國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Graduate Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

修正源標籤以改進半監督式域適應

Semi-Supervised Domain Adaptation with Source Label Adaptation

余友竹

Yu-Chu Yu

指導教授: 林軒田 博士

Advisor: Hsuan-Tien Lin Ph.D.

中華民國 111 年 12 月

December, 2022

doi:10.6342/NTU202210189

Acknowledgements

I would like to express my sincere gratitude to the people who helped me complete my master thesis. First and foremost, I thank my advisor, Professor Hsuan-Tien Lin, for his guidance and support throughout the process. His expertise and knowledge were invaluable, and I learned much from him. During my research process, I encountered numerous setbacks and periods where I struggled to develop ideas. However, with the help of Professor Lin's experience, we identify the key points to address the problem, allowing me to complete this work. His dedication to the work and his passion for teaching is truly inspiring, and I am fortunate to have had him as my advisor.

I would also like to thank my oral defense committee members, Professor Hung-Yi Lee and Professor Chu-Song Chen, for their assistance in completing my oral defense and for providing me with professional recommendations to improve my master thesis. Their support and expertise have been very precious to me.

I would particularly like to thank the members of the CLLab, Si-An Chen, Po-Yi Lu, Mai Tấn Hà, Yu-Hsin Chou, Oscar Chew, Wei-I Lin, and the alumni members, Wei-Chao Cheng, Sheng-Feng Wu, who provided valuable feedback and ideas throughout the process. Their support was crucial in helping me complete my thesis, and I am lucky to have been a part of such a collaborative and supportive team.

doi:10.6342/NTU202210189

I am grateful to my classmates, Pin-Yen Huang and Chi-Chang Lee, for their support and collaboration. We worked together on our research and encouraged each other to ensure progress on schedule. I really appreciate their valuable feedback and ideas, which helped me improve my work.

I am also grateful to my family, especially my parents, for their support. I could not have done it without them. Besides, I am truly grateful to have my girlfriend by my side. She has been a constant source of love and support throughout the process of completing my master thesis. She always believed in me and encouraged me to keep going, even on the toughest days. I am also thankful for my dog. Whenever I needed a break from my studies, my dog was there to provide a much-needed distraction and to help me relax. Having such wonderful people and animals in my life is definitely one of my luckiest things.

Finally, I would like to thank the Ministry of Science and Technology and National Center for High-performance Computing for providing computational and storage resources. This work would not have been completed without their contribution.

I truly appreciate support from such wonderful people during the completion of my master thesis. Thank you all once again for your help.

doi:10.6342/NTU202210189

摘要

半監督式域適應涉及到學習使用少量的標記目標數據和許多未標記的目標數據，以及來自相關領域的標記源數據，以對未標記的目標數據進行分類。當前的半監督式域適應方法通常旨在通過特徵空間映射和偽標籤分配將目標數據與標記的源數據對齊。然而，這種源導向的模型有時會將目標數據與錯誤類別的源數據對齊，從而降低分類的表現。我們提出了一種新穎的域適應典範，可以調整源數據以匹配目標數據。我們的核心思想是將源數據視為一種含有噪聲標記的理想目標數據。我們提出了一個半監督式域適應模型，該模型借助從目標的角度設計的清理元件來動態清除源數據的噪聲標籤。由於這種想法與現有的其他半監督式域適應方法背後的核心理念有很大的不同，因此，我們提出的模型可以很容易地與這些方法結合以提高它們的性能。在兩種主流的半監督式域適應方法上的實驗結果表明，我們提出的模型有效地清除了源標籤內的噪聲，並在主流的數據集上得到優於這些方法的表現。

關鍵字：域適應、半監督式域適應、機器學習、遷移學習、噪聲標籤學習

doi:10.6342/NTU202210189

Abstract

Semi-supervised domain adaptation (SSDA) involves learning to classify unseen target data with a few labeled data and many unlabeled target data, along with many labeled source data from a related domain. Current SSDA approaches typically aim at aligning the target data to the labeled source data with feature space mapping and pseudo-label assignment. Nevertheless, such a source-oriented model sometimes aligns the target data to source data of the wrong class, degrading the classification performance. We present 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. We propose an SSDA model that cleans up the label noise dynamically with the help of a robust cleaner component designed from the perspective of the target. Since this paradigm differs greatly from the core ideas behind existing SSDA approaches, our proposed model can be easily coupled with such approaches to improve their performance. Empirical results on two state-of-the-art SSDA approaches demonstrate that the proposed model ef-

doi:10.6342/NTU202210189

fectively cleans up noise within the source labels and exhibits superior performance over those approaches across benchmark datasets.

Keywords: Domain Adaptation, Semi-Supervised Domain Adaptation, Machine Learning, Transfer Learning, Noisy Label Learning

doi:10.6342/NTU202210189

Contents

doi:10.6342/NTU202210189

doi:10.6342/NTU202210189

List of Figures

doi:10.6342/NTU202210189

doi:10.6342/NTU202210189

List of Tables

doi:10.6342/NTU202210189

doi:10.6342/NTU202210189

Chapter 1 Introduction

Domain adaptation (DA) focuses on a general machine learning scenario where training and test data originate from two related but distinct domains: the source domain and the target domain. Extensive studies have been conducted on unsupervised DA (UDA), where no labels in the target domain can be accessed, from theoretical [2, 18, 35] and algorithmic [5, 8, 14, 15, 21, 36] angles. Recently, semi-supervised domain adaptation (SSDA), another DA setting that allows access to a few target labels, has become popular as it is simple but reflects the needs of real-world applications.

The most naive strategy for SSDA, commonly known as S+T [20, 32] is to train the model using source data and labeled target data with standard cross entropy loss. This strategy is generally vulnerable to the well-known domain shift problem, which stems from the gap between different data distributions. To address this issue, many state-of-the-art algorithms explore better use of unlabeled target data to align the target distribution with the source distribution. Recently, semi-supervised learning (SSL) algorithms have been adopted for SSDA [11, 20, 29] to regularize unlabeled data via entropy minimization [6], pseudo-labeling [10, 23], and consistency regularization [1, 23]. These classic source-oriented strategies have prevailed for a long time. However, they usually overlook the potential of making the alignment bi-directional. Therefore, once the S+T space has been misaligned, it is generally hard to escape the situation illustrated in Figure 1.1.

doi:10.6342/NTU202210189

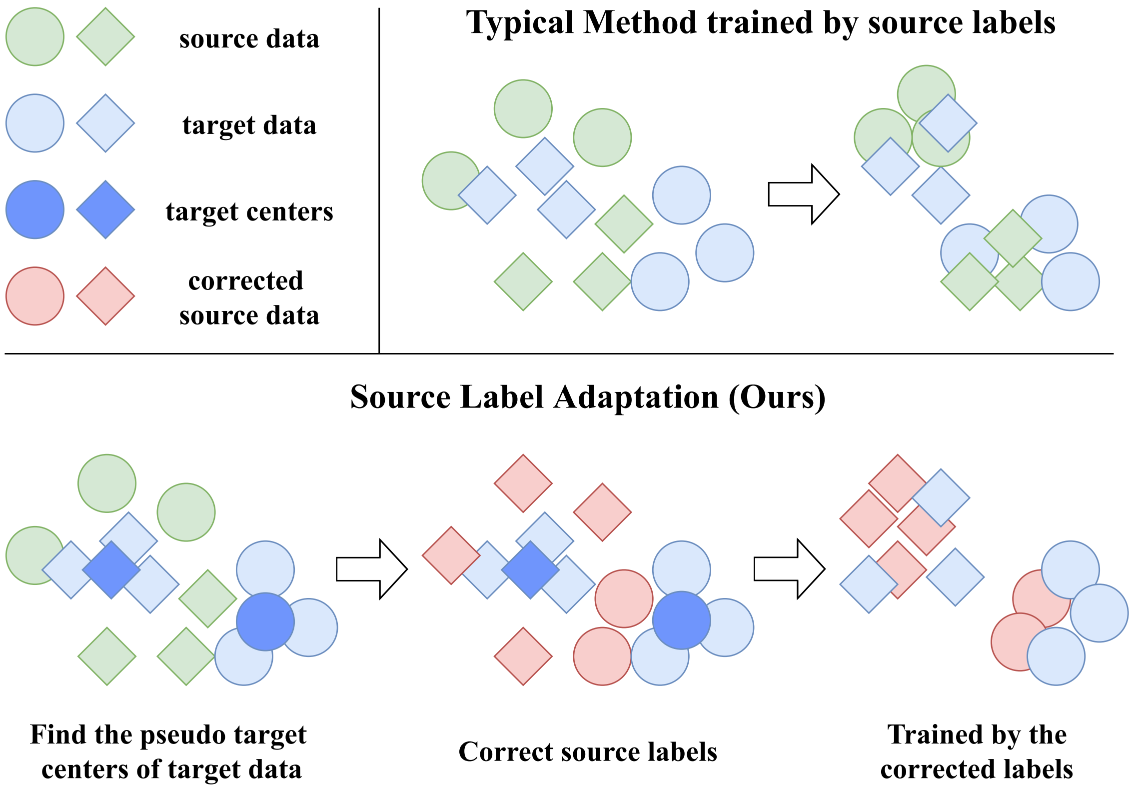

Figure 1.1: **Top.** Training the model with the original source labels can produce misaligned target data. **Bottom.** After cleaning up noisy source labels with our SLA framework, the target data aligns with the correct classes.

doi:10.6342/NTU202210189

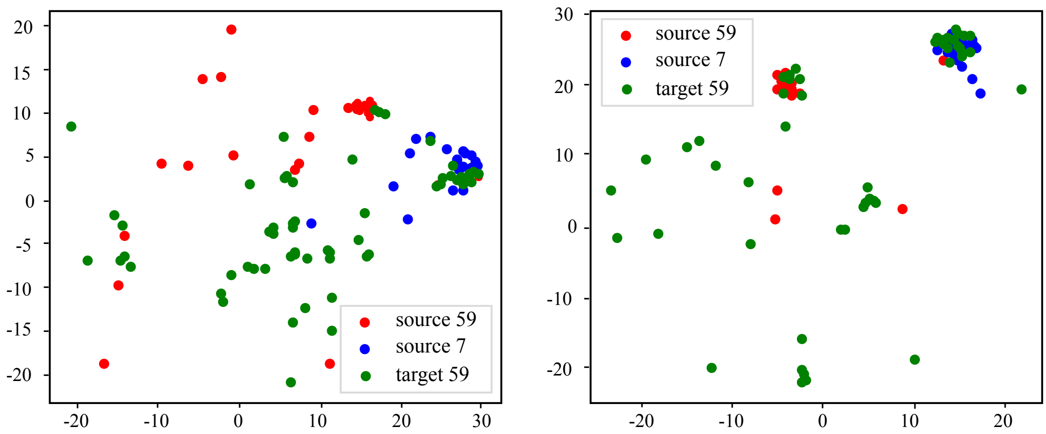

Figure 1.2: T-SNE feature visualizations that illustrate misalignment on Office-Home A $\rightarrow$ C dataset with ResNet34. The model is trained by S+T. **Left**: 0-th iteration. **Right**: 5000-th iteration: misalignment has already occurred at an early stage. Guided by source labels and a few target labels, a portion of the target data from the 59th class misaligns with 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 1.1: Partial confusion matrix of S+T on 3-shot Office-Home A $\rightarrow$ C dataset with ResNet34. About 40% of the target data from the 59th class is wrongly classified as the 7th class. Only about 20% of the data is predicted correctly.

We take a deeper look at a specific example from the Office-Home dataset [26] to confirm this issue. Figure 1.2 visualizes the feature space trained by S+T using t-SNE [3]. We observed that misalignment between the source and target data has occurred at an 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 have access only to the 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 1.1 shows the partial confusion matrix of S+T. Roughly 40% of the target data in the 59th class is mispredicted as the 7th class, and only around 20% of the data is classified correctly.

From the case study above, we argue that relying on source labels like S+T misguides the model to learn the wrong classes for some target data. That is, source labels can be

doi:10.6342/NTU202210189

viewed as a "noisy" version of the ideal labels for target classification.

Based on this conjecture, the SSDA setting 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: this is also known as label correction [27]. To approach domain adaptation as an NLL problem, we borrow from label correction by proposing a source label adaptation (SLA) framework, as shown in Figure 1.1. We construct a label adaptation component that provides the view from the target data and dynamically cleans up noisy source labels at each iteration. Unlike other earlier studies that study how to regularize unlabeled data, we mainly investigate how to train source data with adapted labels to better reflect the ideal target space. This source-adaptive paradigm is entirely different from the core ideas behind existing SSDA approaches. Thus, we can combine our framework with other strategies to produce superior results. We summarize our contributions as follows.

- We argue that classic source-oriented methods are still characterized by a biased feature space from S+T. We address this by adapting the source data to the target space by modifying the original source labels.

- We address DA as a particular NLL problem and present a novel source-adaptive paradigm. As our SLA framework can be coupled with other methods, the adaptation can be bi-directional, further enhancing performance.

- We demonstrate the usefulness of our proposed SLA framework when coupled with state-of-the-art SSDA algorithms. The framework significantly improves existing

doi:10.6342/NTU202210189

algorithms on two major benchmarks, inspiring a new direction for solving DA problems.

doi:10.6342/NTU202210189

Chapter 2 Related Work

2.1 Problem Setup

DA focuses on K-class classification 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 the k-th element is 1 and all others are 0. We consider two domains over $X \times Y$: the 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 $|U|$, for instance, 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.

doi:10.6342/NTU202210189

2.2 Semi-Supervised Domain Adaptation (SSDA)

SSDA can be viewed as a relaxed yet realistic version of UDA. An SSDA algorithm usually involves three loss functions:

$$\mathcal{L}_{\text{SSDA}} = \mathcal{L}_s + \mathcal{L}_\ell + \mathcal{L}_u \tag{2.1}$$

where $\mathcal{L}_s$ stands for the loss derived by the source data and $\mathcal{L}_\ell$ and $\mathcal{L}_u$ denote the losses from the labeled and unlabeled target data. As discussed in Section **??**, 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. Many recent methods tackle SSDA using SSL techniques because of the problem similarity [34]. [20] proposes a variant of entropy minimization [6] to explicitly align the target data with source clusters. [30] 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. [11] groups target features into clusters by measuring pairwise feature similarity. [29] utilizes consistency regularization at three different levels to perform domain alignment. In addition, [11] and [29] both apply pseudo labeling with data augmentations [23] for improved performance. To the best of our knowledge, these methods primarily explore the usage of unlabeled target data while adopting the most straightforward strategy for the source data. In our study, we observe that source labels can seem noisy from the viewpoint of the target data. We thus develop a source-adaptive framework to gradually adapt the source data to the target space. Since we are addressing a new facet of this problem, our framework can be easily applied to the SSDA algorithms mentioned above, further improving the overall performance.

doi:10.6342/NTU202210189

2.3 Noisy Label Learning (NLL)

The effectiveness of a machine learning algorithm depends greatly on the quality of the collected labels. In particular, for current deep neural network architectures [7], such problems might become much worse as a deep model can usually arbitrarily fit the dataset even if the labels are random [33]. To clean the noisy labels, [19] proposes a smoothing mechanism to mix noisy labels with self-prediction. [25] models clean labels as trainable parameters and uses joint optimization to alternatively update parameters. [16, 24, 31] estimate a transition matrix to correct the corrupted labels. However, learning a global transition matrix usually requires a strong assumption concerning the source of noisy labels, which is hard to verify in real-world scenarios [28]. [37] trains a label correction network in a meta-learning manner to help correct noisy labels. Motivated by [19, 37], we propose a simple framework that efficiently builds a label adaptation model. By modifying the source labels, we adapt the noisy source labels to better fit the ideal labels for target classification.

doi:10.6342/NTU202210189

Chapter 3 Proposed Framework

We propose source label adaptation (SLA), a novel SSDA framework. An overview of our proposed framework is provided in Figure 3.1. In Section 3.1, we connect the (SS)DA problem to NLL and show that a classic NLL method [19] cannot be directly applied to solve SSDA. In Section 3.2, we review the prototypical network [22], a classic few-shot learning algorithm, and propose protonet with pseudo centers (PPC) to better estimate the prototypes. In Section 3.3, we summarize our framework and describe the implementation in detail.

doi:10.6342/NTU202210189

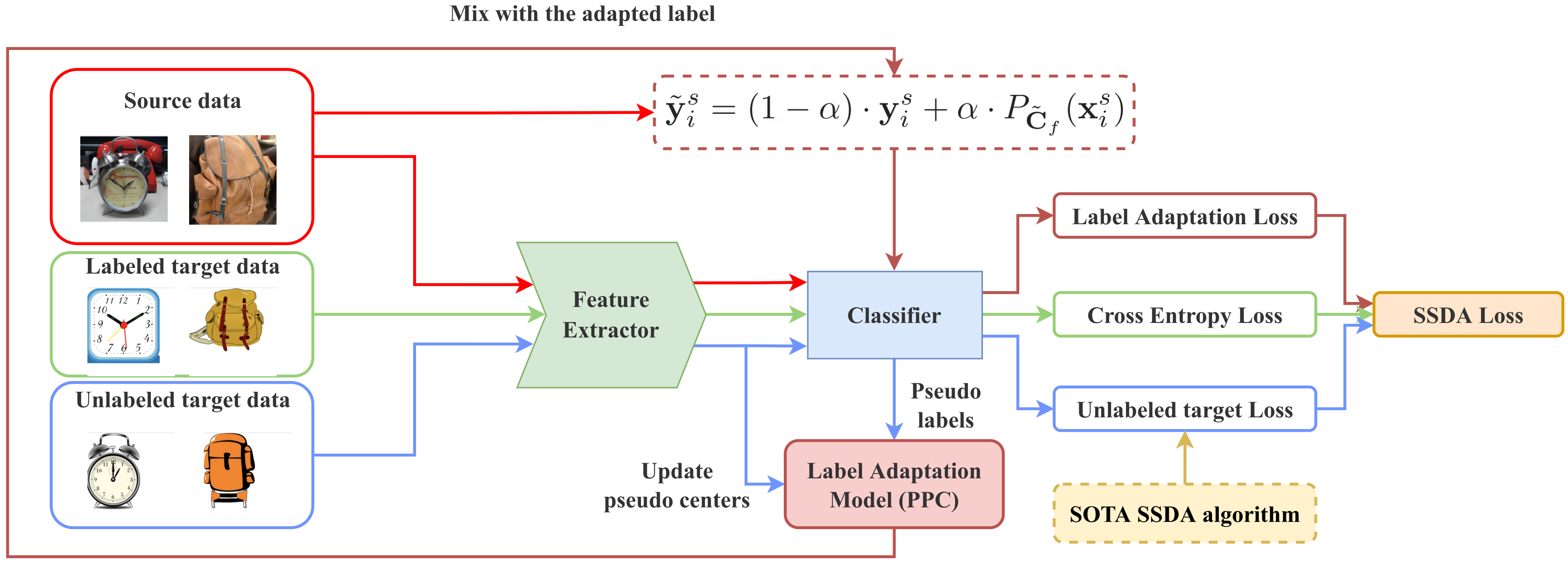

Figure 3.1: Overview of source label adaptation for SSDA. For source data, we adapt the original source labels to better fit the target feature space using PPC (protonet with pseudo centers) and calculate the label adaptation loss. We train using labeled target data with standard cross entropy loss. 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 produce a more reliable label adaptation model.

doi:10.6342/NTU202210189

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

Table 3.1: Accuracy (%) of S+T and ideal S+T on 3-shot OfficeHome dataset with ResNet34. In the ideal case, where we have access to the ideal target model, the performance is dramatically influenced simply by modifying the source labels to match the target view.

3.1 Domain Adaptation as Noisy Label Learning

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

$$\mathcal{L}_s^*(g|S) = \frac{1}{|S|}\sum_{i=1}^{|S|} \ell_{\text{ce}}(g(\mathbf{x}_i^s), g^*(\mathbf{x}_i^s)). \tag{3.1}$$

Combined with the labeled target loss $\mathcal{L}_\ell$, we refer to the model trained by $\mathcal{L}_s^*$ and $\mathcal{L}_\ell$ as ideal S+T. A normal S+T and an ideal S+T are compared in Table 3.1: performance is influenced dramatically simply by modifying the source labels.

In practice, however, we can only approximate the ideal model. We thus 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 [19] to help correct the source labels; we refer to this as *label correction with self-prediction* [27]. 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 of α:

$$\hat{\mathbf{y}}_i^s = (1-\alpha)\cdot\mathbf{y}_i^s + \alpha\cdot g(\mathbf{x}_i^s). \tag{3.2}$$

doi:10.6342/NTU202210189

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

$$\hat{\mathcal{L}}_s(g|S) = \frac{1}{|S|}\sum_{i=1}^{|S|} \ell_{\text{ce}}(g(\mathbf{x}_i^s), \hat{\mathbf{y}}_i^s). \tag{3.3}$$

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,

$$\begin{aligned} \hat{\mathbf{y}}_i^s &= (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot g(\mathbf{x}_i^s) \\ &\approx (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot \mathbf{y}_i^s = \mathbf{y}_i^s. \end{aligned} \tag{3.4}$$

Figure 3.2 shows that when performing label correction with self-prediction, the KL divergence from $\mathbf{y}^s$ to $g(\mathbf{x}^s)$ approximates 0 after 2000 iterations, indicating that self-prediction is almost the same as the original label. In this case, label correction is nearly equivalent to doing nothing.

To benefit from the modified labels, we must eliminate supervision from the 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 approximates the ideal model and adapt the source labels to the view of the target data. We define an adapted label $\tilde{\mathbf{y}}_i^s$ as a convex combination between the original label $\mathbf{y}_i^s$ and the output from g_c, which is the same as [19]:

$$\tilde{\mathbf{y}}_i^s = (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot g_c(\mathbf{x}_i^s). \tag{3.5}$$

doi:10.6342/NTU202210189

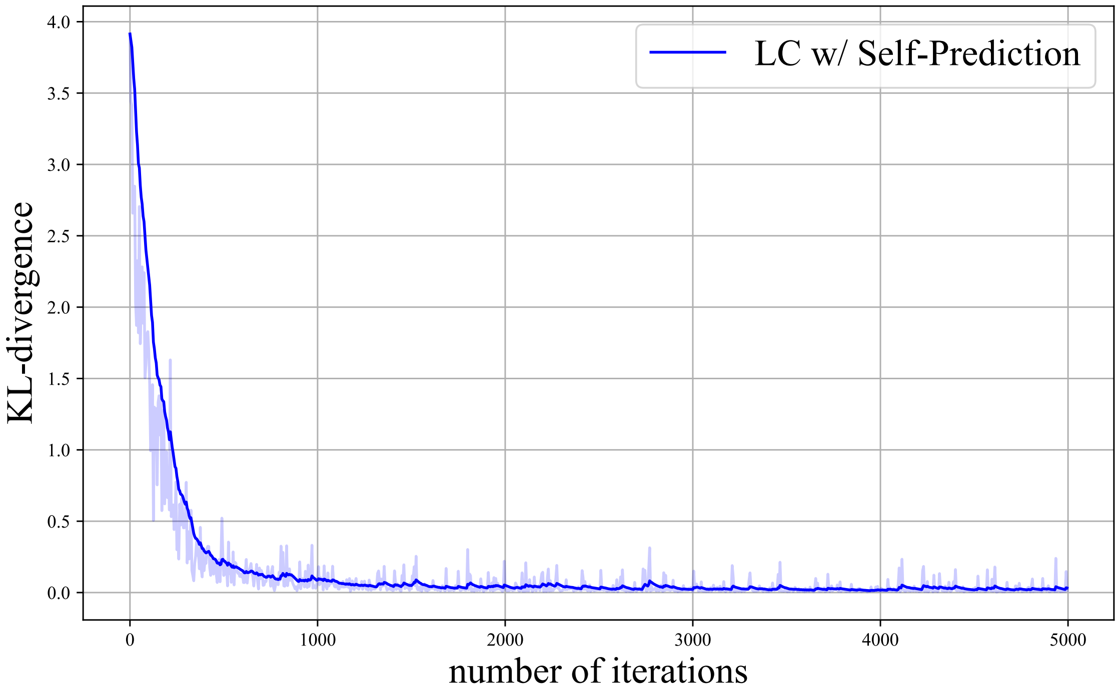

Figure 3.2: Average KL divergence from $\mathbf{y}^s$ to $g(\mathbf{x}^s)$ at each iteration (3-shot Office-Home A $\rightarrow$ C with ResNet34, smoothing by EMA with a ratio of 0.8)

3.2 Protonet with Pseudo Centers

In the semi-supervised setting, we are given access to a few target labels. Nonetheless, learning from a limited number of target labels can lead to severe overfitting. Thus, we learn a prototypical network (protonet) [22] to mitigate 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 the features with the same class:

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{i=1}^{N} \mathbb{1}\{y_i = k\} \cdot f(\mathbf{x}_i). \tag{3.6}$$

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

doi:10.6342/NTU202210189

as a protonet with centers $\mathbf{C}_f$:

$$P_{\mathbf{C}_f}(\mathbf{x}_i)_k = \frac{\exp(-d(f(\mathbf{x}_i), \mathbf{c}_k) \cdot T)}{\sum_{j=1}^{K} \exp(-d(f(\mathbf{x}_i), \mathbf{c}_j) \cdot T)}. \tag{3.7}$$

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

Since we have access to the labeled target dataset L, by Eqs. 3.6 and 3.7, we can derive labeled target centers $\mathbf{C}_f^\ell$ and construct a protonet with labeled target centers $P_{\mathbf{C}_f^\ell}$.

When d measures Euclidean distance, a protonet is equivalent to a linear classifier with a particular parameterization over F [22]. Thus, we can take the protonet as a label adaptation model over a particular feature space. The protonet with labeled target centers is built purely from the viewpoint of the target data, which should reduce our concerns about the issue mentioned in Section 3.1.

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 located far from the ideal centers $\mathbf{C}_f^*$. To better estimate the ideal centers, we propose finding 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

$$\tilde{y}_i^u = \arg\max_k g_s(\mathbf{x}_i^u)_k. \tag{3.8}$$

After deriving unlabeled target data with pseudo labels $\{(\mathbf{x}_i^u, \tilde{y}_i^u)\}_{i=1}^{|U|}$, we obtain pseudo centers $\tilde{\mathbf{C}}_f$ by Eq. 3.6 and further define a protonet with pseudo centers (PPC) $P_{\tilde{\mathbf{C}}_f}$ by

doi:10.6342/NTU202210189

From / to	Labeled target centers	Pseudo centers
Ideal centers	10.02	4.06

Table 3.2: 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)

Eq. 3.7.

Table 3.2 compares the average L2 distance from the ideal centers $\mathbf{C}_f^*$ to the labeled target centers $\mathbf{C}_f^\ell$ and the 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 that between $\mathbf{C}_f^\ell$ and $\mathbf{C}_f^*$, which indicates that 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

$$\tilde{\mathbf{y}}_i^s = (1-\alpha) \cdot \mathbf{y}_i^s + \alpha \cdot P_{\tilde{\mathbf{C}}_f}(\mathbf{x}_i^s). \tag{3.9}$$

3.3 Source Label Adaptation for SSDA

We propose a label adaptation loss for source data to replace the typical source loss with 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. 3.9. Then, the label adaptation loss $\tilde{L}^s$ is

$$\tilde{\mathcal{L}}_s(g|S) = \frac{1}{|S|}\sum_{i=1}^{|S|} \ell(g(\mathbf{x}_i^s), \tilde{\mathbf{y}}_i^s). \tag{3.10}$$

The proposed SLA for SSDA framework can be trained by the following loss function:

$$\mathcal{L}_{\text{SSDA w/ SLA}} = \tilde{\mathcal{L}}_s + \mathcal{L}_\ell + \mathcal{L}_u. \tag{3.11}$$

doi:10.6342/NTU202210189

L^ℓ is the loss function for labeled target data L, which can still be standard cross entropy loss. In contrast to other widely used methods, we concentrate primarily on improving the usage of the source data. Therefore, the loss function for unlabeled target data L^u can be derived using any state-of-the-art algorithm, and our framework can be easily coupled with other methods without contradiction.

3.3.1 Implementation Details

3.3.1.1 Warmup Stage

Our label adaptation framework relies on the quality of the predicted pseudo labels. However, as prediction from the initial model can be noisy, we introduce a warmup hyper-parameter W to produce 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

$$\tilde{\mathbf{y}}_i^s = \begin{cases} \mathbf{y}_i^s & \text{if } e \leq W \\ (1-\alpha) \cdot \mathbf{y}_i^s + \alpha \cdot P_{\tilde{\mathbf{C}}_f}(\mathbf{x}_i^s) & \text{otherwise.} \end{cases} \tag{3.12}$$

3.3.1.2 Dynamic Updates

The feature space and the predicted pseudo labels constantly evolve during the training phase. Updating the pseudo labels and centers ensures the quality of the projected pseudo centers. It theory, it would be best to update the centers at each iteration. In practice, though, we update the pseudo labels using Eq. 3.8 and update the centers with the current feature extractor f using Eq. 3.6 for every specific interval I. A similar issue was

doi:10.6342/NTU202210189

addressed by [13], who propose maintaining a memory bank to update the estimated centers dynamically. In our experiments, we update the pseudo labels simultaneously. As maintaining a memory bank is time-consuming, we choose a relatively straightforward approach.

doi:10.6342/NTU202210189

Chapter 4 Experiments

We first describe the experimental setup, including the datasets, competing methods, and parameter settings in Section 4.1. We then present the experimental results to validate the superiority of the proposed SLA framework in Section 4.2. We further analyze the proposed framework and highlight limitations in Section 4.3.

4.1 Experimental Setup

4.1.1 Datasets

We evaluated the proposed SLA framework on two sets of SSDA benchmarks, including Office-Home [26] and DomainNet [17]. Office-Home is a mainstream benchmark for both UDA and SSDA that contains four domains: Art (A), Clipart (C), Product (P), and Real (R), with 65 categories. DomainNet was initially designed for benchmarking multi-source domain adaptation approaches. [20] picks four domains (Real (R), Clipart (C), Painting (P), and Sketch (S)) with 126 classes to build a cleaner dataset for SSDA, and focuses on seven scenarios instead of combining all pairs. Our experiments follow the settings of recent studies [11, 20, 29], with the same sampling strategy for both the training set and validation set, and we adopt both 1-shot and 3-shot settings on all datasets.

doi:10.6342/NTU202210189

4.1.2 Implementation

Our framework can be applied with many state-of-the-art methods. We applied it with MME [20] and CDAC [11] to validate the efficacy of our method; the corresponding methods are named MME + SLA and CDAC + SLA. For a fair comparison, we chose ResNet34 [7] as our backbone, which was pre-trained on the ImageNet-1K dataset [4], with the model architecture, batch size, learning rate scheduler, optimizer, weight decay, and initialization strategy all following previous work [11, 20, 29]. We used the hyperparameters recommended for MME and CDAC. We set the mix ratio α in Eq. 3.12 to 0.3 and the temperature parameter T in Eq. 3.7 to 0.6. The update interval I mentioned in Section 3.3 was 500. The warmup parameter W in Eq. 3.12 was 500 for MME on Office-Home, 2000 for CDAC on Office-Home, 3000 for MME on DomainNet, and 50000 for CDAC on DomainNet. After the warmup stage, we refreshed the learning rate scheduler so that the label adaptation loss would be updated at a higher learning rate. All hyperparameters were properly tuned via the validation process. For each subtask, we conducted the experiments three times. Detailed statistics concerning our results can be found in our supplementary materials.

4.2 Comparison with State-of-the-Art Methods

We compare our results with several baselines, including S+T, DANN [5], ENT [6], MME [20], APE [9], CDAC [11], DECOTA [30], MCL [29]. 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, which [20] reproduces by training with additional labeled target data. ENT is a standard entropy minimization orig-

doi:10.6342/NTU202210189

inally designed for semi-supervised learning, also reproduced by [20]. Note that for MCL, we only compare with their DomainNet results. We leave the detailed analysis for MCL on Office-Home to Section 4.3.

4.2.1 *DomainNet*

We show the results on the DomainNet dataset with 1-shot and 3-shot settings in Table A.3. Note first that for MME and CDAC, almost all sub-tasks show improvement after applying our SLA framework, except for two cases where CDAC + SLA performs roughly the same as CDAC. Second, note that the overall performance of CDAC + SLA for 1-shot and 3-shot settings reaches 75.0% and 76.9%, respectively; both outperform previous methods and achieve new state-of-the-art results.

doi:10.6342/NTU202210189

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	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	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	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	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	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	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	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 4.1: Accuracy (%) on DomainNet for 1-shot and 3-shot semi-supervised domain adaptation (ResNet34)

doi:10.6342/NTU202210189

4.2.2 *Office-Home*

We show the results on the Office-Home dataset with 1-shot and 3-shot settings in Table 4.2. Similarly, after applying SLA to MME and CDAC, the performance improves greatly except for one case under the 3-shot setting. Overall, our framework outperforms the original methods by at least 1.5% under all settings.

doi:10.6342/NTU202210189

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	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	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	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	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	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	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	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	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	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	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	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	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 4.2: Accuracy (%) on Office-Home for 1-shot and 3-shot semi-supervised domain adaptation (ResNet34)

doi:10.6342/NTU202210189

4.3 Analysis

4.3.1 MCL Reproducibility

MCL [29] uses consistency regularization for SSDA at three different levels and achieves excellent results. However, in our experiments we were unable to fully reproduce their reported numbers. The reproduced 3-shot Office-Home results are shown in Table A.5. After applying our SLA framework, although we stably improve the reproduction, we are still unable to compete with their reported values. We include our detailed reproduced results in the supplementary materials, and will make the code publicly available.

doi:10.6342/NTU202210189

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	**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 the authors

Table 4.3: Accuracy (%) of MCL and MCL + SLA on Office-Home for 3-shot semi-supervised domain adaptation (ResNet34)

doi:10.6342/NTU202210189

Method	A → P	C → A	P → A	R → C
S+T	74.7	56.3	58.1	59.1
S+T + PPC	77.1	59.8	60.9	62.1
S+T + SLA	**77.7**	**60.5**	**61.3**	**62.5**

Table 4.4: Accuracy (%) of S+T, S+T + PPC, and S+T + SLA on 3-shot Office-Home with ResNet34. Although directly applying PPC to S+T improves performance, we show that learning from the PPC-modified labels yields much better performance.

4.3.2 PPC for Inference

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 [12], in which the method is applied to boost their final performance. If PPC performs well, a natural question is this: *Is it necessary to first modify the source labels by PPC and then learn from these modified labels?* As shown in Table 4.4, S+T + SLA outperforms directly taking PPC for inference. This also confirms that we can do much better by carefully revisiting the usage of source data.

4.3.3 Illustration of Adapted Labels

As discussed in Section 3.1, we seek to adapt the original source label $\mathbf{y}_i^s$ to the ideal label $g^*(\mathbf{x}_i^s)$. In practice, PPC helps predict the adapted labels. To demonstrate the effectiveness of our framework, we implement S+T + SLA, predict the adapted labels by PPC over a particular class, and illustrate the average probability distribution of the adapted labels. The results are shown in Figure 4.1. Compared with the original source labels, which are one-hot-encoded, our adapted labels are much closer to the ideal labels.

doi:10.6342/NTU202210189

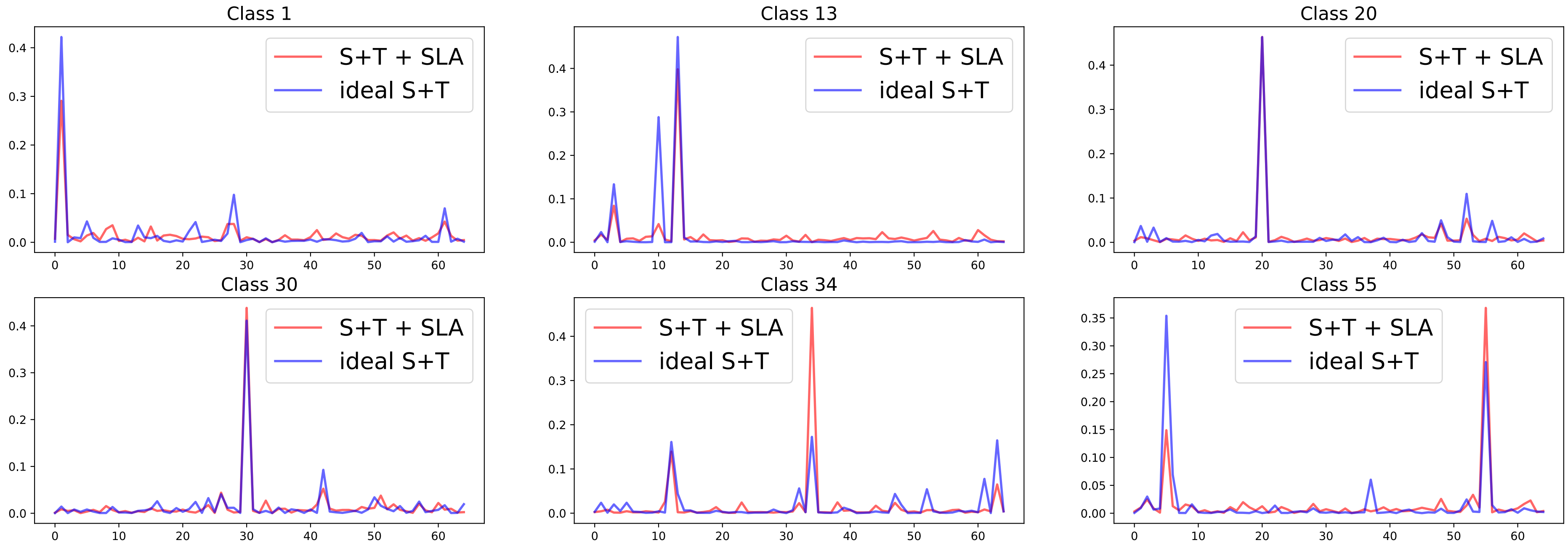

Figure 4.1: Average adapted source labels from PPC and ideal S+T for a certain class (3-shot Office-Home A $\rightarrow$ C with ResNet34). X-axis: the classes; Y-axis: the probability of the average adapted labels. We illustrate the average adapted source labels in S+T + SLA for six representative classes. Note that the original source labels should be one-hot encoded. The results show that the adapted labels can be much closer to the ideal labels.

doi:10.6342/NTU202210189

4.3.4 Warmup 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 treat the warmup strategy as a two-stage algorithm. Taking MME as our backbone method, the algorithm works in this fashion:

1. Train a model with normal MME loss for W iterations.
2. Take the model above as a pre-trained model and further apply 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. Empirically, however, we found that the performance of MME + SLA degrades if we train an MME model until it converges. Table 4.5 shows the sensitivity test of W of MME + SLA on Office-Home. We observe that regardless of the 1-shot or 3-shot setting, the performance generally worsens with the number of warmup stages. To better understand this effect, we first pre-trained a normal MME for W iterations, and then observed the label adaptation loss of MME + SLA. Figure 4.2 plots the label adaptation loss of MME + SLA when first pre-training MME for W iterations. We observe that when $W = 5000$, the initial label adaptation loss is already close to 0. Label adaptation in this situation is almost equivalent to doing nothing, as mentioned in Section 3.1.

doi:10.6342/NTU202210189

	A → 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 4.5: Accuracy (%) for various warmup stages W of MME + SLA on 3-shot Office-Home A → C with ResNet34

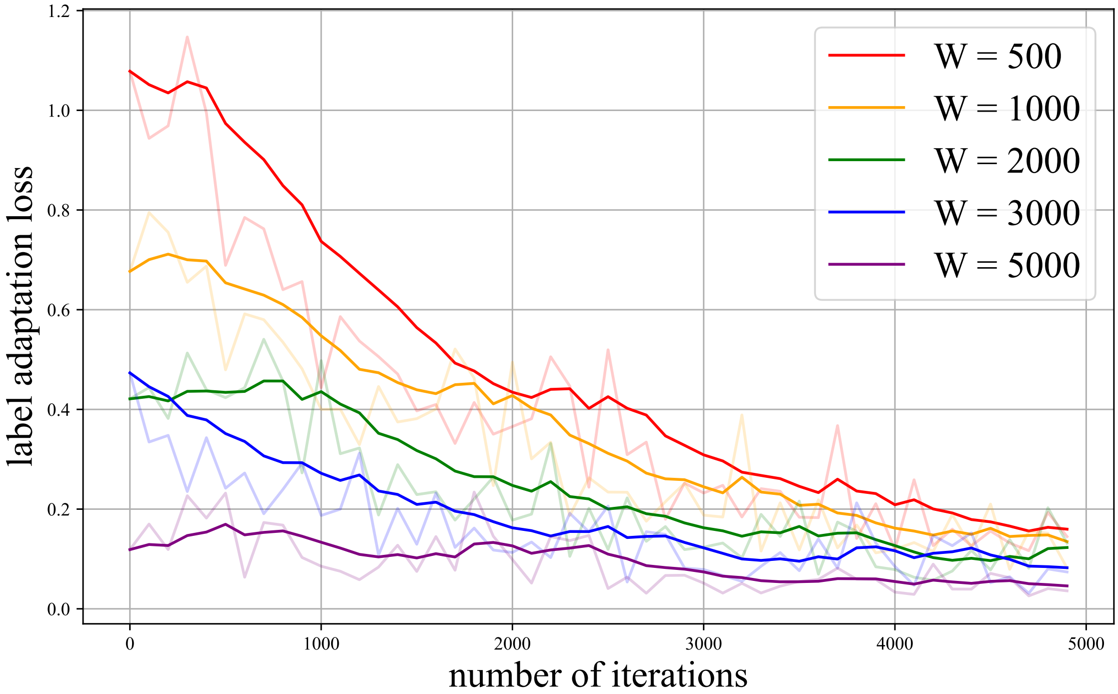

Figure 4.2: Label adaptation loss of MME + SLA by first pre-training MME for W iterations on 3-shot Office-Home A → C with ResNet34 (smoothing by EMA with a ratio of 0.8)

4.3.5 Limitations

The proposed SLA framework might not be helpful if the label adaptation loss approaches 0. Although we address this using protonet with pseudo centers, the loss converges to 0 in MME + SLA. We leave the analysis of the reason behind this convergence as future work. Nevertheless, we argue that it is unnecessary to discuss the reason in our proposed scope since we can strike a balance by carefully tuning the warmup parameter W, making this simply a problem of hyperparameter selection.

doi:10.6342/NTU202210189

Chapter 5 Conclusion

In this work, we present source label adaptation (SLA), a general framework for semi-supervised domain adaptation. Our work demonstrates that the usage of source data should be revisited carefully. We argue that from the perspective of the target data, the original source labels are often noisy. We thus approach domain adaptation as a noisy label learning problem and correct source labels with predictions from protonet with pseudo centers. Our approach primarily addresses an issue that is orthogonal to other existing works focused on improving the usage of unlabeled data. The empirical results show that when applied to state-of-the-art algorithms for SSDA, the proposed framework further improves their performance.

doi:10.6342/NTU202210189

References

[1] P. Bachman, O. Alsharif, and D. Precup. Learning with pseudo-ensembles. Advances in neural information processing systems, 27, 2014.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. Machine Learning, 79:151–175, 2010.

[3] D. M. Chan, R. Rao, F. Huang, and J. F. Canny. Gpu accelerated t-distributed stochastic neighbor embedding. Journal of Parallel and Distributed Computing, 131:1–13, 2019.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. The journal of machine learning research, 17(1):2096–2030, 2016.

[6] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. Advances in neural information processing systems, 17, 2004.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition.

doi:10.6342/NTU202210189

In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[8] G. Kang, L. Jiang, Y. Yang, and A. 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.

[9] T. Kim and C. 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.

[10] D.-H. 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.

[11] J. Li, G. Li, Y. Shi, and Y. 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.

[12] J. Liang, D. Hu, and J. 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.

[13] J. Liang, D. Hu, and J. 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.

[14] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. Advances in neural information processing systems, 29, 2016.

doi:10.6342/NTU202210189

[15] J. Na, H. Jung, H. J. Chang, and W. 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.

[16] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. 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.

[17] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In Proceedings of the IEEE International Conference on Computer Vision, pages 1406–1415, 2019.

[18] I. Redko, A. Habrard, and M. 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.

[19] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596, 2014.

[20] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised domain adaptation via minimax entropy. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8050–8058, 2019.

[21] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3723–3732, 2018.

doi:10.6342/NTU202210189

[22] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. Advances in neural information processing systems, 30, 2017.

[23] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems, 33:596–608, 2020.

[24] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080, 2014.

[25] D. Tanaka, D. Ikami, T. Yamasaki, and K. 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.

[26] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5018–5027, 2017.

[27] X. Wang, Y. Hua, E. Kodirov, S. S. Mukherjee, D. A. Clifton, and N. M. Robertson. Proselflc: Progressive self label correction towards a low-temperature entropy state. arXiv preprint arXiv:2207.00118, 2022.

[28] X. Xia, T. Liu, B. Han, N. Wang, M. Gong, H. Liu, G. Niu, D. Tao, and M. Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. Advances in Neural Information Processing Systems, 33:7597–7610, 2020.

[29] Z. Yan, Y. Wu, G. Li, Y. Qin, X. Han, and S. Cui. Multi-level consistency learning for semi-supervised domain adaptation. arXiv preprint arXiv:2205.04066, 2022.

doi:10.6342/NTU202210189

[30] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K. Q. Weinberger, W.-L. Chao, and S.-N. 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.

[31] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. Advances in neural information processing systems, 33:7260–7271, 2020.

[32] H.-J. Ye, H. Hu, D.-C. Zhan, and F. 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.

[33] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3):107–115, 2021.

[34] Y. Zhang, H. Zhang, B. Deng, S. Li, K. Jia, and L. Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. arXiv preprint arXiv:2106.00417, 2021.

[35] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon. On learning invariant representations for domain adaptation. In International Conference on Machine Learning, pages 7523–7532. PMLR, 2019.

[36] Y. Zhao, L. Cai, et al. Reducing the covariate shift by mirror samples in cross domain alignment. Advances in Neural Information Processing Systems, 34:9546–9558, 2021.

doi:10.6342/NTU202210189

[37] G. Zheng, A. H. Awadallah, and S. Dumais. Meta label correction for noisy label learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 11053–11061, 2021.

doi:10.6342/NTU202210189

Appendix A — Introduction

In this chapter, we provide our detailed implementation results. The link to the code to reproduce our main results on *Office-Home* and *DomainNet* datasets will be made publicly available.

A.1 Implementation Detail

Our proposed framework, Source Label Adaptation (SLA) invovles cooperation with other state-of-the-art algorithms. We take MME [20] and CDAC [11] as our backbone models, named MME + SLA and CDAC + SLA, respectively. For **MME + SLA**, we use the official implementation in https://github.com/VisionLearningGroup/SSDA_MME to obtain the MME loss. For **CDAC + SLA**, we use the official implementation in https://github.com/lijichang/CVPR2021-SSDA to obtain the CDAC loss. We follow the suggestions in both papers to select all hyper-parameters across different datasets.

A.2 Experiment Detail

For each sub-task on *DomainNet* and *Office-Home* datasets, we run three times with different seeds and take the average to obtain the value. This sections provides the average

doi:10.6342/NTU202210189

values and the standard deviations of our experiments. Table A.1 and Table A.2 shows the detailed statistics for one-shot and three-shot Semi-Supervised Domain Adaptation (SSDA) on *Office-Home* dataset, respectively. Table A.3 shows the statistics for one-shot and three-shot SSDA on *DomainNet* dataset.

A.3 Reproducibility Issue for MCL

MCL [29] is a state-of-the-art algorithm for SSDA, which performs consistency learning at three different levels and achieve great results. In our study, we also try to couple the MCL loss with our SLA framework. We follow the official implementation in `https://github.com/chester256/MCL` to reproduce the experiments. However, when reproducing the results on 3-shot *Office-Home* dataset. We found that it is generally hard to reach the reported numbers provided in their original paper. We address the issue by first reproducing MCL five times with different seeds using totally the same code in above. The detailed statistics are shown in Table A.4. We then run another three trials for MCL and MCL + SLA by fixing the seed for the generator in the DataLoader. This step is to compare the two approaches in a much more fair manner. The link to our modified code will also be made publicly available, and the results are shown in Table A.5. As we stated in the main paper, though after applying SLA, we can generally do better than our reproducing MCL, we are still not able to achieve the reported values in the original work.

doi:10.6342/NTU202210189

Stats	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**
MME + SLA													
avg.	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
std.	0.231	0.243	0.129	0.208	0.378	0.033	0.104	0.366	0.033	0.080	0.306	0.032	0.179
CDAC + SLA													
avg.	63.0	78.0	79.2	66.9	77.6	77.0	67.3	61.8	80.6	72.7	66.1	84.6	72.9
std.	0.431	0.873	0.133	0.111	0.653	0.200	0.404	0.324	0.066	0.489	0.270	0.117	0.339

Table A.1: Results on *Office-Home* dataset for 1-shot Semi-Supervised Domain Adaptation with ResNet34.

doi:10.6342/NTU202210189

Stats	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**
						MME + SLA							
avg.	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%
std.	0.119	0.135	0.082	0.279	0.033	0.286	0.084	0.085	0.060	0.329	0.115	0.179	0.149
						CDAC + SLA							
avg.	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%
std.	0.295	0.186	0.060	0.411	0.233	0.178	0.128	0.119	0.181	0.436	0.426	0.073	0.227

Table A.2: Results on *Office-Home* dataset for 3-shot Semi-Supervised Domain Adaptation with ResNet34.

doi:10.6342/NTU202210189

	R → C		R → P		P → C		C → S		S → P		R → S		P → R		**Mean**	
Stats	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**
MME + SLA																
avg.	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%
std.	0.217	0.231	0.082	0.135	0.244	0.207	0.361	0.238	0.129	0.097	0.148	0.083	0.213	0.203	0.199	0.171
CDAC + SLA																
avg.	79.8%	81.6%	75.6%	76.0%	77.4%	80.3%	68.1%	71.2%	71.7%	73.5%	71.7%	73.5%	80.4%	82.5%	75.0%	76.9%
std.	0.224	0.363	0.079	0.122	0.231	0.213	0.713	0.198	0.326	0.235	0.135	0.099	0.387	0.174	0.299	0.201

Table A.3: Results on *DomainNet* dataset for 1-shot and 3-shot Semi-Supervised Domain Adaptation with ResNet34.

doi:10.6342/NTU202210189

Stats	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**
avg.	63.5%	81.6%	80.7%	69.7%	82.4%	79.2%	70.6%	65.0%	82.7%	75.2%	67.8%	86.6%	75.4%
std.	0.678	0.647	0.476	0.648	1.033	0.506	0.311	0.823	0.151	0.269	0.847	0.301	0.558
min.	62.5%	80.7%	79.8%	68.9%	80.5%	78.3%	70.3%	63.8%	82.4%	74.8%	66.7%	86.3%	74.6%
max.	64.4%	82.4%	81.1%	70.7%	83.5%	79.7%	71.2%	66.3%	82.9%	75.5%	69.3%	87.1%	76.2%
reported	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%

Table A.4: The detailed statistics of our reproducing results for MCL on 3-shot *Office-Home* dataset with ResNet34. We reproduce MCL five times with different seeds. **reported**: The reported numbers provided in the original paper [29].

doi:10.6342/NTU202210189

Stats	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*							
avg.	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%
std.	0.237	0.345	0.318	0.678	0.830	0.730	0.073	0.106	0.212	0.147	0.321	0.440	0.370
						MCL + SLA							
avg.	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%
std.	0.380	0.090	0.250	0.551	0.900	0.489	0.077	0.114	0.000	0.090	0.261	0.332	0.295
reported	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

Table A.5: Results of MCL* and MCL + SLA with another 3 different seeds on 3-shot *Office-Home* dataset. *: Reproduced by ourselves. **reported**: The reported values in the original paper [29].

doi:10.6342/NTU202210189