Qdr*), Real (*Rel*) and Sketch (*Skt*).

Implementation details. We adopt ResNet-50 [He *et al.*, 2016] as the backbone network for *Office-31* and *Office-*

[1] Tukey's fences is a statistical method for pinpointing outliers by setting up thresholds, or "fences".

Standard	Method	*Office-31*				*Office-Home*				
		→*A*	→*W*	→*D*	Avg	→*Ar*	→*Cl*	→*Pr*	→*Rw*	Avg
Single Best	DAN [Long *et al.*, 2015]	66.7	96.8	99.5	87.7	68.2	56.5	80.3	75.9	70.2
	D-CORAL [Sun and Saenko, 2016]	65.3	98.0	99.7	87.7	67.0	53.6	80.3	76.3	69.3
	MCD [Saito *et al.*, 2018]	69.7	98.5	100.0	89.4	69.1	52.2	79.6	75.1	69.0
Single Combine	DAN [Long *et al.*, 2015]	67.6	97.8	99.6	88.3	68.5	59.4	79.0	82.5	72.4
	D-CORAL [Sun and Saenko, 2016]	67.1	98.0	99.3	88.1	68.1	58.6	79.5	82.7	72.2
	MCD [Saito *et al.*, 2018]	68.5	99.3	99.4	89.0	67.8	59.9	79.2	80.9	71.9
Multi-Source	SImpAl [Venkat *et al.*, 2020]	70.6	97.4	99.2	89.0	70.8	56.3	80.2	81.5	72.2
	CAiDA [Dong *et al.*, 2021]	75.8	98.9	99.8	91.6	75.2	60.5	84.7	84.2	76.2
	DECISION [Ahmed *et al.*, 2021]	75.4	98.4	99.6	91.1	74.5	59.4	84.4	83.6	75.5
	SPS [Wang *et al.*, 2022]	73.8	**99.3**	**100.0**	91.0	75.1	66.0	84.4	84.2	77.4
	S^{3}DA-LC	**78.1**	99.0	**100.0**	**92.4**	**78.1**	**70.0**	**87.4**	**87.2**	**80.7**

Table 1: Experimental results on *Office-31* and *Office-Home*. The best scores are indicated in **bold**.

Standard	Method	→*Clp*	→*Inf*	→*Pnt*	→*Qdr*	→*Rel*	→*Skt*	Avg
Single Best	DAN [Long *et al.*, 2015]	39.1	11.4	33.3	16.2	42.1	29.7	28.6
	MCD [Saito *et al.*, 2018]	42.6	19.6	42.6	3.8	50.5	33.8	32.2
	CAN [Kang *et al.*, 2019]	63.8	24.0	55.7	27.1	67.7	51.9	48.4
Single Combine	DAN [Long *et al.*, 2015]	45.4	12.8	36.2	15.3	48.6	34.0	32.1
	MCD [Saito *et al.*, 2018]	54.3	22.1	45.7	7.6	58.4	43.5	38.5
	CAN [Kang *et al.*, 2019]	67.4	25.3	56.2	26.3	72.5	56.2	50.7
Multi-Source	DCTN [Xu *et al.*, 2018]	48.6	23.5	48.8	7.2	53.5	47.3	38.2
	SImpAl [Venkat *et al.*, 2020]	66.4	26.5	56.6	18.9	68.0	55.5	48.6
	MSCAN [Kang *et al.*, 2022]	69.3	28.0	58.6	**30.3**	73.3	59.5	53.2
	KD3A [Feng *et al.*, 2021]	**72.5**	23.4	60.9	16.4	72.7	60.6	51.1
	STEM [Nguyen *et al.*, 2021]	72.0	28.2	**61.5**	25.7	72.6	60.2	53.4
	SPS [Wang *et al.*, 2022]	70.8	24.6	55.2	19.4	67.5	57.6	49.2
	S^{3}DA-LC	71.9	**31.3**	61.3	27.8	**75.7**	**61.2**	**54.8**

Table 2: Experimental results on *DomainNet*. The best scores are indicated in **bold**.

Home, and adopt ResNet-101 [He *et al.*, 2016] as the backbone network for *DomainNet*. We employ a single fully connected layer as the classifier, *i.e.,* domain specific classifier, for each source domain. We use the Adam optimizer with the learning rate 10^{-5} and weight decay 5×10^{-4}. For S^{3}DA-LC, we set τ to 0.9 for all datasets and the sensitivity analysis of parameters will be discussed later. We follow the common setting for Tukey's fences [Tukey, 1977] and set λ to 1.5 for all datasets. The implementation is available at https://github.com/MengKang98/S3DA-LC.

4.1 Comparing against Existing Baselines

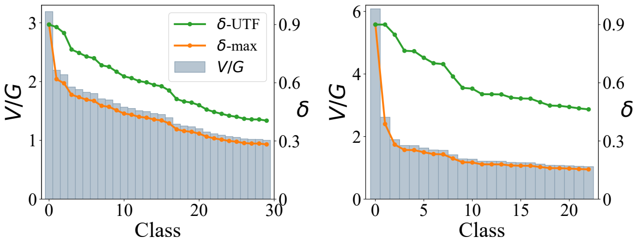
We verify the effectiveness of S^{3}DA-LC by comparing it with existing MSDA methods. The results are shown in Tables 1 and 2, where we take the averages of 3 independent runs as the final results.

For fair comparisons, we adopt 3 evaluation standards defined in [Venkat *et al.*, 2020]: (1) *Single Best*: the best performance of SSDA methods among all source domains; (2) *Source Combine*: all source domains are combined into a single source domain to perform SSDA; (3) *Multi-source*: adaptation from all source domains to the target domain. For the first two settings, we compare S^{3}DA-LC with previous SSDA methods, *e.g.,* DAN [Long *et al.*, 2015], D-CORAL [Sun and Saenko, 2016], MCD [Saito *et al.*, 2018] and CAN [Kang *et al.*, 2019]. For the *Multi-source* setting, we select 9 existing MSDA methods for comparison, including DCTN [Xu *et al.*, 2018], SImpAl [Venkat *et al.*, 2020], KD3A [Feng *et al.*, 2021], MSCAN [Kang *et al.*, 2022], CAiDA [Dong *et al.*, 2021], STEM [Nguyen *et al.*, 2021], DECISION [Ahmed *et al.*, 2021] and SPS [Wang *et al.*, 2022]. The results of baselines under *Single Combine* setting are cited from STEM [Nguyen *et al.*, 2021] and SPS [Wang *et al.*, 2022] and the results under *Single Best* and *Multi-Source* settings are cited from corresponding papers.

As shown in Tables 1 and 2, S^{3}DA-LC achieves competitive results on all datasets averagely. On *Office-31*, S^{3}DA-LC exceeds former SOTA methods CAiDA [Dong *et al.*, 2021] on all tasks. On *Office-Home*, S^{3}DA-LC exceeds former SOTA methods SPS [Wang *et al.*, 2022] on all tasks, with huge improvements of 4.0% on →*Cl*, 3.0% on →*Ar*, →*Pr* and →*Rw*. On *DomainNet*, S^{3}DA-LC exceeds former SOTA

Models	*Office-Home*		*DomainNet*	
	→*Ar*	→*Cl*	→*Inf*	→*Skt*
Warm-up Only	72.3	56.1	26.2	56.2
w/o δ_c	73.3	57.5	27.6	59.1
w/o w^k	77.6	69.9	30.8	60.8
S^{3}DA-LC	**78.1**	**70.0**	**31.3**	**61.2**
with $\Phi = \max(\cdot)$	77.7	65.1	28.8	60.8
with $\lambda = 3.0$	77.8	69.2	31.1	**61.2**

Table 3: Experimental results of ablation study.



(a) →*Ar*

(b) →*Cl*

Figure 2: Negative effects of outliers (warm-up only).

methods STEM [Nguyen *et al.*, 2021] on 4 out of 6 transfer tasks, with huge improvements of 3.1% on →*Inf* and →*Rel*.

4.2 Ablation Study

To evaluate contribution of each part of S^{3}DA-LC, we decompose S^{3}DA-LC to reveal their functions. Table 3 shows the results of ablation study.

We first set $\tau = 0.0$ to verify the label-specific confidences, *i.e.,* **w/o** δ_c. The performance degrades huge on all tasks. This is because all target domain samples are assigned pseudo-labels and participate in self-supervised training when τ equals 0.0. And the pseudo-labels selected in this setting have a very low accuracy rate and are very unbalanced.

We also evaluate the function of transferability weights by setting $w^k = 1$ for $k = 1, ..., K$, *i.e.,* **w/o** w^k. In this case, we treat all source domains equally. S^{3}DA-LC outperforms **w/o** w^k on all tasks. This suggests the effectiveness of transferability weights $\{w^k\}_{k=1}^K$.

4.3 Negative Effects of Outliers

In Eq.(8), we use the upper range of Tukey's fences (UTF), *i.e.,* $\mathbf{\Phi} = Q_3 + \lambda \cdot (Q_3 - Q_1)$, to avoid the negative effects of potential outliers in $\mathcal{V}/\mathcal{G}$. To evaluate the function of UTF, we set $\Phi = \max(\mathcal{V}/\mathcal{G})$, *i.e.,* **with** $\Phi = \max(\cdot)$ and test it on several tasks. As shown in Table 3, performance dropped on all tasks. Notably, performance degrades more on tasks →*Cl* (-4.9%) and →*Inf* (-2.5%) than on →*Ar* (-0.4%) and →*Skt* (-0.4%). This is because the outliers are relatively larger on the former tasks.

As shown in Fig.2, we illustrate the difference between $\mathcal{G}$ and $\mathcal{V}$ labels with $\mathcal{V}/\mathcal{G} > 1.0$. These labels are commonly easy-to-learn labels. The green and orange lines represent the label-specific confidences δ normalized by UTF (δ-UTF) and $\max(\cdot)$ (δ-max) respectively. Comparing Fig.2 (a) and (b), the largest and second largest values of $\mathcal{V}/\mathcal{G}$ are significantly more different on →*Cl* than on →*Ar*. Therefore, for the rest labels with $\mathcal{V}/\mathcal{G} > 1.0$, δ-max is much lower than δ-UTF on →*Cl*, which will weaken the constraining ability on easy-to-learn labels. While on →*Ar*, the gap of δ between δ-max and δ-UTF is smaller. This explains why the performance of **with** $\Phi = \max(\cdot)$ degrades less on →*Ar*. Above all, the experiments show negative effects of outliers and also verify the effectiveness of UTF.

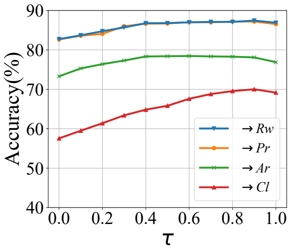


Figure 3: Sensitivity of τ.

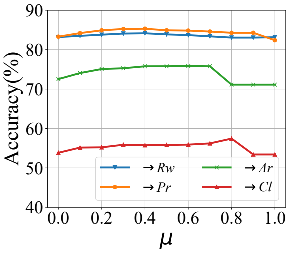


Figure 4: Sensitivity of μ.

4.4 Sensitivity of Parameters

To study the sensitivity of τ, we investigate 11 different τ values (from 0.0 to 1.0) on *Office-Home*. As shown in Fig.3, the performance decay on all four tasks occurs when τ approaches 0.0. This suggests that a too small τ will introduces too much noise into the pseudo-labels, which will deteriorate the adaptation. And with increasing of τ, the performance on tasks →*Rw*, →*Pr* and →*Ar* remains stable when $\tau \in [0.4, 0.9]$. The performance on task →*Cl* achieves best results at $\tau = 0.9$. In summary, we set τ as 0.9 for practice.

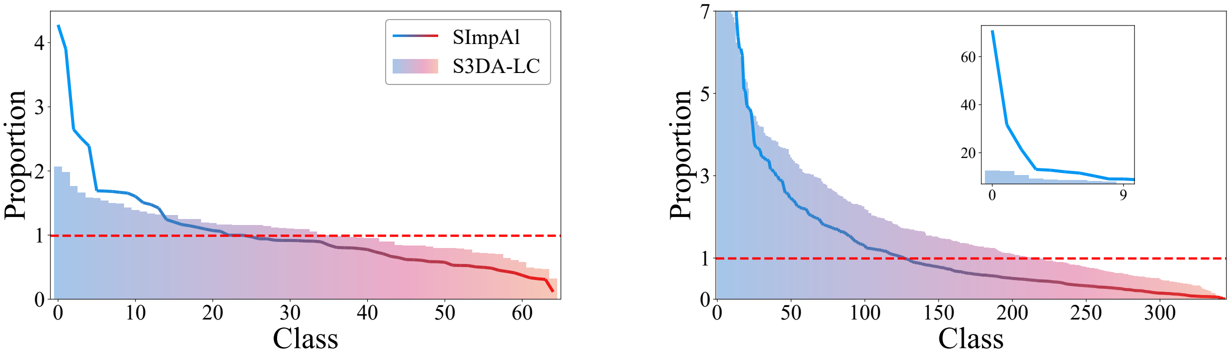
Notably, even performances on multiple tasks decrease when $\tau = 1.0$, S^{3}DA-LC still achieves good results even when τ =1.0. This is because Eq.(12) will gradually update the labels with initial δ_c =1.0 to less than 1.0. Thus the labels abandoned at the beginning will rejoin the training eventually.

S^{3}DA-LC use the "regular" setting in [Tukey, 1977] where $\lambda = 1.5$. We also test the "far" setting where $\lambda = 3.0$, *i.e.,* **with** $\lambda = 3.0$. As shown in Table 3, the performances of **with** $\lambda = 3.0$ are closer to the S^{3}DA-LC comparing with **with** $\Phi = \max(\cdot)$. And S^{3}DA-LC still achieves best performances.

4.5 Fix Threshold Strategy

We adopt idea of fix threshold strategy where we replace the label-specific confidences δ_c with a fixed confidence μ in Eq.(9). We evaluate different μ values (from 0.0 to 1.0) on *Office-Home*.

Comparing Fig.3 and Fig.4, it is evident that label-specific confidences strategy vastly outperforms fix threshold strategy in almost every setting. Notably, there are huge drops on →*Cl* when μ >0.8 and →*Ar* when μ >0.7. This is because the confidence level of the model on the target domain cannot exceed μ when μ is too large, so almost no sample is assigned pseudo-labels, which results in training failure. And fix threshold strategy is also sensitive to μ since the optimal values on different tasks are also different. Notably, fix threshold strategy degrades to warm-up when τ =1.0. This



(a) →*Cl* (*Office-Home*)

(b) →*Inf* (*DomainNet*)

Figure 5: The pseudo-label distributions.

Task	Method	Hard-to-learn		Easy-to-learn		All	
		Num$_{\downarrow}$	Avg$_{\uparrow}$	Num$_{\downarrow}$	Avg$_{\downarrow}$	Var$_{\downarrow}$	Acc$_{\uparrow}$
→*Cl*	SImpAl	43	0.68	**22**	1.78	0.55	56.85
	S^{3}DA-LC	**31**	**0.79**	34	**1.32**	**0.12**	**73.84**
→*Inf*	SImpAl	218	0.41	**127**	3.88	21.67	27.14
	S^{3}DA-LC	**128**	**0.58**	217	**2.72**	**3.59**	**34.09**

Table 4: Statistical comparison. ↑ indicates the higher the value, the better result, and vice versa.

is because no sample will be assigned pseudo-label and the adaptation stage cannot proceed. The results validate the superiority of label-specific confidences.

4.6 Distribution of Pseudo-labels

To verify the effectiveness of S^{3}DA-LC in alleviating the imbalanced pseudo-label distribution, we compare the pseudo-label distributions generated by S^{3}DA-LC and SImpAl [Venkat *et al.*, 2020]. Fig.5 shows the visualized distributions. The deep gradient blue-red line represents the distribution generated by SImpAl which is the simplified version of Fig.1, and the light gradient blue-red bars represent the distribution generated by S^{3}DA-LC. For a fair comparison, we visualize the distributions of the final update of pseudo-labels for both methods.

We expect the pseudo-label distribution to be close to the true label distribution, *i.e.,* the proportions between pseudo-label distribution and true label distribution on the target domain (the vertical axis in Fig.5) should be close to 1.0. As shown in Fig.5, the proportions of S^{3}DA-LC are flatter than SImpAl, and the proportions of leftmost labels are also milder. Apparently, the pseudo-label distributions generated by S^{3}DA-LC are closer to the true label distribution.

We further analyze the pseudo-label distribution generated by S^{3}DA-LC and SImpAl statistically, as shown in Table 4. For hard-to-learn labels, S^{3}DA-LC largely reduces the number (**Num**) of them and increases the average (**Avg**) of proportions. This verifies the enriching ability of S^{3}DA-LC on hard-to-learn labels. S^{3}DA-LC greatly enriches the sample number of hard-to-learn labels, which increases the number of easy-to-learn labels. However, even with more easy-to-learn labels, S^{3}DA-LC still achieves lower average of proportions. This verifies the constraining ability of S^{3}DA-LC on easy-to-learn labels.

For all pseudo-labels, we use the variance of the proportions (**Var**) to measure the difference between the pseudo-label distribution and true label distribution. The smaller the variance, the closer the distribution of pseudo-labels and true labels. If the variance of the proportions equals to 0, the pseudo-labels distribution and true label distribution will be identical. As shown in Table 4, S^{3}DA-LC achieves much lower **Var** than SImpAl on both tasks, which represents S^{3}DA-LC generates more balanced pseudo-labels. Meanwhile, S^{3}DA-LC also generates more accurate pseudo-labels. All together verifies that S^{3}DA-LC can alleviate the imbalanced pseudo-label distribution.

5 Conclusion

In this paper, we focus on the imbalanced characteristic of pseudo-label distribution in target domain, where both easy-to-learn and hard-to-learn labels extremely hurt the generalization performance. We propose a self-supervised MSDA method named S^{3}DA-LC to alleviate such problem. S^{3}DA-LC estimates label-specific confidences by measuring the differences between the prior distribution of all domains and the distribution of ensemble prediction over the target domain, then uses them as dynamic thresholds to generate more precise pseudo-labels for target domain. Along with self-supervised training, S^{3}DA-LC achieves state-of-the-art performance comparing against the existing MSDA baselines.

Acknowledgments

We sincerely thank the anonymous reviewers for their careful work and thoughtful suggestions, which have greatly improved this article. This work was supported by the Natural Science Research Foundation of Jilin Province of China under Grant Nos. 20220101106JC and YDZJ202201ZYTS423, the National Natural Science Foundation of China under Grant No. 62276113, the China Postdoctoral Science Foundation under Grant No. 2022M721321, and the Fundamental Research Funds for the Central Universities of China under Grant Nos. 2412022ZD018, 2412022QD040 and 93K172022K10.

References

[Ahmed *et al.*, 2021] Sk Miraj Ahmed, Dripta S. Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K. Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, pages 10103–10112, 2021.

[Berthelot *et al.*, 2020] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the International Conference on Learning Representations (ICLR 2020)*, 2020.

[Dong *et al.*, 2021] Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, pages 2848–2860, 2021.

[Feng *et al.*, 2021] Haozhe Feng, Zhaoyang You, Minghao Chen, Tianye Zhang, Minfeng Zhu, Fei Wu, Chao Wu, and Wei Chen. KD3A: Unsupervised multi-source decentralized domain adaptation via knowledge distillation. In *Proceedings of the International Conference on Machine Learning (ICML 2021)*, pages 3274–3283, 2021.

[Ganin and Lempitsky, 2015] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning (ICML 2015)*, pages 1180–1189, 2015.

[Guo *et al.*, 2020] Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Multi-source domain adaptation for text classification via distancenet-bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 7830–7838, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 770–778, 2016.

[Kang *et al.*, 2019] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 4893–4902, 2019.

[Kang *et al.*, 2022] Guoliang Kang, Lu Jiang, Yunchao Wei, Yi Yang, and Alexander Hauptmann. Contrastive adaptation network for single-and multi-source domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1793–1804, 2022.

[Kundu *et al.*, 2020] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, Rahul M. V., and R. Venkatesh Babu. Towards inheritable models for open-set domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, pages 12373–12382, 2020.

[Li and Wang, 2020] Ximing Li and Yang Wang. Recovering accurate labeling information from partially valid data for effective multi-label learning. In *Proceedings of the International Joint Conference on Artificial Intelligence, (IJCAI 2020)*, pages 1373–1380, 2020.

[Li *et al.*, 2020] Changchun Li, Ximing Li, and Jihong Ouyang. Learning with noisy partial labels by simultaneously leveraging global and local consistencies. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM 2020)*, pages 725–734, 2020.

[Liang *et al.*, 2020] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML 2020)*, pages 6028–6039, 2020.

[Long *et al.*, 2015] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the International Conference on Machine Learning (ICML 2015)*, pages 97–105, 2015.

[Long *et al.*, 2017] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the International Conference on Machine Learning (ICML 2017)*, pages 2208–2217, 2017.

[Nguyen *et al.*, 2021] Van-Anh Nguyen, Tuan Nguyen, Trung Le, Quan Hung Tran, and Dinh Phung. STEM: An approach to multi-source domain adaptation with guarantees. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2021)*, pages 9332–9343, 2021.

[Peng *et al.*, 2019] 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 International Conference on Computer Vision (ICCV 2019)*, pages 1406–1415, 2019.

[Saenko *et al.*, 2010] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV 2010)*, volume 6314, pages 213–226, 2010.

[Saito *et al.*, 2018] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), pages 3723–3732, 2018.

[Saito *et al.*, 2020] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Sun and Saenko, 2016] Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, pages 443–450, 2016.

[Tan *et al.*, 2018] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning (ICANN 2018)*, volume 11141, pages 270–279, 2018.

[Tukey, 1977] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[Venkat *et al.*, 2020] Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and Venkatesh Babu R. Your classifier can secretly suffice multi-source domain adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Venkateswara *et al.*, 2017] 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 (CVPR 2017)*, pages 5385–5394, 2017.

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

[Wang *et al.*, 2022] Zengmao Wang, Chaoyang Zhou, Bo Du, and Fengxiang He. Self-paced supervision for multi-source domain adaptation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2022)*, pages 3551–3557, 2022.

[Wen *et al.*, 2020] Junfeng Wen, Russell Greiner, and Dale Schuurmans. Domain aggregation networks for multi-source domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML 2020)*, pages 10214–10224, 2020.

[Xie *et al.*, 2020] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Xu *et al.*, 2018] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep Cocktail Network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, pages 3964–3973, 2018.

[Zhang *et al.*, 2021] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

[Zhao *et al.*, 2018] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, João Paulo Costeira, and Geoffrey J. Gordon. Adversarial multiple source domain adaptation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2018)*, pages 8568–8579, 2018.

[Zhao *et al.*, 2020a] Sicheng Zhao, Bo Li, Colorado Reed, Pengfei Xu, and Kurt Keutzer. Multi-source domain adaptation in the deep learning era: A systematic survey. *arXiv*, abs/2002.12169, 2020.

[Zhao *et al.*, 2020b] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 12975–12983, 2020.

[Zhao *et al.*, 2020c] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):473–493, 2020.