Semi-Supervised Domain Adaptation with Source Label Adaptation

Yu-Chu Yu Hsuan-Tien Lin
National Taiwan University
{r09922104, htlin}@csie.ntu.edu.tw

Abstract

Semi-Supervised Domain Adaptation (SSDA) involves learning to classify unseen target data with a few labeled and lots of unlabeled target data, along with many labeled source data from a related domain. Current SSDA approaches usually aim at aligning the target data to the labeled source data with feature space mapping and pseudo-label assignments. Nevertheless, such a source-oriented model can sometimes align the target data to source data of the wrong classes, degrading the classification performance. This paper presents a novel source-adaptive paradigm that adapts the source data to match the target data. Our key idea is to view the source data as a noisily-labeled version of the ideal target data. Then, we propose an SSDA model that cleans up the label noise dynamically with the help of a robust cleaner component designed from the target perspective. Since the paradigm is very different from the core ideas behind existing SSDA approaches, our proposed model can be easily coupled with them to improve their performance. Empirical results on two state-of-the-art SSDA approaches demonstrate that the proposed model effectively cleans up the noise within the source labels and exhibits superior performance over those approaches across benchmark datasets. Our code is available at https://github.com/chu0802/SLA.

1. Introduction

Domain Adaptation (DA) focuses on a general machine learning scenario where training and test data may originate from two related but distinct domains: the source domain and the target domain. Many works have extensively studied unsupervised DA (UDA), where labels in the target domain cannot be accessed, from both theoretical [2, 19, 36] and algorithmic [5, 8, 15, 16, 22, 37] perspectives. Recently, Semi-Supervised Domain Adaptation (SSDA), another DA setting that allows access to a few target labels, has received more research attention because it is a simple yet realistic setting for application needs.

The most naïve strategy for SSDA, commonly known as S+T [21, 33], aims to train a model using the source data and labeled target data with a standard cross entropy loss. This strategy often suffers from a well-known *domain shift* issue, which stems from the gap between different data distributions. To address this issue, many state-of-the-art algorithms attempt to explore better use of the unlabeled target data so that the target distribution can be aligned with the source distribution. Recently, several Semi-Supervised Learning (SSL) algorithms have been applied for SSDA [12, 21, 30] to regularize the unlabeled data, such as entropy minimization [6], pseudo-labeling [11, 24] and consistency regularization [1, 24]. These classic source-oriented strategies have prevailed for a long time. However, these algorithms typically require the target data to closely match some semantically similar source data in the feature space. Therefore, if the S+T space has been misaligned, it can be challenging to recover from the misalignment, as illustrated in Figure 1.

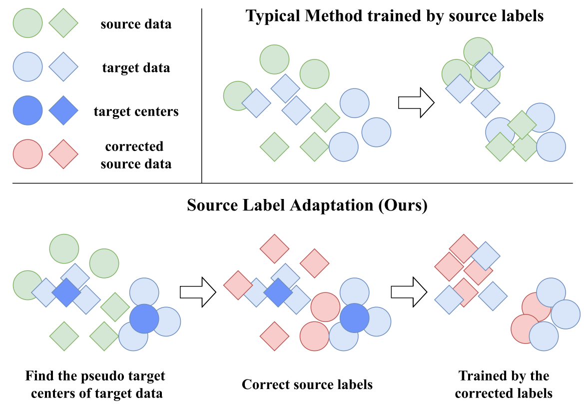

Figure 1. **Top.** Training the model with the original source labels might lead to the misalignment of the target data. **Bottom.** After cleaning up the noisy source labels with our SLA framework, the target data can be aligned with the correct classes.

We take a deeper look into a specific example from the *Office-Home* dataset [27] to confirm the abovementioned issue. Figure 2 visualizes the feature space trained by S+T using t-SNE [3]. We observed that the misalignment between

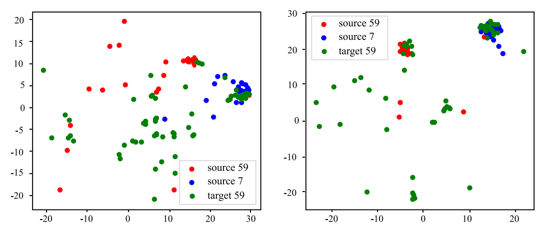

Figure 2. Feature visualizations with t-SNE for an example of the misalignment on the *Office-Home* A $\rightarrow$ C dataset with ResNet34. The model is trained by S+T. **Left**: 0-th iteration. **Right**: 5000-th iteration . We observe that the misalignment has already happened at a very early stage. Guided by source labels and a few target labels, a portion of the target data from the 59th class misaligns with the source data from the 7th class.

True\Pred	Class 7	Class 59	Class 41	Others
Class 59	38.5%	19.8%	13.5%	28.2%

Table 1. A partial confusion matrix of S+T on the 3-shot *Office-Home* A $\rightarrow$ C dataset with ResNet34.

the source and the target data has happened at a very early stage. For instance, in the beginning, a portion of the target data from the 59th class is close to the source data from the 7th class. Since we only have access to source labels and a few target labels, without proper guidance from enough target labels, such misalignment becomes more severe after being trained by S+T. Table 1 shows the partial confusion matrix of S+T. Roughly 40% of the target data in the 59th class is mispredicted to the 7th class, and only around 20% of the data is classified correctly.

From the case study above, we argue that relying on the source labels like S+T can misguide the model to learn wrong classes for some target data. That is, source labels can be viewed as a *noisy* version of the ideal labels for target classification. Based on the argument, the setting of SSDA is more like a Noisy Label Learning (NLL) problem, with a massive amount of noisy labels (source labels) and a small number of clean labels (target labels).

Learning with noisy labels is a widely studied machine learning problem. A popular solution is to clean up the noisy labels with the help of another model, which can also be referred to as label correction [28]. To approach Domain Adaptation as an NLL problem, we borrow the idea from label correction and propose a Source Label Adaptation (SLA) framework, as shown in Figure 1. We construct a label adaptation component that provides the view from the target data and dynamically cleans up the noisy source labels at each iteration. Unlike other earlier works that study how to leverage the unlabeled target data, we mainly investigate how to train the source data with the adapted labels to better suit the ideal target space. This source-adaptive paradigm is entirely orthogonal to the core ideas behind existing SSDA algorithms. Thus, we can combine our framework with those algorithms to get superior results. We summarize our contributions as follows.

- We argue that the classic source-oriented methods might still suffer from the biased feature space derived from S+T. To escape this predicament, we propose adapting the source data to the target space by modifying the original source labels.
- We address DA as a particular case of NLL problems and present a novel source-adaptive paradigm. Our SLA framework can be easily coupled with other existing algorithms to boost their performance.
- We demonstrate the usefulness of our proposed SLA framework when coupled with state-of-the-art SSDA algorithms. The framework significantly improved existing algorithms on two major benchmarks, inspiring a new direction for solving DA problems.

2. Related Work

Problem Setup. DA focuses on a K-class classification task with an m-dimensional input space $X \subseteq \mathbb{R}^m$ and a set of labels $\{1, 2, \ldots, K\}$. For simplicity, we define a label space Y on the probability simplex Δ^K. A label $y = k \in \{1, 2, \ldots, K\}$ is equivalent to a one-hot encoded vector $\mathbf{y} \in Y$, where only the k-th element is 1 and the others are 0. We consider two domains over $X \times Y$, named source domain D_s and target domain D_t. In SSDA, we sample an amount of labeled source data $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{|S|}$ from D_s, labeled target data $L = \{(\mathbf{x}_i^\ell, y_i^\ell)\}_{i=1}^{|L|}$ from D_t, and unlabeled target data $U = \{\mathbf{x}_i^u\}_{i=1}^{|U|}$ from the marginal distribution of D_t over X. Typically, $|L|$ is considerably smaller than $|S|$ and $|U|$, such as one or three examples per class. Our goal is to train an SSDA model g with S, L, and U to perform well on the target domain.

Semi-Supervised Domain Adaptation (SSDA). SSDA can be viewed as a relaxed yet realistic version of UDA. An SSDA algoirthm usually involves three loss functions:

$$\mathcal{L}_{\text{SSDA}} = \mathcal{L}_s + \mathcal{L}_\ell + \mathcal{L}_u \tag{1}$$

where $\mathcal{L}_s$ stands for the loss derived by the source data. $\mathcal{L}_\ell, \mathcal{L}_u$ denotes the losses from the labeled and unlabeled target data. As discussed in Section 1, based on S+T, a typical SSDA algorithm usually focuses on designing $\mathcal{L}_u$ to better align the target data with the source data. Recently, many existing works have borrowed SSL techniques to conquer SSDA because of the problem similarity [35]. [21]

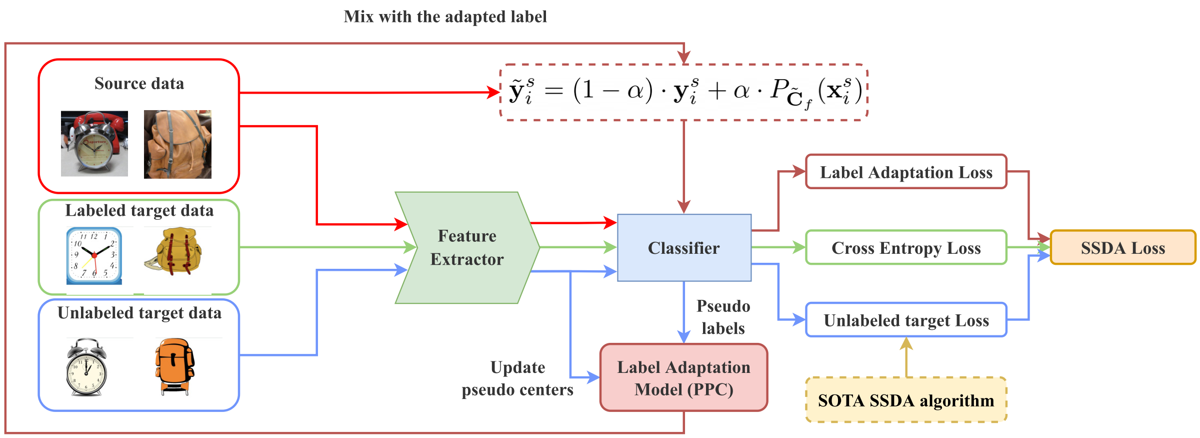

Figure 3. An overview of our proposed framework, Source Label Adaptation for SSDA. For source data, we adapt the original source labels to better fit the target feature space by the Protonet with Pseudo Centers (PPC) and calculate the label adaptation loss. For labeled target data, we train it with a standard cross entropy loss. We can apply a state-of-the-art algorithm to derive the unlabeled target loss for unlabeled data. For every specific interval I, we update the pseudo labels and pseudo centers to get a more reliable label adaptation model.

proposes a variant of entropy minimization [6] to explicitly align the target data with source clusters. [31] decomposes SSDA into an SSL and a UDA task. The two different sub-tasks produce pseudo labels respectively, and learn from each other via co-training. [12] groups target features into clusters by measuring pairwise feature similarity. [30] utilizes consistency regularization at three different levels to perform domain alignment. Besides, both [12, 30] apply pseudo labeling with data augmentations [24] to enhance their performance. To the best of our knowledge, all methods listed above mainly explore the usage of unlabeled target data while treating the source data with the most straightforward strategy. In our study, we noticed that source labels could appear noisy from the viewpoint of the target data. We thus developed a source-adaptive framework to gradually adapt the source data to the target space. Since we are addressing a new facet of the issue, our framework can be easily applied to several SSDA algorithms mentioned above, further improving the overall performance.

Noisy Label Learning (NLL). The effectiveness of a machine learning algorithm highly depends on the quality of collected labels. With regard to the present deep neural network design [7], the aforementioned issue could worsen as deep models have the capability to fit the data set in a seemingly random manner, regardless of the quality of the labels [34].To clean the noisy labels, [20] proposes a smoothing mechanism to mix noisy labels with self-prediction. [26] models clean labels as trainable parameters and designs a joint optimization algorithm to alternatively update parameters. [17, 25, 32] estimate a transition matrix to correct the corrupted labels. However, learning a global transition matrix usually need a strong assumption of how noisy labels come from, which is difficult to verify in the real-world scenarios [29]. [38] trains a label correction network in a meta-learning manner to help correct noisy labels. Motivated by [20, 38], we propose a simple framework that can efficiently build a label adaptation model to correct the noisy source labels.

3. Proposed Framework

Next, we propose a novel SSDA framework, *Source Label Adaptation*. An overview of our proposed framework is shown in Figure 3. In Section 3.1, we connect the (SS)DA problem to NLL and point out that a classic NLL method [20] cannot be directly applied to solve SSDA. In Section 3.2, we review a classic few-shot learning algorithm, *Prototypical Network* [23] and propose *Protonet with Pseudo Centers* to better estimate the prototypes. In Section 3.3, we summarize our framework and describe the implementation details.

3.1. Domain Adaptation as Noisy Label Learning

In Domain Adaptation, we seek an ideal model g^* that can minimize unlabeled target risk. Ideally, the most suitable label for a source instance $\mathbf{x}_i^s$ in the target space should be $g^*(\mathbf{x}_i^s)$. That is, the ideal source loss $\mathcal{L}_s^*$ is:

$$\mathcal{L}_s^*(g|S) = \frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), g^*(\mathbf{x}_i^s)), \tag{2}$$

where H measures the cross entropy between two distributions.

Combining with the labeled target loss $\mathcal{L}_\ell$, we refer to the model trained by $\mathcal{L}_s^*$ and $\mathcal{L}_\ell$ as *ideally-adapted S+T*.

Method	A → C 1-shot	A → C 3-shot	P → C 1-shot	P → C 3-shot
S+T	52.9	58.1	48.8	55.5
ideally-adapted S+T	82.9	87.4	81.6	86.0

Table 2. Accuracy (%) of S+T and ideally-adapted S+T on the 3-shot OfficeHome dataset with ResNet34.

The results of the ideally-adapted S+T reveal the full potential to adapt source labels. As shown in Table 2, there is a significant difference in performance between a standard S+T and an ideally-adapted S+T, demonstrating that performance can be dramatically affected by only modifying the source labels.

In practice, however, we can only approximate the ideal model. To address the issue, we take the original source labels as a noisy version of the ideal labels and approach DA as a NLL problem. We first apply a simple method proposed by [20] to help correct the source labels, which we refer to it as *label correction with self-prediction* [28]. Specifically, for each source instance $\mathbf{x}_i^s$, we construct the modified source label $\hat{\mathbf{y}}_i^s$ by combining the original label $\mathbf{y}_i^s$ and the prediction from the current model g with a ratio α.

$$\hat{\mathbf{y}}_i^s = (1-\alpha)\cdot\mathbf{y}_i^s + \alpha\cdot g(\mathbf{x}_i^s) \tag{3}$$

Then, the modified source loss $\hat{\mathcal{L}}_s$ is:

$$\hat{\mathcal{L}}_s(g|S) = \frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \hat{\mathbf{y}}_i^s) \tag{4}$$

However, in DA, such a method might not be helpful since the model usually overfits the source data, which makes $g(\mathbf{x}_i^s) \approx \mathbf{y}_i^s$. That is, the modified source label $\hat{\mathbf{y}}_i^s$ can be almost the same as the original source label $\mathbf{y}_i^s$ according to Eq. 3.

Figure 4 shows that when doing *label correction with self-prediction*, the KL divergence from $\mathbf{y}^s$ to $g(\mathbf{x}^s)$ could be close to 0 after 2000 iterations, indicating that the self-prediction is almost the same as the original label. In this case, doing the correction is nearly equivalent to not doing so.

To benefit from the modified labels, we need to eliminate supervision from source data. As an ideal clean label is the output from an ideal model g^*, we should instead find a *label adaptation model* g_c that can approximate the ideal model and adapt the source labels to the view of target data. We define an adapted labels $\tilde{\mathbf{y}}_i^s$ as a convex combination between the original labels $\mathbf{y}_i^s$ and the output from g_c, which is the same as [20].

$$\tilde{\mathbf{y}}_i^s = (1-\alpha)\cdot\mathbf{y}_i^s + \alpha\cdot g_c(\mathbf{x}_i^s) \tag{5}$$

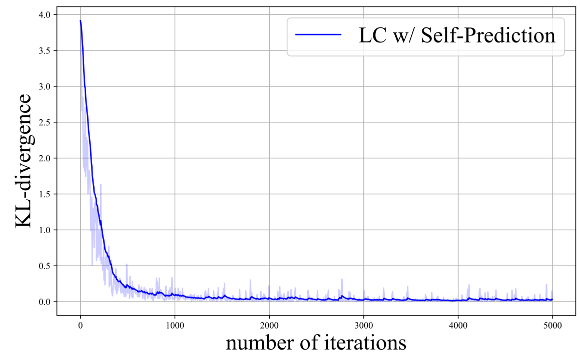

Figure 4. Average KL divergence from $\mathbf{y}^s$ to $g(\mathbf{x}^s)$ at each iteration (3-shot *Office-Home* A → C with ResNet34, smoothing by EMA with a ratio 0.8).

3.2. Protonet with Pseudo Centers

In the semi-supervised setting, we can access a few target labels. Nonetheless, learning from a limited number of target labels might suffer from a severe overfitting issue. Thus, we learn a prototypical network (protonet) [23] to overcome the few-shot problem.

Given a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^N$ and a feature extractor f; let N_k denote the number of data labeled with k. The prototype of class k is defined as the center of features with the same class:

$$\mathbf{c}_k = \frac{1}{N_k}\sum_{i=1}^{N} \mathbb{1}\{y_i = k\}\cdot f(\mathbf{x}_i). \tag{6}$$

Let $\mathbf{C}_f = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ collects all centers with extractor f. We define $P_{\mathbf{C}_f} : X \mapsto Y$ as a protonet with centers $\mathbf{C}_f$:

$$P_{\mathbf{C}_f}(\mathbf{x}_i)_k = \frac{\exp(-d(f(\mathbf{x}_i), \mathbf{c}_k)\cdot T)}{\sum_{j=1}^{K}\exp(-d(f(\mathbf{x}_i), \mathbf{c}_j)\cdot T)} \tag{7}$$

Here $d : F \times F \mapsto [0, \infty)$ is a distance measure over feature space F, usually measuring Euclidean distance. T is a hyper-parameter that controls the smoothness of output. As $T \to 0$, the output of a protonet would be close to a uniform distribution.

Since we have access to the labeled target dataset L, by Eq. 6 and Eq. 7, we can derive labeled target centers $\mathbf{C}_f^\ell$, and construct a protonet with labeled target centers $P_{\mathbf{C}_f^\ell}$.

[23] demonstrated that when d measures Euclidean distance, a protonet is equivalent to a linear classifier with particular parameterization over F. Thus, we can take the protonet as a label adaptation model over a particular feature space. The protonet with labeled target centers is purely built from the viewpoint of target data, which should reduce our concerns about the issue mentioned in Section 3.1.

From / To	labeled target centers	pseudo centers
ideal centers	10.02	4.06

Table 3. Average L2 Distance from ideal centers to labeled target centers / pseudo centers over the feature space trained by S+T (3-shot *Office-Home* A $\rightarrow$ C with ResNet34).

However, for a protonet, the ideal centers $\mathbf{C}_f^*$ should be derived through the unlabeled target dataset $\{(\mathbf{x}_i^u, y_i^u)\}_{i=1}^{|U|}$. Since we have only a few target labels per class, the labeled target centers $\mathbf{C}_f^\ell$ are far away from the ideal centers $\mathbf{C}_f^*$. To better estimate the ideal centers, we propose to find the pseudo centers for unlabeled target data.

With the current model g, the pseudo label $\tilde{y}_i^u$ for an unlabeled target instance $\mathbf{x}_i^u$ is:

$$\tilde{y}_i^u = \arg\max_k g(\mathbf{x}_i^u)_k \tag{8}$$

After deriving unlabeled target data with pseudo labels $\{(\mathbf{x}_i^u, \tilde{y}_i^u)\}_{i=1}^{|U|}$, we can get pseudo centers $\tilde{\mathbf{C}}_f$ by Eq. 6, and further define a Protonet with Pseudo Centers (PPC) $P_{\tilde{\mathbf{C}}_f}$ by Eq. 7.

Table 3 compares the average L2 distance from ideal centers $\mathbf{C}_f^*$ to labeled target centers $\mathbf{C}_f^\ell$ and pseudo centers $\tilde{\mathbf{C}}_f$ over the feature space trained by S+T. The distance between $\tilde{\mathbf{C}}_f$ and $\mathbf{C}_f^*$ is significantly shorter than the distance between $\mathbf{C}_f^\ell$ and $\mathbf{C}_f^*$, which means the pseudo centers are indeed much closer to the ideal centers.

Taking PPC as the label adaptation model, the modified label $\tilde{\mathbf{y}}_i^s$ turns out to be:

$$\tilde{\mathbf{y}}_i^s = (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot P_{\tilde{\mathbf{C}}_f}(\mathbf{x}_i^s) \tag{9}$$

3.3. Source Label Adaptation for SSDA

We propose a label adaptation loss for source data to replace the typical source loss with a standard cross entropy loss. For each source instance $\mathbf{x}_i^s$ with label $\mathbf{y}_i^s$, we first compute the modified source label $\tilde{\mathbf{y}}_i^s$ by Eq. 9. Then, the label adaptation loss $\tilde{\mathcal{L}}_s$ is:

$$\tilde{\mathcal{L}}_s(g|S) = \frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \tilde{\mathbf{y}}_i^s) \tag{10}$$

Our framework, Source Label Adaptation (SLA) for SSDA, can be trained by the following loss function.

$$\mathcal{L}_{\text{SSDA w/ SLA}} = \tilde{\mathcal{L}}_s + \mathcal{L}_\ell + \mathcal{L}_u \tag{11}$$

$\mathcal{L}_\ell$ is the loss function for labeled target data L, which can still be a standard cross entropy loss. In contrast to other widely used methods, we primarily concentrate on improving the usage of source data. Therefore, the loss function for unlabeled target data $\mathcal{L}_u$ can be derived through any state-of-the-art algorithm, and our framework can be easily coupled with other methods without contradiction.

3.3.1 Implementation Details

Warmup Stage. Our label adaptation framework relies on the quality of the predicted pseudo labels. However, the prediction from the initial model can be noisy. Thus, we introduce a hyperparameter W for warmup to get more stable pseudo labels. During the warmup stage, we train our model normally with original source labels. Specifically, at the e-th iteration, we compute the modified source label $\tilde{\mathbf{y}}_i^s$ as follows:

$$\tilde{\mathbf{y}}_i^s = \begin{cases} \mathbf{y}_i^s & \text{if } e \le W \\ (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot P_{\tilde{\mathbf{C}}_f}(\mathbf{x}_i^s) & \text{otherwise} \end{cases} \tag{12}$$

Dynamic Update. The feature space and the predicted pseudo labels constantly evolve during the training phase. By updating the pseudo labels and centers, we can remain the quality of the projected pseudo centers the same. It would be ideal for updating the centers at each iteration. In practice, we update the pseudo labels through Eq. 8 and update centers with the current feature extractor f through Eq. 6 for every specific interval I. Prior works [14] have addressed a similar issue and proposed to maintain a memory bank for dynamic updates of the estimated centers. However, in our framework, we need to update both the estimated centers and pseudo labels simultaneously. Therefore, we decided to adopt a more straightforward solution to mitigate the demands on time and complexity.

4. Experiments

We first sketch our experiment setup, including data sets, competing methods, and parameter settings in Section 4.1. We then present experimental results to validate the superiority of the proposed SLA framework in Section 4.2. We further analyze our proposed framework and highlight the limitation in Section 4.3.

4.1. Experiment Setup

Datasets. We evaluate our proposed SLA framework on two sets of SSDA benchmarks, including *Office-Home* [27] and *DomainNet* [18]. *Office-Home* is a mainstream benchmark for both UDA and SSDA. It contains four domains: Art (A), Clipart (C), Product (P), and Real (R), with 65 categories. *DomainNet* is initially designed for benchmarking Multi-Source Domain Adaptation approaches. [21] pickup four domains: Real (R), Clipart (C), Painting (P), and Sketch (S) with 126 classes to build a cleaner dataset for SSDA. Besides, they focus on seven scenarios instead

Method	R → C		R → P		P → C		C → S		S → P		R → S		P → R		**Mean**	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	**1-shot**	**3-shot**
S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
DANN [5]	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
ENT [6]	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
APE [10]	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
DECOTA [31]	79.1	80.4	74.9	75.2	76.9	78.7	65.1	68.6	72.0	72.7	69.7	71.9	79.6	81.5	73.9	75.6
MCL [30]	77.4	79.4	74.6	**76.3**	75.5	78.8	66.4	70.9	**74.0**	**74.7**	70.7	72.3	**82.0**	**83.3**	74.4	76.5
MME [21]	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
MME + SLA (ours)	71.8	73.3	68.2	70.1	70.4	72.7	59.3	63.4	64.9	67.3	61.8	63.9	77.2	79.6	68.8	70.0
CDAC [12]	77.4	79.6	74.2	75.1	75.5	79.3	67.6	69.9	71.0	73.4	69.2	72.5	80.4	81.9	73.6	76.0
CDAC + SLA (ours)	**79.8**	**81.6**	**75.6**	76.0	**77.4**	**80.3**	**68.1**	**71.3**	71.7	73.5	**71.7**	**73.5**	80.4	82.5	**75.0**	**76.9**

Table 4. Accuracy (%) on *DomainNet* for 1-shot and 3-shot Semi-Supervised Domain Adaptation (ResNet34).

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Mean
							One-shot						
S+T	50.9	69.8	73.8	56.3	68.1	70.0	57.2	48.3	74.4	66.2	52.1	78.6	63.8
DANN [5]	52.3	67.9	73.9	54.1	66.8	69.2	55.7	51.9	68.4	64.5	53.1	74.8	62.7
ENT [6]	52.9	75.0	76.7	63.2	73.6	73.2	63.0	51.9	79.9	70.4	53.6	81.9	67.9
APE [10]	53.9	76.1	75.2	63.6	69.8	72.3	63.6	58.3	78.6	72.5	60.7	81.6	68.9
DECOTA [31]	42.1	68.5	72.6	60.3	70.4	70.7	60.0	48.8	76.9	71.3	56.0	79.4	64.8
MME [21]	59.6	75.5	77.8	65.7	74.5	74.8	64.7	57.4	79.2	71.2	61.9	82.8	70.4
MME + SLA (ours)	62.1	76.3	78.6	**67.5**	77.1	75.1	66.7	59.9	80.0	**72.9**	64.1	83.8	72.0
CDAC [12]	61.2	75.9	78.5	64.5	75.1	75.3	64.6	59.3	80.0	72.7	61.9	83.1	71.0
CDAC + SLA (ours)	**63.0**	**78.0**	**79.2**	66.9	**77.6**	**77.0**	**67.3**	**61.8**	**80.5**	72.7	**66.1**	**84.6**	**72.9**
							Three-shot						
S+T	54.0	73.1	74.2	57.6	72.3	68.3	63.5	53.8	73.1	67.8	55.7	80.8	66.2
DANN [5]	54.7	68.3	73.8	55.1	67.5	67.1	56.6	51.8	69.2	65.2	57.3	75.5	63.5
ENT [6]	61.3	79.5	79.1	64.7	79.1	76.4	63.9	60.5	79.9	70.2	62.6	85.7	71.9
APE [10]	63.9	81.1	80.2	66.6	79.9	76.8	66.1	65.2	82.0	73.4	66.4	86.2	74.0
DECOTA [31]	64.0	81.8	80.5	68.0	**83.2**	79.0	69.9	68.0	82.1	74.0	**70.4**	**87.7**	75.7
MME [21]	63.6	79.0	79.7	67.2	79.3	76.6	65.5	64.6	80.1	71.3	64.6	85.5	73.1
MME + SLA (ours)	65.9	81.1	80.5	**69.2**	81.9	79.4	69.7	67.4	81.9	**74.7**	68.4	87.4	75.6
CDAC [12]	65.9	80.3	80.6	67.4	81.4	**80.2**	67.5	67.0	81.9	72.2	67.8	85.6	74.8
CDAC + SLA (ours)	**67.3**	**82.6**	**81.4**	**69.2**	82.1	80.1	**70.1**	**69.3**	**82.5**	73.9	70.1	87.1	**76.3**

Table 5. Accuracy (%) on *Office-Home* for 1-shot and 3-shot Semi-Supervised Domain Adaptation (ResNet34).

of combining all pairs. Our experiments follow the settings in recent works [12,21,30], with the same sampling strategy for both the training set and validation set, and we conduct both 1-shot and 3-shot settings on all datasets.

Implementation Details. Our framework can be applied with many state-of-the-art methods, we choose MME [21], and CDAC [12] as our cooperators to validate the efficacy of our method, named MME + SLA and CDAC + SLA, respectively. For a fair comparison, we choose ResNet34 [7] as our backbone. The backbone is pre-trained on ImageNet-1K dataset [4], and the model architecture, batch size, learning rate scheduler, optimizer, weight-decay, and initialization strategy are all followed as previous works [12, 21, 30]. We follow the same hyper-parameters for MME and CDAC as their suggestions. We set the mix ratio α in Eq. 12 to 0.3 and the temperature parameter T in Eq. 7 to 0.6. The update interval I mentioned in Section 3.3 is 500. The warmup parameter W in Eq. 12 is 500 for MME on *Office-Home*; 2000 for CDAC on *Office-Home*; 3000 for MME on *DomainNet*; 50000 for CDAC on *DomainNet*. After the warmup stage, we refresh the learning rate scheduler so that the label adaptation loss can be updated with a higher learning rate. All hyper-parameters can be properly tuned via the validation process. For each subtask, we conducted the experiments three times. The detailed statistics of our results can be found in our supplementary materials.

4.2. Comparison with State-of-the-Arts

We compare our results with several baselines, including **S+T**, **DANN** [5], **ENT** [6], **MME** [21] **APE** [10], **CDAC** [12], **DECOTA** [31], **MCL** [30]. **S+T** is a baseline method for SSDA, with only source data and labeled target data involved in the training process. **DANN** is a classic unsupervised domain adaptation method, and [21] reproduces it by training with additional labeled target data. **ENT** is a standard entropy minimization originally designed for Semi-Supervised Learning, and the reproduction was also done by [21]. Note that for MCL, we only compare with their

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	**Mean**
MCL [30]	**67.5**	**83.9**	**82.4**	**71.4**	**84.3**	**81.6**	69.9	**68.0**	**83.0**	75.3	**70.1**	**88.1**	**77.1**
MCL*	64.1	81.6	80.6	70.3	82.2	79.2	70.6	64.0	81.8	75.3	67.8	86.6	75.3
MCL + SLA (ours)	64.3	81.6	80.8	70.2	82.6	79.4	**70.9**	64.2	82.2	**75.5**	68.0	86.8	75.6

*: Reproduced by ourselves

Table 6. Accuracy (%) of MCL and MCL + SLA on *Office-Home* for 3-shot Semi Supervised Domain Adaptation (ResNet34).

results on *DomainNet*. We leave the detailed analysis for MCL on *Office-Home* in Section 4.3.

***DomainNet*.** We show the results on *DomainNet* dataset with 1-shot and 3-shot settings on Table 4. It is worth noting two things. First, for MME and CDAC, almost all sub-tasks get improvement after applying our SLA framework, except for only two cases where CDAC + SLA performs roughly the same as CDAC. Second, the overall performance of CDAC + SLA for 1-shot and 3-shot settings reaches 75.0% and 76.9%, respectively; both outperform the previous methods and achieve new state-of-the-art results.

***Office-Home*.** We show the results on *Office-Home* dataset with 1-shot and 3-shot settings on Table 5. Similarly, after applying SLA to MME and CDAC, the performances get much better except for only one case under the 3-shot setting. Overall, our framework improves the original works by at least 1.5% under all settings.

4.3. Analysis

Reproducibility issue for MCL. MCL [30] designs consistency regularization for SSDA at three different levels and achieves excellent results. However, our experiments cannot fully reproduce their reported numbers. The reproduced results on 3-shot *Office-Home* dataset are shown in Table 6. After applying our SLA framework, although we can stably improve our reproduction, we are still unable to compete with their reported values. We put our detailed reproducing results into the supplementary materials, and the implementation is avaiable at `https://github.com/chester256/MCL`.

The intermediate results in SLA. In SLA, we build a PPC to provide the view from the target data. PPC can be viewed as a variant of the pseudo-labeling method proposed in [13]. They apply such a method to boost their final performance in their work. If PPC has performed well, a natural question is: *Is it necessary to modify source labels by PPC?*

To reveal the intermediate steps within SLA, we plot the test performance of MME (red), MME + PPC (orange), MME + SLA (blue), and PPC within MME + SLA (purple) during the training phase in Figure 5. Initially, PPC (target view) performs at a higher level. However, if the source labels are not adapted, it will end up converge to the same performance as MME. In contrast, within our SLA framework, the model leverages the benefits of PPC, further

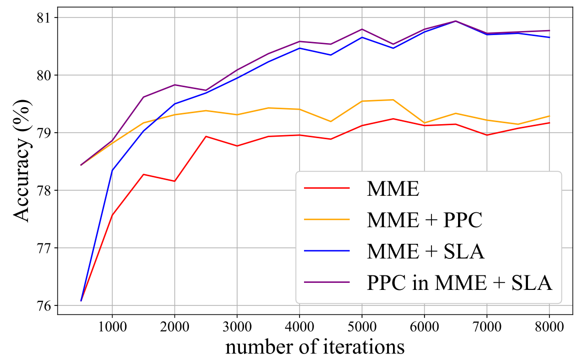

Figure 5. The intermediate results in SLA on 3-shot *Office-Home* A → P with ResNet34.

α	0.1	0.3	0.5	0.7	0.9
validation	74.87	**75.94**	75.76	75.38	72.39
test	74.43	**75.57**	75.38	74.79	72.76

Table 7. Average accuracy (%) of MME + SLA on 3-shot *Office-Home* with ResNet34.

producing an enhanced version of PPC, resulting in better overall performance compared to the original MME.

Sensitivity study of α. Table 7 shows the sensitivity study results for α using MME + SLA on 3-shot *OfficeHome* dataset. We selected 0.3 from the best validation performance and kept 0.3 throughout all experiments for simplicity and resource saving. In the mid-range, $\alpha = 0.3$ (favoring source view) or $\alpha = 0.7$ (favoring target view) perform similarly, hinting that it is stable enough for a proper range of choices. It is worth noting that the dynamic adjustment of α is a promising direction. We proposed the warmup stage that changes α from 0 to the desired value after a period of warming up. More sophisticated scheduling techniques, such as the linear growth approach [9], could potentially replace the warmup parameter and lead to further improvements. We leave exploration of these techniques for future work.

Illustration of the adapted labels. We aim to adapt the original source label $\mathbf{y}_i^s$ to the ideal label $g^*(\mathbf{x}_i^s)$. To demonstrate the change of labels, we visualize the top 3 probabilities of the average adapted source labels on two classes in Figure 6. For Backpack (class 1), ideally-adapted

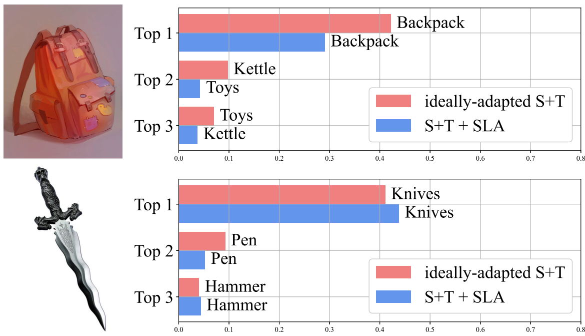

Figure 6. Top-3 probabilities of the average adapted source labels on 3-shot *Office-Home* A $\rightarrow$ C with ResNet34. **Top.** Class 1 (backpack). **Bottom.** Class 30 (knives).

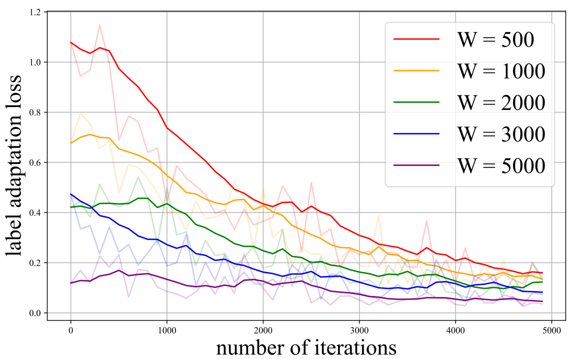

Figure 7. Label Adaptation Loss of MME + SLA by first pre-training MME for W iterations on 3-shot *Office-Home* A $\rightarrow$ C with ResNet34. (Smoothing by EMA with a ratio 0.8.)

S+T should change the source label to 40% Backpack + 10% Kettle + 8% Toys; SLA proposes to change to 30% Backpack + 5% Toys + 4% Kettle, which is closer to the ideally-adapted label than the original label of 100% Backpack. We draw the same conclusion for knives (class 30).

Warmup issue for MME + SLA. As described in Section 3.3, our framework relies on the quality of the predicted pseudo labels. Thus, we introduce a warmup stage parameter W to derive a robust model. We can treat the warmup strategy as a two-stage algorithm. Take MME as our backbone method; the algorithm works like:

1. Train a model with normal MME loss for W iteration.
2. Take the model above as a pre-trained model and further applying label adaptation loss.

For the first step, intuitively, we should train the model until the loss converges. That is how we select the warmup stage parameter for CDAC + SLA. However, empirically we found that the performance of MME + SLA will degrade if we train an MME model until it converges. Table 8 shows

	A $\rightarrow$ C	
Warmup Stage (W)	1-shot	3-shot
500	**62.09**	**65.90**
1000	61.95	64.99
2000	61.37	64.72
3000	61.53	64.87
5000	61.79	64.68

Table 8. Accuracy (%) for different warmup stage W of MME + SLA on *Office-Home* A $\rightarrow$ C with ResNet34.

the sensitivity test of W using MME + SLA on *Office-Home* dataset. We can observe that no matter the 1-shot or 3-shot settings, the performance is generally getting worse as the number of warmup stages increases. To analyze the effect, we first pre-train a normal MME for W iterations, then observe the label adaptation loss of MME + SLA. Figure 7 plots the label adaptation loss of MME + SLA by first pre-training MME for W iterations. We can observe that when $W = 5000$, the initial label adaptation loss has already been close to 0. Doing label adaptation in the situation is almost equivalent to not doing so, as we mentioned in Section 3.1.

Limitation. Our SLA framework might not be helpful if the label adaptation loss approaches 0. Although we have applied the Protonet with Pseudo Centers to avoid the issue, the loss will converge to 0 in MME + SLA. We leave the analysis of the reason for the convergence as a future work. On the other hand, we argue that it is unnecessary to discuss the reason in our proposed scope since we can make a trade-off by carefully tuning the warmup parameter W, and the issue turns out to be part of the hyper-parameters selection.

5. Conclusion

In this work, we present a general framework, *Source Label Adaptation* for Semi-Supervised Domain Adaptation. Our work highlights that the usage of source data should be revisited carefully. We argue that the original source labels might be noisy from the perspective of target data. We approach Domain Adaptation as a Noisy Label Learning problem and correct source labels with the predictions from Protonet with Pseudo Centers. Our approach mainly addresses an issue that is orthogonal to other existing works, which focus on improving the usage of unlabeled data. The empirical results show that we can apply our framework to several state-of-the-art algorithms for SSDA and further boost their performances.

Acknowledgement. We thank the anonymous reviewers for valuable comments. This work is supported by the Ministry of Science and Technology in Taiwan via the grant MOST 111-2628-E-002-018. We thank to National Center for High-performance Computing (NCHC) in Taiwan for providing computational and storage resources.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014. 1

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

[3] David M Chan, Roshan Rao, Forrest Huang, and John F Canny. Gpu accelerated t-distributed stochastic neighbor embedding. *Journal of Parallel and Distributed Computing*, 131:1–13, 2019. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 6

[6] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 1, 3, 6

[7] 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*, pages 770–778, 2016. 3, 6

[8] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 1

[9] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6567–6576, 2021. 7

[10] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European conference on computer vision*, pages 591–607. Springer, 2020. 6

[11] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013. 1

[12] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2505–2514, 2021. 1, 3, 6

[13] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 7

[14] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021. 5

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

[16] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1094–1103, June 2021. 1

[17] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017. 3

[18] 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*, pages 1406–1415, 2019. 5

[19] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753. Springer, 2017. 1

[20] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 3, 4

[21] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019. 1, 2, 5, 6

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

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3, 4

[24] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1, 3

[25] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014. 3

[26] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5552–5560, 2018. 3

[27] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 1, 5

[28] Xinshao Wang, Yang Hua, Elyor Kodirov, Sankha Subhra Mukherjee, David A Clifton, and Neil M Robertson. Proselflc: Progressive self label correction towards a low-temperature entropy state. *arXiv preprint arXiv:2207.00118*, 2022. 2, 4

[29] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020. 3

[30] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. *arXiv preprint arXiv:2205.04066*, 2022. 1, 3, 6, 7

[31] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8906–8916, 2021. 3, 6

[32] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems*, 33:7260–7271, 2020. 3

[33] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 1

[34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 3

[35] Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021. 2

[36] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019. 1

[37] Yin Zhao, Longjun Cai, et al. Reducing the covariate shift by mirror samples in cross domain alignment. *Advances in Neural Information Processing Systems*, 34:9546–9558, 2021. 1

[38] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11053–11061, 2021. 3