Cycle Self-Training for Domain Adaptation

Hong Liu
Dept of Electronic Engineering
Tsinghua University
`hongliu9903@gmail.com`

Jianmin Wang
School of Software, BNRist
Tsinghua University
`jimwang@tsinghua.edu.cn`

Mingsheng Long[*]
School of Software, BNRist
Tsinghua University
`mingsheng@tsinghua.edu.cn`

Abstract

Mainstream approaches for unsupervised domain adaptation (UDA) learn domain-invariant representations to narrow the domain shift, which are empirically effective but theoretically challenged by the hardness or impossibility theorems. Recently, self-training has been gaining momentum in UDA, which exploits unlabeled target data by training with target pseudo-labels. However, as corroborated in this work, under distributional shift, the pseudo-labels can be unreliable in terms of their large discrepancy from target ground truth. In this paper, we propose *Cycle Self-Training* (CST), a principled self-training algorithm that explicitly enforces pseudo-labels to generalize across domains. CST cycles between a forward step and a reverse step until convergence. In the forward step, CST generates target pseudo-labels with a source-trained classifier. In the reverse step, CST trains a target classifier using target pseudo-labels, and then updates the shared representations to make the target classifier perform well on the source data. We introduce the Tsallis entropy as a confidence-friendly regularization to improve the quality of target pseudo-labels. We analyze CST theoretically under realistic assumptions, and provide hard cases where CST recovers target ground truth, while both invariant feature learning and vanilla self-training fail. Empirical results indicate that CST significantly improves over the state-of-the-arts on visual recognition and sentiment analysis benchmarks.

1 Introduction

Transferring knowledge from a source domain with rich supervision to an unlabeled target domain is an important yet challenging problem. Since deep neural networks are known to be sensitive to subtle change in underlying distributions [70], models trained on one labeled dataset often fail to generalize to another unlabeled dataset [58, 1]. Unsupervised domain adaptation (UDA) addresses the challenge of distributional shift by adapting the source model to the unlabeled target data [50, 43].

The mainstream paradigm for UDA is *feature adaptation*, a.k.a. *domain alignment*. By reducing the distance of the source and target feature distributions, these methods learn *invariant representations* to facilitate knowledge transfer between domains [34, 22, 36, 54, 37, 73], with successful applications in various areas such as computer vision [63, 27, 77] and natural language processing [75, 49]. Despite their popularity, the impossibility theories [6] uncovered intrinsic limitations of learning invariant representations when it comes to label shift [74, 32] and shift in the support of domains [29].

Recently, *self-training* (a.k.a. *pseudo-labeling*) [21, 78, 30, 32, 47, 68] has been gaining momentum as a promising alternative to feature adaptation. Originally tailored to semi-supervised learning, self-training generates pseudo-labels of unlabeled data, and jointly trains the model with source labels and target pseudo-labels [31, 39, 30]. However, the *distributional shift* in UDA makes pseudo-labeling more difficult. Directly using all pseudo-labels is risky due to accumulated error and even trivial solution [14]. Thus previous works tailor self-training to UDA by selecting trustworthy pseudo-labels. Using confidence threshold or reweighting, recent works try to alleviate the negative effect of domain

[*]Corresponding author: Mingsheng Long (`mingsheng@tsinghua.edu.cn`)

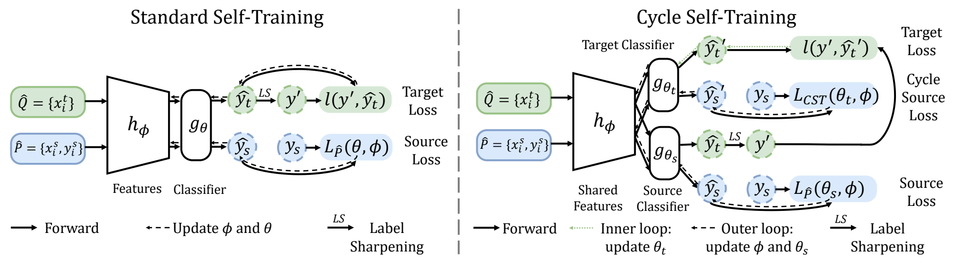


Figure 1: **Standard self-training vs. cycle self-training.** In standard self-training, we generate target pseudo-labels with a source model, and then train the model with both source ground-truths and target pseudo-labels. In cycle self-training, we train a target classifier with target pseudo-labels in the **inner loop**, and make the target classifier perform well on the source domain by updating the shared representations in the **outer loop**.

shift in standard self-training [78, 47], but they can be brittle and require expensive tweaking of the threshold or weight for different tasks, and their performance gain is still inconsistent.

In this work, we first analyze the quality of pseudo-labels with or without domain shift to delve deeper into the difficulty of standard self-training in UDA. On popular benchmark datasets, when the source and target are the same, our analysis indicates that the pseudo-label distribution is almost identical to the ground-truth distribution. However, with distributional shift, their discrepancy can be *very large* with examples of several classes mostly misclassified into other classes. We also study the difficulty of selecting correct pseudo-labels with popular criteria under domain shift. Although entropy and confidence are reasonable selection criteria for correct pseudo-labels without domain shift, the domain shift makes their accuracy decrease sharply.

Our analysis shows that domain shift makes pseudo-labels *unreliable* and that self-training on selected target instances with accurate pseudo-labels is less successful. Thereby, more principled improvement of standard self-training should be tailored to UDA and address the domain shift explicitly. In this work, we propose *Cycle Self-Training* (CST), a principled self-training approach to UDA, which overcomes the limitations of standard self-training (see Figure 1). Different from previous works to select target pseudo-labels with hard-to-tweak protocols, CST learns to generalize the pseudo-labels across domains. Specifically, CST *cycles* between the use of target pseudo-labels to train a target classifier, and the update of shared representations to make the target classifier perform well on the source data. In contrast to the standard Gibbs entropy that makes the target predictions over-confident, we propose a confidence-friendly uncertainty measure based on the *Tsallis entropy* in information theory, which adaptively minimizes the uncertainty without manually tuning or setting thresholds. Our method is simple and generally applicable to vision and language tasks with various backbones.

We empirically evaluate our method on a series of standard UDA benchmarks. Results indicate that CST outperforms previous state-of-the-art methods in 21 out of 25 tasks for object recognition and sentiment classification. Theoretically, we prove that the minimizer of CST objective is endowed with general guarantees of target performance. We also study hard cases on specific distributions, showing that CST recovers target ground-truths while both feature adaptation and standard self-training fail.

2 Preliminaries

We study unsupervised domain adaptation (UDA). Consider a source distribution P and a target distribution Q over the input-label space $\mathcal{X}\times\mathcal{Y}$. We have access to n_s labeled *i.i.d.* samples $\widehat{P}=\{x_i^s,y_i^s\}_{i=1}^{n_s}$ from P and n_t unlabeled *i.i.d.* samples $\widehat{Q}=\{x_i^t\}_{i=1}^{n_t}$ from Q. The model f comprises a feature extractor h_ϕ parametrized by ϕ and a head (linear classifier) g_θ parametrized by θ, i.e. $f_{\theta,\phi}(x)=g_\theta(h_\phi(x))$. The loss function is $\ell(\cdot,\cdot)$. Denote by $L_P(\theta,\phi):=\mathbb{E}_{(x,y)\sim P}\ell(f_{\theta,\phi}(x),y)$ the expected error on P. Similarly, we use $L_{\widehat{P}}(\theta,\phi)$ to denote the empirical error on dataset $\widehat{P}$.

We discuss two mainstream UDA methods and their formulations: feature adaptation and self-training.

Feature Adaptation trains the model f on the source dataset $\widehat{P}$, and simultaneously matches the source and target distributions in the representation space $\mathcal{Z}=h(\mathcal{X})$:

$$\min_{\theta,\phi} L_{\widehat{P}}(\theta,\phi)+d(h_\sharp\widehat{P},h_\sharp\widehat{Q}). \tag{1}$$

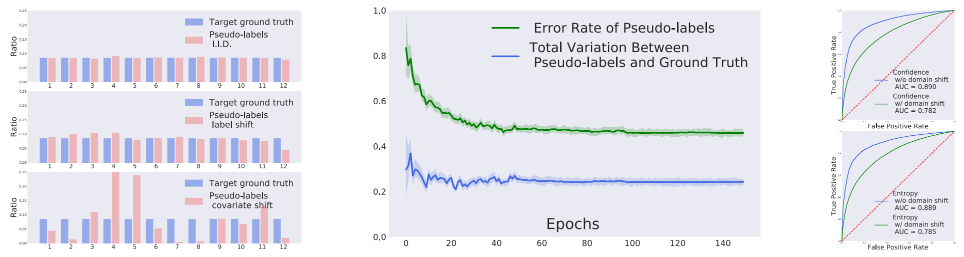


Figure 2: **Analysis of pseudo-labels under domain shift on VisDA-2017.** Left: Pseudo-label distributions with and without domain shift. Middle: Changes of pseudo-label distributions throughout training. Right: Quality of pseudo-labels under different pseudo-label selection criteria.

Here, $h_\sharp \widehat{P}$ denotes the pushforward distribution of $\widehat{P}$, and $d(\cdot,\cdot)$ is some distribution distance. For instance, Long et al. [34] used maximum mean discrepancy d_{MMD}, and Ganin et al. [22] approximated the $\mathcal{H}\Delta\mathcal{H}$-distance $d_{\mathcal{H}\Delta\mathcal{H}}$ [7] with adversarial training. Despite its pervasiveness, recent works have shown the intrinsic limitations of feature adaptation under real-world situations [6, 74, 33, 32, 29].

Self-Training is considered a promising alternative to feature adaptation. In this work we mainly focus on pseudo-labeling [31, 30]. Stemming from semi-supervised learning, standard self-training trains a source model f_s on the source dataset $\widehat{P}$: $\min_{\theta_s,\phi_s} L_{\widehat{P}}(\theta_s, \phi_s)$. The *target pseudo-labels* are then generated by f_s on the target dataset $\widehat{Q}$. To leverage unlabeled target data, self-training trains the model on the source and target datasets together with source ground-truths and target pseudo-labels:

$$\min_{\theta,\phi} L_{\widehat{P}}(\theta,\phi) + \mathbb{E}_{x\sim\widehat{Q}} \ell(f_{\theta,\phi}(x), \arg\max_i \{f_{\theta_s,\phi_s}(x)_{[i]}\}). \tag{2}$$

Self-training also uses label-sharpening as a standard protocol [31, 57]. Another popular variant of pseudo-labeling is the teacher-student model [4, 61], which iteratively improves the quality of pseudo-labels via alternatively replacing θ_s and ϕ_s with θ and ϕ of the previous iteration.

2.1 Limitations of Standard Self-Training

Standard self-training with pseudo-labels uses unlabeled data efficiently for semi-supervised learning [31, 39, 57]. Here we carry out exploratory studies on the popular VisDA-2017 [45] dataset using ResNet-50 backbones. We find that domain shift makes the pseudo-labels biased towards several classes and thereby unreliable in UDA. See Appendix C.1 for details and results on more datasets.

Pseudo-label distributions with or without domain shift. We resample the original VisDA-2017 to simulate different relationship between source and target domains: 1) *i.i.d.*, 2) covariate shift, and 3) label shift. We train the model on the three variants of source dataset and use it to generate target pseudo-labels. We show the distributions of target ground-truths and pseudo-labels in Figure 2 (Left). When the source and target distributions are identical, the distribution of pseudo-labels is almost the same as ground-truths, indicating the reliability of pseudo-labels. In contrast, when exposed to label shift or covariate shift, the distribution of pseudo-labels is significantly different from target ground-truths. Note that classes 2, 7, 8 and 12 appear rarely in the target pseudo-labels in the covariate shift setting, indicating that the pseudo-labels are biased towards several classes due to domain shift. Self-training with these pseudo-labels is risky since it may lead to misalignment of distributions and misclassify many examples of classes 2, 7, 8 and 12.

Change of pseudo-label distributions throughout training. To further study the change of pseudo-labels in standard self-training, we compute the total variation (TV) distance between target ground-truths and target pseudo-labels: $d_{\text{TV}}(c, c') = \frac{1}{2}\sum_i \|c_i - c'_i\|$, where c_i is the ratio of class i. We plot its change during training in Figure 2 (Middle). Although the error rate of pseudo-labels continues to decrease, d_{TV} remains almost unchanged at 0.26 throughout training. Note that d_{TV} is the lower bound of the error rate of the pseudo-labels (shown in Appendix C.1). If d_{TV} converges to 0.26, then the accuracy of pseudo-labels is upper-bounded by 0.74. This indicates that the important denoising ability [66] of pseudo-labels in standard self-training is hindered by domain shift.

Difficulty of selecting reliable pseudo-labels under domain shift. To mitigate the negative effect of false pseudo-labels, recent works proposed to select correct pseudo-labels based on thresholding the entropy or confidence criteria [35, 21, 37, 57]. However, it remains unclear whether these strategies are still effective under domain shift. Here we compare the quality of pseudo-labels selected by different strategies with or without domain shift. For each strategy, we compute False Positive Rate and True Positive Rate for different thresholds and plot its ROC curve in Figure 2 (Right). When the source and target distributions are identical, both entropy and confidence are reasonable strategies for selecting correct pseudo-labels (AUC=0.89). However, when the target pseudo-labels are generated by the source model, the quality of pseudo-labels decreases sharply under domain shift (AUC=0.78).

3 Approach

We present Cycle Self-Training (CST) to improve pseudo-labels under domain shift. An overview of our method is given in Figure 1. Cycle Self-Training iterates between a *forward step* and a *reverse step* to make self-trained classifiers generalize well on both target and source domains.

3.1 Cycle Self-Training

Forward Step. Similar to standard self-training, we have a source classifier θ_s trained on top of the shared representations ϕ on the labeled source domain, and use it to generate target pseudo-labels as

$$y' = \arg\max_i \{f_{\theta_s,\phi}(x)_{[i]}\}, \tag{3}$$

for each x in the target dataset $\widehat{Q}$. Traditional self-training methods use confidence thresholding or reweighting to select reliable pseudo-labels. For example, Sohn et al. [57] select pseudo-labels with softmax value and Long et al. [37] add entropy reweighting to rely on examples with more confidence prediction. However, the output of deep networks is usually miscalibrated [25], and is not necessarily related to the ground-truth confidence even on the same distribution. In domain adaptation, as shown in Section 2.1, the discrepancy between the source and target domains makes pseudo-labels even more unreliable, and the performance of commonly used selection strategies is also unsatisfactory. Another drawback is the expensive tweaking in order to find the optimal confidence threshold for new tasks. To better apply self-training to domain adaptation, we expect that the model can gradually refine the pseudo-labels by itself without the cumbersome selection or thresholding.

Reverse Step. We design a complementary step with the following insights to improve self-training. Intuitively, the labels on the source domain contain both useful information that can transfer to the target domain and harmful information that can make pseudo-labels incorrect. Similarly, *reliable pseudo-labels* on the target domain can transfer to the source domain in turn, while models trained with incorrect pseudo-labels on the target domain cannot transfer to the source domain. In this sense, if we explicitly train the model to make target pseudo-labels informative of the source domain, we can gradually make the pseudo-labels more accurate and learn to generalize to the target domain.

Specifically, with the pseudo-labels y' generated by the source classifier θ_s at hand as in equation 3, we train a target head $\hat{\theta}_t(\phi)$ on top of the representation ϕ with pseudo-labels on the target domain $\widehat{Q}$,

$$\hat{\theta}_t(\phi) = \arg\min_\theta \mathbb{E}_{x\sim\widehat{Q}}\ell(f_{\theta,\phi}(x), y'). \tag{4}$$

We wish to make the target pseudo-labels informative of the source domain and gradually refine them. To this end, we update the shared feature extractor ϕ to predict accurately on the source domain and jointly *enforce the target classifier $\hat{\theta}_t(\phi)$ to perform well on the source domain*. This naturally leads to the objective of **Cycle Self-Training**:

$$\underset{\theta_s,\phi}{\text{minimize}}\, L_{\text{Cycle}}(\theta_s,\phi) := L_{\widehat{P}}(\theta_s,\phi) + L_{\widehat{P}}(\hat{\theta}_t(\phi),\phi). \tag{5}$$

Bi-level Optimization. The objective in equation 5 relies on the solution $\hat{\theta}_t(\phi)$ to the objective in equation 4. Thus, CST formulates a *bi-level* optimization problem. In the **inner loop** we generate target pseudo-labels with the source classifier (equation 3), and train a target classifier with target pseudo-labels (equation 4). After each inner loop, we update the feature extractor ϕ for one step in the **outer loop** (equation 5), and start a new inner loop again. However, since the inner loop of the optimization in equation 4 only involves the light-weight linear head θ_t, we propose to calculate the analytical form of $\hat{\theta}_t(\phi)$ and directly back-propagate to the feature extractor ϕ instead of calculating the second-order derivatives as in MAML [18]. The resulting framework is as fast as training two heads jointly. Also note that the solution $\hat{\theta}_t(\phi)$ relies on θ_s implicitly through y'. However, both standard self-training and our implementation use label sharpening, making y' not differentiable. Thus we follow vanilla self-training and *do not* consider the gradient of $\hat{\theta}_t(\phi)$ w.r.t. y' in the outer loop optimization. We defer the derivation and implementation of bi-level optimization to Appendix B.2.

3.2 Tsallis Entropy Minimization

Gibbs entropy is widely used by existing semi-supervised learning methods to regularize the model output and minimize the uncertainty of predictions on unlabeled data [24]. In this work, we generalize Gibbs entropy to Tsallis entropy [62] in information theory. Suppose the softmax output of a model is $y \in \mathbb{R}^K$, then the α-*Tsallis entropy* is defined as

$$S_\alpha(y) = \frac{1}{\alpha - 1}\left(1 - \sum y_{[i]}^\alpha\right), \tag{6}$$

where $\alpha > 0$ is the *entropic-index*. Note that $\lim_{\alpha\to 1} S_\alpha(y) = \sum_i -y_{[i]}\log(y_{[i]})$ which exactly recovers the Gibbs entropy. When $\alpha = 2$, $S_\alpha(y)$ becomes the Gini impurity $1 - \sum_i y_{[i]}^2$.

We propose to control the uncertainty of target pseudo-labels based on **Tsallis entropy minimization**:

$$L_{\widehat{Q},\text{Tsallis},\alpha}(\theta, \phi) := \mathbb{E}_{x\sim\widehat{Q}} S_\alpha(f_{\theta,\phi}(x)). \tag{9}$$

Figure 3 shows the change of Tsallis entropy with different entropic-indices α for binary problems. Intuitively, smaller α exerts more penalization on uncertain predictions and larger α allows several scores y_i's to be similar. This is critical in self-training since an overly small α (as in Gibbs entropy) will make the incorrect dimension of pseudo-labels close to 1 and have no chance to be corrected throughout training. In Section 5.4, we further verify this property with experiments.

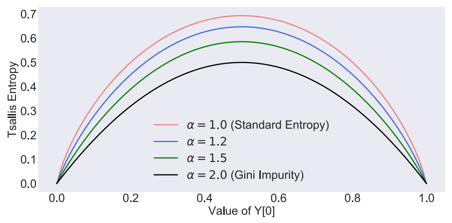


Figure 3: Tsallis entropy vs. entropic-index α.

An important improvement of the Tsallis entropy over Gibbs entropy is that it can choose the suitable measure of uncertainty for different systems to avoid over-confidence caused by overly penalizing the uncertain pseudo-labels. To automatically find the suitable α, we adopt a similar strategy as Section 3.1. The intuition is that if we use the suitable entropic-index α to train the source classifier $\theta_{s,\alpha}$, the target pseudo-labels generated by $\theta_{s,\alpha}$ will contain desirable knowledge of the source dataset, i.e. a target classifier $\theta_{t,\alpha}$ trained with these pseudo-labels will perform well on the source domain. Therefore, we semi-supervisedly train a classifier $\hat{\theta}_{s,\alpha}$ on the source domain with the α-Tsallis entropy regularization $L_{\widehat{Q},\text{Tsallis},\alpha}$ on the target domain as: $\hat{\theta}_{s,\alpha} = \arg\min_\theta L_{\widehat{P}}(\theta,\phi) + L_{\widehat{Q},\text{Tsallis},\alpha}(\theta,\phi)$, from which we obtain the target pseudo-labels. Then we train another head $\hat{\theta}_{t,\alpha}$ with target pseudo-labels. We automatically find α by minimizing the loss of $\hat{\theta}_{t,\alpha}$ on the source data:

$$\hat{\alpha} = \arg\min_{\alpha\in[1,2]} L_{\widehat{P}}(\hat{\theta}_{t,\alpha}, \phi) \tag{10}$$

To solve equation 10, we discretize the feasible region $[1, 2]$ of α and use discrete optimization to lower computational cost. We also update α at the start of each epoch, since we found more frequent

Algorithm 1 Cycle Self-Training (CST)

```
1: Input: source dataset $\widehat{P}$ and target dataset $\widehat{Q}$.
2: for epoch $= 0$ to MaxEpoch do
3:     Select $\hat{\alpha}$ as equation 10 at the start of each epoch.
4:     for $t = 0$ to MaxIter do
5:         Forward Step
6:         Generate pseudo-labels on the target domain with $\phi$ and $\theta_s$: $y' = \arg\max_i \{f_{\theta_s,\phi}(x)_{[i]}\}$.
7:         Reverse Step
8:         Train a target head $\hat{\theta}_t(\phi)$ with target pseudo-labels $y'$ on the feature extractor $\phi$:
               $\hat{\theta}_t(\phi) = \arg\min_\theta \mathbb{E}_{x\sim\widehat{Q}} \ell(f_{\theta,\phi}(x), y')$.
9:         Update the feature extractor $\phi$ and the source head $\theta_s$ to make $\hat{\theta}_t(\phi)$ perform well on the
           source dataset and minimize the $\hat{\alpha}$-Tsallis entropy on the target dataset:
               $\phi \leftarrow \phi - \eta\nabla_\phi[L_{\widehat{P}}(\theta_s,\phi) + L_{\widehat{P}}(\hat{\theta}_t(\phi),\phi) + L_{\widehat{Q},\text{Tsallis},\hat{\alpha}}(\theta_s,\phi)]$.   (7)
               $\theta_s \leftarrow \theta_s - \eta\nabla_{\theta_s}[L_{\widehat{P}}(\theta_s,\phi) + L_{\widehat{Q},\text{Tsallis},\hat{\alpha}}(\theta_s,\phi)]$.   (8)
10:    end for
11: end for
```

update leads to no performance gain. Details are deferred to Appendix B.3. Finally, with the optimal $\hat{\alpha}$ found, we add the $\hat{\alpha}$-Tsallis entropy minimization term $L_{\widehat{Q},\text{Tsallis},\hat{\alpha}}$ to the overall objective:

$$\underset{\theta_s,\phi}{\text{minimize}}\, L_{\text{Cycle}}(\theta_s,\phi) + L_{\widehat{Q},\text{Tsallis},\hat{\alpha}}(\theta_s,\phi). \tag{11}$$

In summary, Algorithm 1 depicts the complete training procedure of Cycle Self-Training (CST).

4 Theoretical Analysis

We analyze the properties of CST theoretically. First, we prove that the minimizer of the CST loss $L_{\text{CST}}(f_s, f_t)$ will lead to small target loss $\text{Err}_Q(f_s)$ under a simple but realistic expansion assumption. Then, we further demonstrate a concrete instantiation where cycle self-training provably recovers the target ground truth, but both feature adaptation and standard self-training fail. *Due to space limit, we state the main results here and defer all proof details to Appendix A.*

4.1 CST Provably Works under the Expansion Assumption

We start from a K-way classification model, $f : \mathcal{X} \to [0,1]^K \in \mathcal{F}$ and $\tilde{f}(x) := \arg\max_i f(x)_{[i]}$ denotes the prediction. Denote by P_i the conditional distribution of P given $y = i$. Assume the supports of P_i and P_j are disjoint for $i \neq j$. The definition is similar for Q_i. We further Assume $P(y = i) = Q(y = i)$. For any $x \in \mathcal{X}$, $\mathcal{N}(x)$ is defined as the *neighboring set* of x with a proper metric $d(\cdot,\cdot)$, $\mathcal{N}(x) = \{x' : d(x,x') \leq \xi\}$. $\mathcal{N}(A) := \cup_{x\in A}\mathcal{N}(x)$. Denote the expected error on the target domain by $\text{Err}_Q(f) := \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\tilde{f}(x) \neq y)$.

We study the CST algorithm under the *expansion assumption* of the mixture distribution [66, 11]. Intuitively, this assumption indicates that the conditional distributions P_i and Q_i are closely located and regularly shaped, enabling knowledge transfer from the source domain to the target domain.

Definition 1 (**(q,ϵ)-constant expansion** [66]). *We say P and Q satisfy (q,ϵ)-constant expansion for some constant $q, \epsilon \in (0,1)$, if for any set $A \in \mathcal{X}$ and any $i \in [K]$ with $\frac{1}{2} > P_{\frac{1}{2}(P_i+Q_i)}(A) > q$, we have $P_{\frac{1}{2}(P_i+Q_i)}(\mathcal{N}(A)\backslash A) > \min\{\epsilon, P_{\frac{1}{2}(P_i+Q_i)}(A)\}$.*

Based on this expansion assumption, we consider a *robustness-constrained* version of CST. Later we will show that the robustness is closely related to the uncertainty. Denote by f_s the source model and f_t the model trained on the target with pseudo-labels. Let $R(f_t) := P_{\frac{1}{2}(P+Q)}(\{x : \exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\})$ represent the robustness [66] of f_t on P and Q. Suppose $\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\tilde{f}_s(x) \neq \tilde{f}_t(x)) \leq c$ and $R(f_t) \leq \rho$. The following theorem states that when f_s and f_t behave similarly on the target domain Q and f_t is robust to local changes in input, the minimizer of the cycle source error $\text{Err}_P(f_t)$ will guarantee low error of f_s on the target domain Q.

Theorem 1. *Suppose Definition 1 holds for P and Q. For any f_s, f_t satisfying $\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\tilde{f}_s(x) \neq \tilde{f}_t(x)) \leq c$ and $R(f_t) \leq \rho$, the expected error of f_s on the target domain Q is bounded,*

$$\text{Err}_Q(f_s) \leq \text{Err}_P(f_t) + c + 2q + \frac{\rho}{\min\{\epsilon, q\}}. \tag{12}$$

To further relate the expected error with the CST training objective and obtain finite-sample guarantee, we use the multi-class margin loss: $l_\gamma(f(x), y) := \psi_\gamma(-\mathcal{M}(f(x), y))$, where $\mathcal{M}(v, y) = v_{[y]} - \max_{y'\neq y} v_{[y']}$ and ψ_γ is the ramp function. We then extend the margin loss: $\mathcal{M}(v) = \max_y(v_{[y]} - \max_{y'\neq y} v_{[y']})$ (The difference between the largest and the second largest scores in v), and $l_\gamma(f_t(x), f_s(x)) := \psi_\gamma(-\mathcal{M}(f_t(x), \tilde{f}_s(x)))$. Further suppose $f_{[i]}$ is L_f-Lipschitz w.r.t. the metric $d(\cdot,\cdot)$ and $\tau := 1 - 2L_f\xi\min\{\epsilon, q\} > 0$. Consider the following training objective for CST, denoted by $L_{\text{CST}}(f_s, f_t)$, where $L_{\widehat{P},\gamma}(f_t) := \mathbb{E}_{(x,y)\sim\widehat{P}} l_\gamma(f_t(x), y)$ corresponds to the cycle source loss in equation 5, $L_{\widehat{Q},\gamma}(f_t, f_s) := \mathbb{E}_{(x,y)\sim\widehat{Q}} l_\gamma(f_t(x), f_s(x))$ is consistent with the target loss in equation 4, and $\mathcal{M}(f_t(x))$ is closely related to the uncertainty of predictions in equation 11.

$$\min L_{\text{CST}}(f_s, f_t) := L_{\widehat{P},\gamma}(f_t) + L_{\widehat{Q},\gamma}(f_t, f_s) + \frac{1 - \mathbb{E}_{(x,y)\sim\frac{1}{2}(\widehat{P}+\widehat{Q})}\mathcal{M}(f_t(x))}{\tau}. \tag{13}$$

The following theorem shows that the minimizer of the training objective $L_{\text{CST}}(f_s, f_t)$ guarantees low population error of f_s on the target domain Q.

Theorem 2. *$\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}})$ denotes the empirical Rademacher complexity of function class $\mathcal{F}$ on dataset $\widehat{P}$. For any solution of equation 13 and $\gamma > 0$, with probability larger than $1-\delta$,*

$$\mathrm{Err}_Q(f_s) \le L_{\mathrm{CST}}(f_s, f_t) + 2q + \frac{4K}{\gamma}\left[\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}}) + \widehat{\mathcal{R}}(\tilde{\mathcal{F}} \times \mathcal{F}|_{\widehat{Q}})\right] + \frac{2}{\tau}\left[\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}}) + \widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{Q}})\right] + \zeta,$$

where $\zeta = O\left(\sqrt{\log(1/\delta)/n_s} + \sqrt{\log(1/\delta)/n_t}\right)$ is a low-order term. $\tilde{\mathcal{F}} \times \mathcal{F}$ refers to the function class $\{x \to f(x)_{[\tilde{f}'(x)]} : f, f' \in \mathcal{F}\}$.

Main insights. Theorem 2 justifies CST under the expansion assumption. The generalization error of the classifier f_s on the target domain is bounded with the CST loss objective $L_{\mathrm{CST}}(f_s, f_t)$, the intrinsic property of the data distribution q, and the complexity of the function classes. In our algorithm, $L_{\mathrm{CST}}(f_s, f_t)$ is minimized by the neural networks and q is a constant. The complexity of the function class can be controlled with proper regularization.

4.2 Hard Case for Feature Adaptation and Standard Self-Training

To gain more insight, we study UDA in a quadratic neural network $f_{\theta,\phi}(x) = \theta^\top(\phi^\top x)^{\odot 2}$, where $\odot$ is element-wise power. In UDA, the source can have *multiple solutions* but we aim to learn the one working on the target [34]. We design the underlying distributions p and q in Table 6 to reflect this. Consider the following P and Q. $x_{[1]}$ and $x_{[2]}$ are sampled *i.i.d.* from distribution p on P, and from q on Q. For $i \in [3, d]$, $x_{[i]} = \sigma_i x_{[2]}$ on P and $x_{[i]} = \sigma_i x_{[1]}$ on Q. $\sigma_i \in \{\pm 1\}$ are *i.i.d.* and uniform. We also assume realizability: $y = x_{[1]}^2 - x_{[2]}^2$ for both source and target. Note that $y = x_{[1]}^2 - x_{[i]}^2$ for all $i \in [2, d]$ are solutions to P but *only* $y = x_{[1]}^2 - x_{[2]}^2$ works on Q. We visualize this specialized setting in Figure 4.

Table 1: The design of p and q.

Distribution	-1	$+1$	0
Source p	0.05	0.05	0.90
Target q	0.25	0.25	0.50

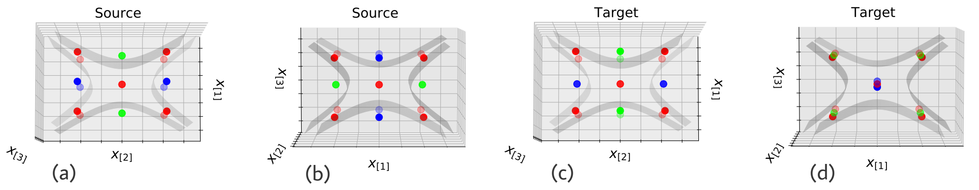


Figure 4: **The hard case where $d = 3$.** Green dots for $y = 1$, red dots for $y = 0$, and blue dots for $y = -1$. The grey curve is the classification boundary of different features. The good feature $x_{[1]}^2 - x_{[2]}^2$ works on the target domain (shown in (a) and (c)), whereas the spurious feature $x_{[1]}^2 - x_{[3]}^2$ only works on the source domain (shown in (b) and (d)). In Section 4.2, we show that feature adaptation and standard self-training learn $x_{[1]}^2 - x_{[3]}^2$, while CST learns $x_{[1]}^2 - x_{[2]}^2$.

To make the features more tractable, we study the norm-constrained version of the algorithms (details are deferred to Section A.3.2). We compare the features learned by feature adaptation, standard self-training, and CST. Intuitively, feature adaptation fails because the *ideal* target solution $y = x_{[1]}^2 - x_{[2]}^2$ has larger distance in the feature space than other spurious solutions $y = x_{[1]}^2 - x_{[i]}^2$. Standard self-training also fails since it will choose randomly among all solutions. In comparison, CST can recover the ground truth, because it can distinguish the spurious solution resulting in *bad pseudo-labels*. A classifier trained with those pseudo-labels *cannot* work on the source domain in turn. This intuition is rigorously justified in the following two theorems.

Theorem 3. *For $\epsilon \in (0, 0.5)$, the following statements hold for feature adaptation and self-training:*

- *For failure rate $\xi > 0$, and target dataset size $n_t > \Theta(\log \frac{1}{\xi})$, with probability at least $1 - \xi$ over the sampling of target data, the solution $(\hat{\theta}_{\mathrm{FA}}, \hat{\phi}_{\mathrm{FA}})$ found by feature adaptation satisfies*

$$\mathrm{Err}_Q(\hat{\theta}_{\mathrm{FA}}, \hat{\phi}_{\mathrm{FA}}) \ge \epsilon. \tag{14}$$

- *With probability at least $1 - \frac{1}{d-1}$, the solution $(\hat{\theta}_{\mathrm{ST}}, \hat{\phi}_{\mathrm{ST}})$ of standard self-training satisfies*

$$\mathrm{Err}_Q(\hat{\theta}_{\mathrm{ST}}, \hat{\phi}_{\mathrm{ST}}) \ge \epsilon. \tag{15}$$

Theorem 4. *For failure rate $\xi > 0$, and target dataset size $n_t > \Theta(\log \frac{1}{\xi})$, with probability at least $1-\xi$, the solution of CST $(\hat{\phi}_{\text{CST}}, \hat{\theta}_{\text{CST}})$ recovers the ground truth of the target dataset:*

$$\text{Err}_Q(\hat{\theta}_{\text{CST}}, \hat{\phi}_{\text{CST}}) = 0. \tag{16}$$

5 Experiments

We test the performance of the proposed method on both vision and language datasets. Cycle Self-Training (CST) consistently outperforms state-of-the-art feature adaptation and self-training methods. Code is available at `https://github.com/Liuhong99/CST`.

5.1 Setup

Datasets. We experiment on visual object recognition and linguistic sentiment classification tasks: *Office-Home* [64] has 65 classes from four kinds of environment with large domain gap: *Artistic* (**Ar**), *Clip Art* (**Cl**), *Product* (**Pr**), and *Real-World* (**Rw**); *VisDA-2017* [45] is a large-scale UDA dataset with two domains named **Synthetic** and **Real**. The datasets consist of over 200k images from 12 categories of objects; *Amazon Review* [10] is a linguistic sentiment classification dataset of product reviews in four products: *Books* (**B**), *DVDs* (**D**), *Electronics* (**E**), and *Kitchen* (**K**).

Implementation. We use **ResNet-50** [26] (pretrained on ImageNet [53]) as feature extractors for vision tasks, and **BERT** [16] for linguistic tasks. On VisDA-2017, we also provide results of ResNet-101 to include more baselines. We use cross-entropy loss for classification on the source domain. When training the target head $\hat{\theta}_t$ and updating the feature extractor with CST, we use squared loss to get the analytical solution of $\hat{\theta}_t$ directly and avoid calculating second order derivatives as meta-learning [18]. Details on adapting squared loss to multi-class classification are deferred to Appendix B. We adopt SGD with initial learning rate $\eta_0 = 2e-3$ for image classification and $\eta_0 = 5e-4$ for sentiment classification. Following standard protocol in [26], we decay the learning rate by 0.1 each 50 epochs until 150 epochs. We run all the tasks 3 times and report mean and deviation in top-1 accuracy. For VisDA-2017, we report the mean class accuracy. Following Theorem 2, we also enhance CST with sharpness-aware regularization [19] (CST+SAM), which help regularize the Lipschitzness of the function class. Due to space limit, we report mean accuracies in Tables 2 and 3 and defer standard deviation to Appendix C.

5.2 Baselines

We compare with two lines of works in domain adaptation: feature adaptation and self-training. We also compare with more complex state-of-the-arts and create stronger baselines by combining feature adaptation and self-training.

Feature Adaptation: DANN [22], MCD [54], CDAN [37] (which improves DANN with pseudo-label conditioning), MDD [73] (which improves previous domain adaptation with margin theory), Implicit Alignment (IA) [28] (which improves MDD to deal with label shift).

Self-Training. We include VAT [40], MixMatch [8] and FixMatch [57] in the semi-supervised learning literature as self-training methods. We also compare with self-training methods for UDA: CBST [77], which considers class imbalance in standard self-training, and KLD [78], which improves CBST with label regularization. However, these methods involve tricks specified for convolutional networks. Thus, in sentiment classification tasks where we use BERT backbones, we compare with other consistency regularization baselines: VAT [40], VAT+Entropy Minimization.

Feature Adaptation + Self-Training. DIRT-T [56] combines DANN, VAT, and entropy minimization. We also create more powerful baselines: CDAN+VAT+Entropy and MDD+Fixmatch.

Other SOTA. AFN [69] boosts transferability by large norm. STAR [38] aligns domains with stochastic classifiers. SENTRY [48] selects confident examples with a committee of random augmentations.

5.3 Results

Results on 12 pairs of *Office-Home* tasks are shown in Table 2. When domain shift is large, standard self-training methods such as VAT and FixMatch suffer from the decay in pseudo-label quality. **CST** outperforms feature adaptation and self-training methods significantly in 9 out of 12 tasks. Note that CST does not involve manually setting confidence threshold or reweighting.

Table 2: Accuracy (%) on Office-Home for unsupervised domain adaptation (`ResNet-50`).

Method	Ar-Cl	Ar-Pr	Ar-Rw	Cl-Ar	Cl-Pr	Cl-Rw	Pr-Ar	Pr-Cl	Pr-Rw	Rw-Ar	Rw-Cl	Rw-Pr	Avg.
DANN [22]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN [37]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
CDAN+VAT+Entropy	52.2	71.5	76.4	61.1	70.3	67.8	59.5	54.4	78.6	73.2	59.0	82.7	67.3
FixMatch [57]	51.8	74.2	80.1	63.5	73.8	61.3	64.7	51.4	80.0	73.3	56.8	81.7	67.7
MDD [73]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
MDD+IA [28]	56.2	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
SENTRY [48]	**61.8**	77.4	80.1	66.3	71.6	74.7	66.8	**63.0**	80.9	74.0	**66.3**	84.1	72.2
CST	59.0	**79.6**	**83.4**	**68.4**	**77.1**	**76.7**	**68.9**	56.4	**83.0**	**75.3**	62.2	**85.1**	**73.0**

Table 3: Accuracy (%) on Multi-Domain Sentiment Dataset for domain adaptation with `BERT`.

Method	B-D	B-E	B-K	D-B	D-E	D-K	E-B	E-D	E-K	K-B	K-D	K-E	Avg.
Source-only	89.7	88.4	90.9	90.1	88.5	90.2	86.9	88.5	91.5	87.6	87.3	91.2	89.2
DANN [22]	90.2	89.5	90.9	91.0	90.6	90.2	87.1	87.5	92.8	87.8	87.6	93.2	89.9
VAT [40]	90.6	91.0	91.7	90.8	90.8	92.0	87.2	86.9	92.6	86.9	87.7	92.9	90.1
VAT+Entropy	90.4	91.3	91.5	91.0	91.1	92.4	87.5	86.3	92.4	86.5	87.5	93.1	90.1
MDD [73]	90.4	90.4	91.8	90.2	90.9	91.0	87.5	86.3	92.5	**89.0**	87.9	92.1	90.0
CST	**91.5**	**92.9**	**92.6**	**91.9**	**92.6**	**93.5**	**90.2**	**89.4**	**93.8**	87.9	**88.3**	**93.5**	**91.5**

Table 4 shows the results on *VisDA-2017*. **CST** surpasses state-of-the-arts with ResNet-50 and ResNet-101 backbones. We also combine feature adaptation and self-training (DIRT-T, CDAN+VAT+entropy and MDD+FixMatch) to test if feature adaptation alleviates the negative effect of domain shift in standard self-training. Results indicate that CST is a better solution than simple combination.

While most traditional self-training methods include techniques specified for ConvNets such as Mixup [72], **CST** is a *universal* method and can directly work on sentiment classification by simply replacing the head and training objective of BERT [16]. In Table 3, most feature adaptation baselines improve over source only marginally, but **CST** outperforms all baselines on most tasks significantly.

5.4 Analysis

Table 5: Ablation on VisDA-2017.

Method	Accuracy ↑	d_{TV} ↓
FixMatch [57]	74.5 ± 0.2	0.22
Fixmatch+Tsallis	76.3 ± 0.8	0.15
CST w/o Tsallis	72.0 ± 0.4	0.16
CST+Entropy	76.2 ± 0.6	0.20
CST	**79.9** ± 0.5	0.12

Ablation Study. We study the role of each part of CST in self-training. CST w/o Tsallis removes the Tsallis entropy $L_{\text{Tsallis},\alpha}$. CST+Entropy replaces the Tsallis entropy with standard entropy. FixMatch+Tsallis adds $L_{\text{Tsallis},\alpha}$ to standard self-training. Observations are shown in Table 5. CST+Entropy performs 3.7% worse than CST, indicating that Tsallis entropy is a better regularization for pseudo-labels than standard entropy. CST performs 5.4% better than FixMatch, indicating that CST is better adapted to domain shift than standard self-training. While FixMatch+Tsallis outperforms FixMatch, it is still 3.6% behind CST, with much larger total variation distance d_{TV} between pseudo-labels and ground-truths, indicating that CST makes pseudo-labels more reliable than standard self-training under domain shift.

Quality of Pseudo-labels. We visualize the error of pseudo-labels during training on VisDA-2017 in Figure 5 (Left). The error of target classifier θ_t on the source domain decreases quickly in training, when both the error of pseudo-labels (error of θ_s on Q) and the total variation (TV) distance between pseudo-labels and ground-truths continue to decay, indicating that CST gradually refines pseudo-labels. This forms a clear contrast to standard self-training as visualized in Figure 2 (Middle), where the distance d_{TV} remains nearly unchanged throughout training.

Comparison of Gibbs entropy and Tsallis entropy. We compare the pseudo-labels learned with standard Gibbs entropy and Tsallis entropy on Ar→Cl with ResNet-50 at epoch 40. We compute the difference between *the largest and the second largest softmax scores* of each target example and plot the histogram in Figure 5 (Right). Gibbs entropy makes the largest softmax output close to 1, indicating over-confidence. In this case, if the prediction is wrong, it can be hard to correct it using self-training. In contrast, Tsallis entropy allows the largest and the second largest scores to be similar.

Table 4: Mean Class Accuracy (%) for unsupervised domain adaptation on VisDA-2017.

Method	ResNet-50	ResNet-101	Method	ResNet-50	ResNet-101
DANN [22]	69.3	79.5	CBST [77]	–	76.4 ± 0.9
VAT [40]	68.0 ± 0.3	73.4 ± 0.5	KLD [78]	–	78.1 ± 0.2
DIRT-T [56]	68.2 ± 0.3	77.2 ± 0.5	MDD [73]	74.6	81.6 ± 0.3
MCD [54]	69.2	77.7	AFN [69]	–	76.1
CDAN [37]	70.0	80.1	MDD+IA [28]	75.8	–
CDAN+VAT+Entropy	76.5 ± 0.5	80.4 ± 0.7	MDD+FixMatch	77.8 ± 0.3	82.4 ± 0.4
MixMatch	69.3 ± 0.4	77.0 ± 0.5	STAR [38]	–	82.7
FixMatch [57]	74.5 ± 0.2	79.5 ± 0.3	SENTRY [48]	76.7	–
CST	$\underline{79.9} \pm 0.5$	$\underline{84.8} \pm 0.6$	**CST+SAM**	$\mathbf{80.6} \pm 0.5$	$\mathbf{86.5} \pm 0.7$

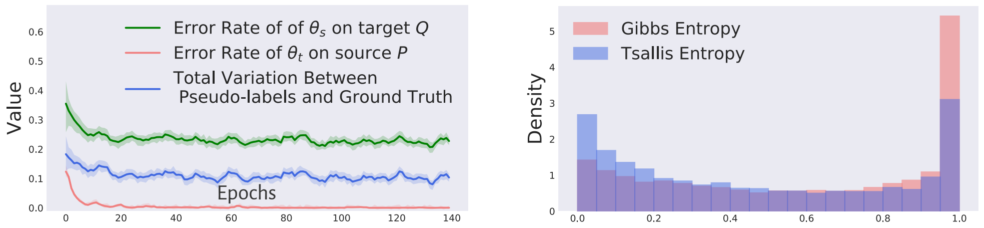


Figure 5: **Analysis.** Left: Error of pseudo-labels and reverse pseudo-labels. The error of target classifier θ_t on the source domain decreases, indicating the quality of pseudo-labels is refined. Right: Histograms of the difference between the largest and the second largest softmax scores. Tsallis entropy avoids over-confidence.

6 Related Work

Self-Training. Self-training is a mainstream technique for semi-supervised learning [13]. In this work, we focus on pseudo-labeling [52, 31, 2], which uses unlabeled data by training on pseudo-labels generated by a source model. Other lines of work study consistency regularization [4, 51, 55, 40]. Recent works demonstrate the power of such methods [67, 57, 23]. Equipped with proper training techniques, these methods can achieve comparable results as standard training that uses much more labeled examples [17]. Zoph et al. [76] compare self-training to pre-training and joint training. Vu et al. [65], Mukherjee & Awadallah [42] show that task-level self-training works well in few-shot learning. These methods are tailored to semi-supervised learning or general representation learning and do not take domain shift into consideration explicitly. Wei et al. [66], Frei et al. [20] provide the first nice theoretical analysis of self-training based on the expansion assumption.

Domain Adaptation. Inspired by the generalization error bound of Ben-David et al. [7], Long et al. [34], Zellinger et al. [71] minimize distance measures between source and target distributions to learn domain-invariant features. Ganin et al. [22] (DANN) proposed to approximate the domain distance by adversarial learning. Follow-up works proposed various improvement upon DANN [63, 54, 37, 73, 28]. Popular as they are, failure cases exist in situation like label shift [74, 32], shift in support of domains [29], and large discrepancy between source and target [33]. Another line of works try to address domain adaptation with self-training. Shu et al. [56] improves DANN with VAT and entropy minimization. French et al. [21], Zou et al. [78], Li et al. [32] incorporated various semi-supervised learning techniques to boost domain adaptation performance. Kumar et al. [30], Chen et al. [15] and Cai et al. [11] showed self-training provably works in domain adaptation under certain assumptions.

7 Conclusion

We propose cycle self-training in place of standard self-training to explicitly address the distribution shift in domain adaptation. We show that our method provably works under the expansion assumption and demonstrate hard cases for feature adaptation and standard self-training. Self-training (or pseudo-labeling) is only one line of works in the semi-supervised learning literature. Future work can delve into the behaviors of other semi-supervised learning techniques including consistency regularization and data augmentation under distribution shift, and exploit them extensively for domain adaptation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 62022050 and 62021002, Beijing Nova Program under Grant Z201100006820041, China's Ministry of Industry and Information Technology, the MOE Innovation Plan and the BNRist Innovation Fund.

References

[1] Albadawy, E. A., Saha, A., and Mazurowski, M. A. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical Physics*, 45(3), 2018.

[2] Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *CoRR*, abs/1908.02983, 2019.

[3] Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In *NeurIPS*, pp. 8141–8150. 2019.

[4] Bachman, P., Alsharif, O., and Precup, D. Learning with pseudo-ensembles. In *NeurIPS*, volume 27, pp. 3365–3373, 2014.

[5] Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3(Nov):463–482, 2002.

[6] Ben-David, S. and Urner, R. On the hardness of domain adaptation and the utility of unlabeled target samples. In *ALT*, pp. 139–153, 2012.

[7] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[8] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[9] Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

[10] Blitzer, J., Dredze, M., and Pereira, F. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pp. 440–447, 2007.

[11] Cai, T., Gao, R., Lee, J. D., and Lei, Q. A theory of label propagation for subpopulation shift, 2021.

[12] Carlini, N. Poisoning the unlabeled dataset of semi-supervised learning, 2021.

[13] Chapelle, O., Schölkopf, B., and Zien, A. *Semi-supervised learning*. MIT press Cambridge, 2006.

[14] Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., and Huang, J. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, pp. 627–636, 2019.

[15] Chen, Y., Wei, C., Kumar, A., and Ma, T. Self-training avoids using spurious features under domain shift. In *NeurIPS*, pp. 21061–21071, 2020.

[16] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.

[17] Du, J., Grave, E., Gunel, B., Chaudhary, V., Celebi, O., Auli, M., Stoyanov, V., and Conneau, A. Self-training improves pre-training for natural language understanding. In *NAACL*, pp. 5408–5418, 2021.

[18] Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pp. 1126–1135, 2017.

[19] Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021.

[20] Frei, S., Zou, D., Chen, Z., and Gu, Q. Self-training converts weak learners to strong learners in mixture models. *arXiv preprint arXiv:2106.13805*, 2021.

[21] French, G., Mackiewicz, M., and Fisher, M. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.

[22] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

[23] Ghiasi, G., Zoph, B., Cubuk, E. D., Le, Q. V., and Lin, T.-Y. Multi-task self-training for learning general representations. In *ICCV*, pp. 8856–8865, 2021.

[24] Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *NeurIPS*, pp. 529–536, 2004.

[25] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *ICML*, pp. 1321–1330, 2017.

[26] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

[27] Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pp. 1994–2003, 2018.

[28] Jiang, X., Lao, Q., Matwin, S., and Havaei, M. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *ICML*, pp. 4816–4827, 2020.

[29] Johansson, F. D., Sontag, D., and Ranganath, R. Support and invertibility in domain-invariant representations. In *AISTATS*, pp. 527–536, 2019.

[30] Kumar, A., Ma, T., and Liang, P. Understanding self-training for gradual domain adaptation. In *ICML*, pp. 5468–5479, 2020.

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

[32] Li, B., Wang, Y., Che, T., Zhang, S., Zhao, S., Xu, P., Zhou, W., Bengio, Y., and Keutzer, K. Rethinking distributional matching based domain adaptation. *ArXiv*, abs/2006.13352, 2020.

[33] Liu, H., Long, M., Wang, J., and Jordan, M. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, volume 97, pp. 4013–4022, 2019.

[34] Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In *ICML*, pp. 97–105, 2015.

[35] Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, pp. 136–144, 2016.

[36] Long, M., Zhu, H., Wang, J., and Jordan, M. I. Deep transfer learning with joint adaptation networks. In *ICML*, pp. 2208–2217, 2017.

[37] Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *NeurIPS*, pp. 1640–1650. 2018.

[38] Lu, Z., Yang, Y., Zhu, X., Liu, C., Song, Y.-Z., and Xiang, T. Stochastic classifiers for unsupervised domain adaptation. In *CVPR*, pp. 9111–9120, 2020.

[39] Mey, A. and Loog, M. A soft-labeled self-training approach. In *ICPR*, 2016.

[40] Miyato, T., Maeda, S., Ishii, S., and Koyama, M. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *TPAMI*, 2018.

[41] Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

[42] Mukherjee, S. and Awadallah, A. Uncertainty-aware self-training for few-shot text classification. In *NeurIPS*, volume 33, pp. 21199–21212, 2020.

[43] Pan, S. J. and Yang, Q. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[44] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, pp. 8026–8037, 2019.

[45] Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017.

[46] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *ICCV*, pp. 1406–1415, 2019.

[47] Prabhu, V., Khare, S., Kartik, D., and Hoffman, J. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation, 2020.

[48] Prabhu, V., Khare, S., Kartik, D., and Hoffman, J. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *ICCV*, pp. 8558–8567, October 2021.

[49] Qu, X., Zou, Z., Cheng, Y., Yang, Y., and Zhou, P. Adversarial category alignment network for cross-domain sentiment classification. In *NAACL*, 2019.

[50] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[51] Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. Semi-supervised learning with ladder networks. In *NeurIPS*, volume 28, pp. 3546–3554, 2015.

[52] Rosenberg, C., Hebert, M., and Schneiderman, H. Semi-supervised self-training of object detection models. In *WACV*, volume 1, pp. 29–36, 2005.

[53] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[54] Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.

[55] Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*, volume 29, pp. 1163–1171, 2016.

[56] Shu, R., Bui, H., Narui, H., and Ermon, S. A DIRT-t approach to unsupervised domain adaptation. In *ICLR*, 2018.

[57] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[58] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

[59] Talagrand, M. *Upper and lower bounds for stochastic processes: modern methods and classical problems*, volume 60. Springer Science & Business Media, 2014.

[60] Tan, S., Peng, X., and Saenko, K. Class-imbalanced domain adaptation: An empirical odyssey. In *ECCV Workshop*, 2020.

[61] Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, volume 30, pp. 1195–1204, 2017.

[62] Tsallis, C. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52(1-2):479–487, 1988.

[63] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

[64] Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pp. 5018–5027, 2017.

[65] Vu, T., Luong, M.-T., Le, Q. V., Simon, G., and Iyyer, M. Strata: Self-training with task augmentation for better few-shot learning. *arXiv preprint arXiv:2109.06270*, 2021.

[66] Wei, C., Shen, K., Yining, C., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *ICLR*, 2021.

[67] Xie, Q., Luong, M. T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

[68] Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., and Liang, P. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *ICLR*, 2021.

[69] Xu, R., Li, G., Yang, J., and Lin, L. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, 2019.

[70] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *NeurIPS*, pp. 3320–3328. 2014.

[71] Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., and Saminger-Platz, S. Central moment discrepancy (CMD) for domain-invariant representation learning. In *ICLR*, 2017.

[72] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[73] Zhang, Y., Liu, T., Long, M., and Jordan, M. Bridging theory and algorithm for domain adaptation. In *ICML*, pp. 7404–7413, 2019.

[74] Zhao, H., Combes, R. T. D., Zhang, K., and Gordon, G. On learning invariant representations for domain adaptation. In *ICML*, volume 97, pp. 7523–7532, 2019.

[75] Ziser, Y. and Reichart, R. Pivot based language modeling for improved neural domain adaptation. In *NAACL*, pp. 1241–1251, 2018.

[76] Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. Rethinking pre-training and self-training. In *NeurIPS*, volume 33, pp. 3833–3845, 2020.

[77] Zou, Y., Yu, Z., Vijaya Kumar, B. V. K., and Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pp. 297–313, 2018.

[78] Zou, Y., Yu, Z., Liu, X., Kumar, B. V., and Wang, J. Confidence regularized self-training. In *ICCV*, October 2019.

A Details in Section 4

A.1 Proof of Theorem 1

In Section 4.1, we study CST theoretically. In Theorem 1, we show that when the population error of the target classifier f_t on the source domain P is low and f_t is locally consistent, the source classifier f_s is guaranteed to perform well on the target domain Q. We further show that the consistency (robustness) is guaranteed by the confidence of the model (Lemma 3). Finally, we show in Theorem 2 that the minimizer of an objective function consistent with the CST objective in Section 3.1 leads to small target loss of the source classifier $\mathrm{Err}_Q(f_s)$.

We first review the assumptions made in Section 4.1 in order to prove Theorem 1. Consider a K-way classification problem. $f : X \to [0,1]^K \in \mathcal{F}$ and $\tilde{f}(x) := \arg\max_i f(x)_{[i]}$. We first state the properties of source and target distributions P and Q. Assume the source and target distributions are composed of K sub-populations, each corresponding to one class, and the sub-populations of different classes have disjoint support. This indicates that a ground truth labeling function exists, which is a common assumption as in [7, 66, 11]. We also assume for simplicity of presentation that $P(y=i) = Q(y=i)$. Note that our techniques can be directly applied to the case where $\frac{P(y=i)}{Q(y=i)}$ is bounded as in [11].

Assumption 1. *Denote by P_i and Q_i the conditional distribution of P and Q given $y=i$. We assume that: (1) $P(y=i) = Q(y=i)$, and (2) the supports of P_i and P_j are disjoint for $i \neq j$.*

Our analysis relies on the *expansion assumption* [66, 11], which intuitively states that the data distribution has good continuity within each class. Therefore, the subset in the support of a class will connect to its neighborhood, enabling knowledge transfer between domains. Wei et al. [66] justifies this assumption on real-world datasets with BigGAN.

Assumption 2 (**(q,ϵ)-constant expansion** [66]). *For any $x \in \mathcal{X}$, $\mathcal{N}(x)$ is defined as the* neighboring set *of x, $\mathcal{N}(x) = \{x' : d(x,x') \leq \xi\}$, where d is a proper metric. $\mathcal{N}(A) := \cup_{x\in A}\mathcal{N}(x)$. We say P and Q satisfy (q,ϵ)-constant expansion for some constants $q, \epsilon \in (0,1)$, if for any set $A \in \mathcal{X}$ and any $i \in [K]$ with $\frac{1}{2} > P_{\frac{1}{2}(P_i+Q_i)}(A) > q$, we have $P_{\frac{1}{2}(P_i+Q_i)}(\mathcal{N}(A)\backslash A) > \min\{\epsilon, P_{\frac{1}{2}(P_i+Q_i)}(A)\}$.*

Based on this expansion assumption, we consider a *robustness-constrained* version of CST for now. In Theorem 2, we will show that the population robustness loss is closely related to the uncertainty of of CST. Denote by f_s the source model and f_t the model trained on the target with pseudo-labels. Let $R(f_t) := P_{\frac{1}{2}(P+Q)}(\{x : \exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\})$ represent the robustness [66] of f_t on P and Q. Suppose $\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\tilde{f}_s(x) \neq \tilde{f}_t(x)) \leq c$ and $R(f_t) \leq \rho$. Theorem 1 states that when f_s and f_t behave similarly on Q (f_t fits the pseudo-labels generated by f_s on the target domain) and f_t is robust to local changes in input, the minimizer of the cycle source error $\mathrm{Err}_P(f_t)$ will guarantee low error on the target domain Q.

Theorem 1. *Suppose Assumption 1 and Assumption 2 hold for P and Q. For any f_s, f_t satisfying $\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\tilde{f}_s(x) \neq \tilde{f}_t(x)) \leq c$ and $R(f_t) \leq \rho$, the expected error of f_s on the target domain Q is bounded,*

$$\mathrm{Err}_Q(f_s) \leq \mathrm{Err}_P(f_t) + c + 2q + \rho \,/\, \min\{\epsilon, q\}. \tag{17}$$

We now turn to the proof of Theorem 1. We want to show the error of f_t on the source domain P is close to the error of f_s on the target domain Q. We first show that when the robustness error $R(f_t)$ is controlled, the error of f_t on the source and the target will be close. This is done by analyzing the error on each sub-population P_i and Q_i separately. Then we use the fact that the losses of f_s and f_t are also close when their disagreement on the target domain is controlled to obtain the final result.

Lemma 1 (Robustness on sub-populations). *Divide $[K]$ into S_1 and S_2, where for every $i \in S_1$, $\mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) < \min\{\epsilon, q\}$, and for every $i \in S_2$, $\mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) \geq \min\{\epsilon, q\}$. Under the condition of Theorem 1, we have*

$$\sum_{i\in S_1} P(y=i) \geq 1 - \frac{\rho}{\min\{\epsilon, q\}}. \tag{18}$$

Proof of Lemma 1. Suppose $\sum_{i\in S_1} P(y=i) < 1 - \frac{\rho}{\min\{\epsilon,q\}}$. Then we have $\sum_{i\in S_2} P(y=i) > \frac{\rho}{\min\{\epsilon,q\}}$, which implies

$$\begin{aligned}
&\mathbb{E}_{(x,y)\sim\frac{1}{2}(P+Q)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) \\
&= \sum_{i\in[K]} \mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\})P(y=i) \\
&\geq \sum_{i\in S_2} \mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\})P(y=i) \\
&> \min\{\epsilon,q\}\sum_{i\in S_2} P(y=i) \\
&= \rho.
\end{aligned}$$

Since we have $R(f_t) = \mathbb{E}_{(x,y)\sim\frac{1}{2}(P+Q)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) < \rho$, this forms a contradiction. □

We have established that for a large proportion of the sub-populations, the robustness is guaranteed. The next lemma shows that for each sub-population where the robustness is guaranteed, $\text{Err}_{P_i}(f_t)$ and $\text{Err}_{Q_i}(f_t)$ is close to each other by invoking the expansion assumption [66].

Lemma 2 (Accuracy propagates on robust sub-populations). *Under the condition of Theorem 1, if the sub-populations P_i and Q_i satisfy $\mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) < \min\{\epsilon,q\}$, we have*

$$|\text{Err}_{P_i}(f_t) - \text{Err}_{Q_i}(f_t)| \leq 2q. \tag{19}$$

Proof of Lemma 2. We claim that either $\text{Err}_{\frac{1}{2}(P_i+Q_i)}(f_t) \leq q$ or $\text{Err}_{\frac{1}{2}(P_i+Q_i)}(f_t) \geq 1-q$. On the one hand, if $\frac{1}{2} > \text{Err}_{\frac{1}{2}(P_i+Q_i)}(f_t) > q$, by the (q,ϵ)-expansion property (Definition 1), $P_{\frac{1}{2}(P_i+Q_i)}(\mathcal{N}(\{x : \tilde{f}_t(x) \neq i\})\backslash\{x : \tilde{f}_t(x) \neq i\}) > \min\{\epsilon,q\}$. Note that in $\mathcal{N}(\{x : \tilde{f}_t(x) \neq i\})\backslash\{x : \tilde{f}_t(x) \neq i\}$, $\tilde{f}_t(x) = i$. Thus, for x in the set $\mathcal{N}(\{x : \tilde{f}_t(x) \neq i\})\backslash\{x : \tilde{f}_t(x) \neq i\}$, there exists $x' \in \mathcal{N}(x)$, $\tilde{f}_t(x') \neq \tilde{f}_t(x) = i$.

$$\begin{aligned}
R(f_t) &= \mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) \\
&\geq \mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\})\mathbb{I}(x \in \mathcal{N}(\{x : \tilde{f}_t(x) \neq i\})\backslash\{x : \tilde{f}_t(x) \neq i\}) \\
&= P_{\frac{1}{2}(P_i+Q_i)}(\mathcal{N}(\{x : \tilde{f}_t(x) \neq i\})\backslash\{x : \tilde{f}_t(x) \neq i\}) \\
&> \min\{\epsilon,q\},
\end{aligned}$$

which contradicts the condition that $R(f_t) < \min\{\epsilon,q\}$.

On the other hand, if $\frac{1}{2} \leq \text{Err}_{\frac{1}{2}(P_i+Q_i)}(f_t) < 1-q$, the argument is similar. By the (q,ϵ)-expansion property (Definition 1), $P_{\frac{1}{2}(P_i+Q_i)}(\mathcal{N}(\{x : \tilde{f}_t(x) = i\})\backslash\{x : \tilde{f}_t(x) = i\}) > \min\{\epsilon,q\}$. Note that in $\mathcal{N}(\{x : \tilde{f}_t(x) = i\})\backslash\{x : \tilde{f}_t(x) = i\}$, $\tilde{f}_t(x) \neq i$. Thus, for x in the set $\mathcal{N}(\{x : \tilde{f}_t(x) = i\})\backslash\{x : \tilde{f}_t(x) = i\}$, there exists $x' \in \mathcal{N}(x)$, $i = \tilde{f}_t(x') \neq \tilde{f}_t(x)$.

$$\begin{aligned}
R(f_t) &= \mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\}) \\
&\geq \mathbb{E}_{(x,y)\sim\frac{1}{2}(P_i+Q_i)}\mathbb{I}(\exists x' \in \mathcal{N}(x), \tilde{f}_t(x) \neq \tilde{f}_t(x')\})\mathbb{I}(x \in \mathcal{N}(\{x : \tilde{f}_t(x) = i\})\backslash\{x : \tilde{f}_t(x) = i\}) \\
&= P_{\frac{1}{2}(P_i+Q_i)}(\mathcal{N}(\{x : \tilde{f}_t(x) = i\})\backslash\{x : \tilde{f}_t(x) = i\}) \\
&> \min\{\epsilon,q\},
\end{aligned}$$

which also contradicts the condition that $R(f_t) < \min\{\epsilon,q\}$.

Note that $\text{Err}_{\frac{1}{2}(P_i+Q_i)}(f_t) = \frac{1}{2}\text{Err}_{P_i}(f_t) + \frac{1}{2}\text{Err}_{Q_i}(f_t)$. Also we have $\text{Err}_{P_i}(f_t) \in [0,1]$. In consequence, we have either $\text{Err}_{P_i}(f_t), \text{Err}_{Q_i}(f_t) \in [0,2q]$ or $\text{Err}_{P_i}(f_t), \text{Err}_{Q_i}(f_t) \in [1-2q,1]$, which completes the proof. □

With Lemma 1 and Lemma 2 at hand, we can prove Theorem 1 by putting the analysis on each sub-population together.

Proof of Theorem 1.

$$\begin{aligned}
\mathrm{Err}_Q(f_t) &= \sum_{i\in[K]} \mathrm{Err}_{Q_i}(f_t)P(y=i) \\
&\le \sum_{i\in[S_1]} \mathrm{Err}_{Q_i}(f_t)P(y=i) + \sum_{i\in[S_2]} P(y=i) \\
&\le \sum_{i\in[S_1]} (\mathrm{Err}_{P_i}(f_t)+2q)P(y=i) + \sum_{i\in[S_2]} P(y=i) \\
&\le \mathrm{Err}_P(f_t) + 2q + \frac{\rho}{\min\{\epsilon, q\}},
\end{aligned}$$

where the second inequality holds due to Lemma 2, and the last holds due to Lemma 1. Also note that $\mathrm{Err}_Q(f_s) \le \mathrm{Err}_Q(f_t) + \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\arg\max_{[i]} f_s(x)_{[i]} \ne \arg\max_{[i]} f_t(x)_{[i]})$ by the triangle inequality. Adding these two equations results in Theorem 1. □

A.2 Proof of Theorem 2

To obtain finite-sample guarantee, we need additional assumptions on the function class $\mathcal{F}$.

Assumption 3. *The function class $\mathcal{F}$ satisfies the following properties: (1) $\mathcal{F}$ is closed to permutations of coordinates, (2) $0 \in \mathcal{F}$, and (3) each coordinate of f is L_f-Lipschitz w.r.t. $d(\cdot,\cdot)$.*

This assumption is also standard since common models for multi-class classification are symmetric for each class. Setting all the weight parameters of neural networks to 0 will result in 0 output.

We review the definition of terms in Theorem 2. The ramp function $\psi_\gamma : \mathbb{R} \to [0,1]$ is defined as:

$$\psi_\gamma(x) = \begin{cases} 1, & x \le 0 \\ 1-\frac{x}{\gamma}, & 0 < x \le \gamma \\ 0, & x > \gamma \end{cases} \tag{20}$$

The margin function is defined as $\mathcal{M}(v,y) = v_{[y]} - \max_{y'\ne y} v_{[y']}$ and $\mathcal{M}(v) = \max_y\{v_{[y]} - \max_{y'\ne y} v_{[y']}\}$. For multi-class classification problems, $\mathcal{M}(v)$ is closely related to the confidence, since it is equal to the difference between the largest and the second largest scores. The multi-class margin loss is composed of $\psi_\gamma(x)$ and $\mathcal{M}$: $l_\gamma(f(x),y) := \psi_\gamma(-\mathcal{M}(f(x),y))$. Denote by $L_{\widehat{P},\gamma}(f_t)$ the empirical margin loss of f_t on the source dataset $\widehat{P}$, $L_{\widehat{P},\gamma}(f_t) = \mathbb{E}_{(x,y)\sim\widehat{P}} l_\gamma(f_t(x),y)$. To measure the inconsistency of f_s and f_t, we extend the multi-class margin loss as $l_\gamma(f_s(x), f_t(x)) := \psi_\gamma(-\mathcal{M}(f_s(x), \tilde{f}_t(x)))$. Denote by $L_{\widehat{P},\gamma}(f_t,f_s)$ the empirical margin inconsistency loss of f_t and f_s on the source dataset $\widehat{P}$, $L_{\widehat{P},\gamma}(f_t,f_s) = \mathbb{E}_{(x,y)\sim\widehat{P}} l_\gamma(f_t(x), f_s(x))$.

Consider minimizing the following objective:

$$\min L_{\text{CST}}(f_s,f_t) := \underbrace{L_{\widehat{P},\gamma}(f_t)}_{\text{Cycle Loss}} + \underbrace{L_{\widehat{P},\gamma}(f_t,f_s)}_{\text{Target Loss}} + \underbrace{1-\mathbb{E}_{(x,y)\sim\frac{1}{2}(\widehat{P}+\widehat{Q})}\mathcal{M}(f_t(x))/\tau}_{\text{Uncertainty Loss}}. \tag{21}$$

Note that $L_{\widehat{P},\gamma}(f_t)$ is the loss of f_t on the source dataset (the cycle loss), and $L_{\widehat{P},\gamma}(f_t,f_s)$ is the training error of f_t on the target dataset. $\mathcal{M}(f_t(x))$ equals the difference between the largest and the second largest scores of $f_t(x)$, indicating the confidence of f_t. Thus, $1-\mathbb{E}_{(x,y)\sim\frac{1}{2}(\widehat{P}+\widehat{Q})}\mathcal{M}(f_t(x))$ is the uncertainty of f_t on the source and target datasets.

The following theorem shows that the minimizer of the training objective $L_{\text{CST}}(f_s,f_t)$ guarantees low population error of f_s on the target domain Q.

Theorem 2. *Under the condition of Theorem 1 and Assumption 3. For any solution of equation 13 and $\gamma > 0$, with probability larger than $1-\delta$,*

$$\mathrm{Err}_Q(f_s) \le L_{\text{CST}}(f_s,f_t) + 2q + \frac{4K}{\gamma}\left[\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}}) + \widehat{\mathcal{R}}(\tilde{\mathcal{F}}\times\mathcal{F}|_{\widehat{Q}})\right] + \frac{2}{\tau}\left[\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}}) + \widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{Q}})\right] + \zeta,$$

where $\zeta = O\left(\sqrt{\log(1/\delta)/n_s} + \sqrt{\log(1/\delta)/n_t}\right)$ is a low-order term. $\tilde{\mathcal{F}} \times \mathcal{F}$ refers to the function class $\{x \to f(x)_{[\tilde{f}'(x)]} : f, f' \in \mathcal{F}\}$. $\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}})$ denotes the empirical Rademacher complexity of function class $\mathcal{F}$ on dataset $\widehat{P}$.

We provide the function classes used in the proof. For a function class $f \in \mathcal{F} : \mathbb{R}^d \to [0,1]^K$, $\mathcal{F}_{[i]}$ denotes each coordinate of $\mathcal{F}$: $\mathcal{F}_{[i]} = \{x \to f(x)_{[i]} : f \in \mathcal{F}\}$. We also need other function classes based on $\mathcal{F}_{[i]}$. $\cup\mathcal{F}_{[i]}$ denotes the union of $\mathcal{F}_{[i]}$: $\cup\mathcal{F}_{[i]} = \cup_{i\in[K]}\mathcal{F}_{[i]}$. $\max_i \mathcal{F}_{[i]}$ is composed of the maximum coordinate of $f \in \mathcal{F}$ for all x: $\max_i \mathcal{F}_{[i]} = \{x \to \max_i f_{[i]}(x) : f \in \mathcal{F}\}$. $\max_{i' \neq \tilde{\mathcal{F}}} \mathcal{F}_{[i']}$ denotes the function class composed of the second largest coordinate of $f \in \mathcal{F}$ for all x: $\max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']} = \{x \to \max_{i \neq \tilde{f}(x)} f_{[i]}(x) : f \in \mathcal{F}\}$, which we require to study the finite sample properties of the confidence loss $\mathcal{M}(f(x))$. $\tilde{\mathcal{F}} \times \mathcal{F}$ denotes the function class $\{x \to f_{\tilde{f}'(x)}(x) : f, f' \in \mathcal{F}\}$. The Rademacher complexity of $\mathcal{F}_{[i]}$ on set $S = \{x_j\}_{j=1}^n$ of size n is: $\widehat{\mathcal{R}}(\mathcal{F}_{[i]}|_S) = \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f\in\mathcal{F}} \sum_{j=1}^n \sigma_j f(x_j)_{[i]}$. The Rademacher complexity of $\cup\mathcal{F}_{[i]}$ is $\widehat{\mathcal{R}}(\cup\mathcal{F}_{[i]}|_S) = \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f\in\mathcal{F}, i\in[K]} \sum_{j=1}^n \sigma_j f(x_j)_{[i]}$. The Rademacher complexity of $\max_i \mathcal{F}_{[i]}$ is $\widehat{\mathcal{R}}(\max_i \mathcal{F}_{[i]}|_S) = \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f\in\mathcal{F}, i\in[K]} \sum_{j=1}^n \sigma_j \max_i f(x_j)_{[i]}$. We further denote by $\widehat{\mathcal{R}}(\mathcal{F}|_S)$ the sum of the Rademacher complexity of each $\mathcal{F}_{[i]}$, $\widehat{\mathcal{R}}(\mathcal{F}|_S) = \sum_{i=1}^K \widehat{\mathcal{R}}(\mathcal{F}_{[i]}|_S)$.

To prove Theorem 2, we first observe the relationship between the confidence objective $\mathbb{E}_{(x,y)\sim\frac{1}{2}(\widehat{P}+\widehat{Q})}\mathcal{M}(f_t(x))$ and the robustness constraint $R(f_t) := P_{\frac{1}{2}(P+Q)}(\{x : \exists x' \in \mathcal{N}(x), \max_{[i]} f_t(x) \neq \max_{[i]} f_t(x')\})$ in Theorem 1. In fact, as shown in Lemma 3, when the output of the model is confident on the source and target dataset, i.e. $\mathbb{E}_{(x,y)\sim\frac{1}{2}(\widehat{P}+\widehat{Q})}\mathcal{M}(f_t(x))$ is large, the model is also robust to the change in input.

Lemma 3 (Confidence guarantees robustness)**.** *Under the conditions of Theorem 2, we have*

$$R(f_t) \leq \frac{1 - \mathbb{E}_{(x,y)\sim\frac{1}{2}(P+Q)}\mathcal{M}(f_t(x))}{1 - 2L_f\xi}. \tag{22}$$

Proof of Lemma 3. We first note that when $\max_y\{f_t(x)_{[y]} - \max_{y'\neq y} f_t(x)_{[y']}\} > 2L_f\xi$, the $\arg\max_i f_t(x)_{[i]}$ will not change in the neighborhood $\mathcal{N}(x)$ since $f_{[i]}$ is L_f-Lipschitz for all i. Suppose $y^* = \arg\max_y f_t(x)_{[y]}$. For all $y' \neq y^*$ and $x' \in \mathcal{N}(x)$,

$$\begin{aligned} f_t(x')_{[y^*]} - f_t(x')_{[y']} &> f_t(x)_{[y^*]} - L_f d(x,x') - (f_t(x')_{[y']} + L_f d(x,x')) && (23)\\ &\geq \max_y\{f_t(x)_{[y]} - \max_{y'\neq y} f_t(x)_{[y']}\} - 2L_f d(x,x') && (24)\\ &\geq 0. && (25)\end{aligned}$$

Therefore, we have

$$\begin{aligned} R(f_t) &\leq 1 - P_{\frac{1}{2}(P+Q)}\left(\mathcal{M}(f_t(x)) > 2L_f\xi\right) && (26)\\ &\leq \frac{1 - \mathbb{E}_{(x,y)\sim\frac{1}{2}(P+Q)}\mathcal{M}(f_t(x))}{1-2L_f\xi}, && (27)\end{aligned}$$

where the second inequality holds because $f_{[i]} \in [0,1]$, and $\mathcal{M}(f(x)) \in [0,1]$. □

To obtain finite sample guarantee, we aim to show that each term in equation 13 is close to its population version. We first present Lemma 4, the classical result for multi-class classification.

Lemma 4 (Lemma 3.1 of [41])**.** *Suppose $f \in \mathcal{F}$ and $\gamma > 0$, with probability at least $1-\delta$ over the sampling of $\widehat{P}$, the following holds for all $f \in \mathcal{F}$ simultaneously,*

$$\mathrm{Err}_P(f) \leq L_{\widehat{P},\gamma}(f) + \frac{4K}{\gamma}\widehat{\mathcal{R}}(\cup\mathcal{F}_{[i]}|_{\widehat{P}}) + O\left(\sqrt{\log(1/\delta)\,/\,n_s}\right). \tag{28}$$

We then extend Lemma 4 to study the finite sample properties of $L_{\widehat{Q},\gamma}(f_t, f_s)$ and $\mathbb{E}\mathcal{M}(f_t(x))$.

Lemma 5. *Suppose $f \in \mathcal{F}$ and $\gamma > 0$, with probability at least $1-\delta$ over the sampling of $\widehat{P}$, the following holds for all $f \in \mathcal{F}$ simultaneously,*

$$\mathbb{E}_{(x,y)\sim P}\mathcal{M}(f(x)) \leq \mathbb{E}_{(x,y)\sim \widehat{P}}\mathcal{M}(f(x)) + 4\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_s}\right). \tag{29}$$

Proof of Lemma 5. By standard Rademacher complexity bound (Theorem 7 of Bartlett & Mendelson [5]), we have

$$\mathbb{E}_{(x,y)\sim P}\mathcal{M}(f(x)) \leq \mathbb{E}_{(x,y)\sim \widehat{P}}\mathcal{M}(f(x)) + 2\widehat{\mathcal{R}}(\mathcal{M}\circ\mathcal{F}|_{\widehat{P}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_s}\right). \tag{30}$$

Thus it remains to show $\widehat{\mathcal{R}}(\mathcal{M}\circ\mathcal{F}|_{\widehat{P}}) \leq 2\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}})$. In fact,

$$\begin{aligned}
\widehat{\mathcal{R}}(\mathcal{M}\circ\mathcal{F}|_{\widehat{P}}) &= \frac{1}{n_s}\mathbb{E}_\sigma \sup_{f\in\mathcal{F}}\sum_{i=1}^{n_s}\sigma_i \max_y\{f(x_i)_{[y]} - \max_{y'\neq y} f(x_i)_{[y']}\} \\
&= \frac{1}{n_s}\mathbb{E}_\sigma \sup_{f\in\mathcal{F}}\sum_{i=1}^{n_s}\sigma_i (\max_y f(x_i)_{[y]} - \max_{y'\neq \tilde{f}(x_i)} f(x_i)_{[y']}) \\
&\leq \frac{1}{n_s}\mathbb{E}_\sigma \sup_{f\in\mathcal{F}}\sum_{i=1}^{n_s}\sigma_i \max_y f(x_i)_{[y]} + \frac{1}{n_s}\mathbb{E}_\sigma \sup_{f\in\mathcal{F}}\sum_{i=1}^{n_s}\sigma_i \max_{y'\neq \tilde{f}(x_i)} f(x)_{[y']} \\
&= \widehat{\mathcal{R}}(\max_i \mathcal{F}_{[i]}|_{\widehat{P}}) + \widehat{\mathcal{R}}(\max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']}|_{\widehat{P}}).
\end{aligned}$$

As will be shown in Lemma 7, both $\widehat{\mathcal{R}}(\max_i \mathcal{F}_{[i]}|_{\widehat{P}})$ and $\widehat{\mathcal{R}}(\max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']}|_{\widehat{P}})$ are smaller than $\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}})$, which completes the proof. □

Lemma 6. *Suppose $f_s, f_t \in \mathcal{F}$ and $\gamma > 0$, with probability at least $1-\delta$ over the sampling of $\widehat{Q}$, the following holds for all $f_s, f_t \in \mathcal{F}$ simultaneously,*

$$\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(f_t(x)\neq f_s(x)) \leq L_{\widehat{Q},\gamma}(f_t, f_s) + \frac{2K}{\gamma}\widehat{\mathcal{R}}(\tilde{\mathcal{F}}\times\mathcal{F}|_{\widehat{Q}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_t}\right). \tag{31}$$

Proof of Lemma 6. By the definition of multi-class margin loss, we have $\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(f_t(x) \neq f_s(x)) \leq L_{Q,\gamma}(f_t, f_s)$. Denote by $\mathcal{G}$ the set of $\{x \to (-\mathcal{M}(f_t(x), f_s(x))) : f_t, f_s \in \mathcal{F}\}$. By standard Rademacher complexity bound, we have,

$$L_{Q,\gamma}(f_t, f_s) \leq L_{\widehat{Q},\gamma}(f_t, f_s) + 2\widehat{\mathcal{R}}(\psi_\gamma\circ\mathcal{G}|_{\widehat{Q}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_t}\right).$$

By Talagrand contraction Lemma [59], $\widehat{\mathcal{R}}(\psi_\gamma\circ\mathcal{G}|_{\widehat{Q}}) \leq \frac{1}{\gamma}\widehat{\mathcal{R}}(\mathcal{G}|_{\widehat{Q}})$. Thus, it remains to show $\widehat{\mathcal{R}}(\mathcal{G}|_{\widehat{Q}}) \leq K\widehat{\mathcal{R}}(\tilde{\mathcal{F}}\times\mathcal{F}|_{\widehat{Q}})$. We have

$$\begin{aligned}
\widehat{\mathcal{R}}(\mathcal{G}|_{\widehat{Q}}) &= \frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\mathcal{M}(f_t(x_i), \tilde{f}_s(x_i)) \\
&= \frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\left(f_t(x_i)_{[\tilde{f}_s(x_i)]} - \max_{y'\neq\tilde{f}_s(x_i)} f_t(x_i)_{[y']}\right) \\
&\leq \frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i f_t(x_i)_{[\tilde{f}_s(x_i)]} + \frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\max_{y'\neq\tilde{f}_s(x_i)} f_t(x_i)_{[y']} \\
&= \widehat{\mathcal{R}}(\tilde{\mathcal{F}}\times\mathcal{F}|_{\widehat{Q}}) + \frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\max_{y'\neq\tilde{f}_s(x_i)} f_t(x_i)_{[y']}.
\end{aligned}$$

It remains to show $\frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\max_{y'\neq\tilde{f}_s(x_i)} f_t(x_i)_{[y']} \leq (K-1)\widehat{\mathcal{R}}(\tilde{\mathcal{F}}\times\mathcal{F}|_{\widehat{Q}})$, which is done by noting the closure of $\mathcal{F}$ under the permutation of coordinates. Consider the permutation $\upsilon : \mathbb{R}^K \to \mathbb{R}^K$: $\upsilon(v)_{[i]} = v_{[i-1]}$ for $i \in [2, 3, \cdots K]$ and $\upsilon(v)_{[1]} = v_{[K]}$.

$$\frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\max_{y'\neq\tilde{f}_s(x_i)} f_t(x_i)_{[y']} = \frac{1}{n_t}\mathbb{E}_{\sigma_i}\sup_{f_s,f_t}\sum_{i=1}^{n_t}\sigma_i\max_{k\in[K-1]}\upsilon^k f_t(x_i)_{[\tilde{f}_s(x_i)]}.$$

We have $\upsilon\mathcal{F} \subset \mathcal{F}$ by the closure of $\mathcal{F}$. Thus, $\tilde{\mathcal{F}} \times \upsilon\mathcal{F} \subset \tilde{\mathcal{F}} \times \mathcal{F}$. By Lemma 7, the Rademacher complexity of maximum of function classes is bounded with their sum, so we have $\frac{1}{n_t}\mathbb{E}_{\sigma_i} \sup_{f_s,f_t} \sum_{i=1}^{n_t} \sigma_i \max_{k\in[K-1]} \upsilon^k f_t(x_i)_{[\tilde{f}_s(x_i)]} \leq (K-1)\widehat{\mathcal{R}}(\tilde{\mathcal{F}} \times \mathcal{F}|_{\widehat{Q}})$. □

The next lemma shows the relationship between function classes. We establish the Rademacher complexity bounds of $\cup\mathcal{F}_{[i]}$, $\max_i \mathcal{F}_{[i]}$, and $\max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']}$. We show that the Rademacher complexity of these function classes can be bounded with $\widehat{\mathcal{R}}(\mathcal{F}|_S) = \sum_{i=1}^K \widehat{\mathcal{R}}(\mathcal{F}_{[i]}|_S)$.

Lemma 7. *Suppose* $\max_i \mathcal{F}_{[i]} = \{\max_i f_{[i]} : f \in \mathcal{F}\}$*,* $\cup\mathcal{F}_{[i]} = \{f_{[i]} : f \in \mathcal{F}, i \in [K]\}$*, and* $\max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']} = \max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']}|_{\widehat{P}} = \{x \to \max_{i\neq\tilde{f}(x)} f_{[i]}(x) : f \in \mathcal{F}\}$*.*

$$\widehat{\mathcal{R}}(\cup\mathcal{F}_{[i]}|_S) \leq \widehat{\mathcal{R}}(\mathcal{F}|_S),\ \widehat{\mathcal{R}}(\max_i \mathcal{F}_{[i]}|_S) \leq \widehat{\mathcal{R}}(\mathcal{F}|_S) \quad \textit{and} \quad \widehat{\mathcal{R}}(\max_{i'\neq\tilde{\mathcal{F}}} \mathcal{F}_{[i']}|_S) \leq \widehat{\mathcal{R}}(\mathcal{F}|_S). \tag{32}$$

Proof of Lemma 7. Consider the $K = 2$ case. Then we can repeat the arguments for $K-1$ times to get the final results.

For the first inequality, consider $\mathcal{F}'_{[i]} := \mathcal{F}_{[i]} \cup -\mathcal{F}_{[i]} = \{x \to \pm f(x) : f \in \mathcal{F}_{[i]}\}$. Then we have

$$\begin{aligned}
\widehat{\mathcal{R}}(\mathcal{F}_{[1]} \cup \mathcal{F}_{[2]}|_S) &= \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f_{[i]}\in\mathcal{F}_{[1]}\cup\mathcal{F}_{[2]}} \sum_{j=1}^n \sigma_j f_{[i]}(x_j) && (33)\\
&= \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f'_{[i]}\in\mathcal{F}'_{[1]}\cup\mathcal{F}'_{[2]}} \sum_{j=1}^n \left|\sigma_j f'_{[i]}(x_j)\right| && (34)\\
&\leq \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f'_{[i]}\in\mathcal{F}'_{[1]}} \sum_{j=1}^n \left|\sigma_j f'_{[i]}(x_j)\right| + \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f'_{[i]}\in\mathcal{F}'_{[2]}} \sum_{j=1}^n \left|\sigma_j f'_{[i]}(x_j)\right| && (35)\\
&= \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f_{[i]}\in\mathcal{F}_{[1]}} \sum_{j=1}^n \sigma_j f_{[i]}(x_j) + \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f_{[i]}\in\mathcal{F}_{[2]}} \sum_{j=1}^n \sigma_j f_{[i]}(x_j) && (36)\\
&= \widehat{\mathcal{R}}(\mathcal{F}_{[1]}|_S) + \widehat{\mathcal{R}}(\mathcal{F}_{[2]}|_S), && (37)
\end{aligned}$$

where equation equation 34 and equation 36 hold by the definition of $\mathcal{F}_{[i]}$.

For the second inequality, note that $\max\{x,y\} = \frac{x+y}{2} + \frac{|x-y|}{2}$. Then we apply Talagrand contraction lemma for the absolute value ($|\cdot|$ is 1-Lipschitz),

$$\begin{aligned}
\widehat{\mathcal{R}}(\max_i \mathcal{F}_{[i]}|_S) &= \frac{1}{n}\mathbb{E}_{\sigma_j} \sup_{f_{[1]}\in\mathcal{F}_{[1]}, f_{[2]}\in\mathcal{F}_{[2]}} \sum_{j=1}^n \sigma_j \left(\frac{f_{[1]}(x_j) + f_{[2]}(x_j)}{2} + \frac{|f_{[1]}(x_j) - f_{[2]}(x_j)|}{2}\right)\\
&\leq \frac{1}{2}\widehat{\mathcal{R}}(\mathcal{F}_{[1]}|_S) + \frac{1}{2}\widehat{\mathcal{R}}(\mathcal{F}_{[2]}|_S) + \frac{1}{2}\widehat{\mathcal{R}}(|\mathcal{F}_{[1]} - \mathcal{F}_{[2]}||_S)\\
&\leq \frac{1}{2}\widehat{\mathcal{R}}(\mathcal{F}_{[1]}|_S) + \frac{1}{2}\widehat{\mathcal{R}}(\mathcal{F}_{[2]}|_S) + \frac{1}{2}\widehat{\mathcal{R}}(\mathcal{F}_{[1]} - \mathcal{F}_{[2]}|_S)\\
&\leq \widehat{\mathcal{R}}(\mathcal{F}_{[1]}|_S) + \widehat{\mathcal{R}}(\mathcal{F}_{[2]}|_S).
\end{aligned}$$

For the third inequality, observe that the second largest of the set $\{x,y,z\}$ can be expressed as $\max\{\min\{x,y\}, \min\{\max\{x,y\}, z\}\}$. Following argument similar to the second inequality gives the proof. □

Now equipped with the lemmas above, we are ready to prove Theorem 2.

Proof of Theorem 2. By Lemmas 4, 5, and 6, we have the following inequalities hold with probability larger than $1 - \frac{\delta}{3}$,

$$\mathrm{Err}_P(f) \leq L_{\widehat{P},\gamma}(f) + \frac{4K}{\gamma}\widehat{\mathcal{R}}(\cup\mathcal{F}_{[i]}|_{\widehat{P}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_s}\right). \tag{38}$$

$$\mathbb{E}_{(x,y)\sim\frac{1}{2}(P+Q)}\mathcal{M}(f(x)) \le \mathbb{E}_{(x,y)\sim\frac{1}{2}(\widehat{P}+\widehat{Q})}\mathcal{M}(f(x)) + 2\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{P}}) + 2\widehat{\mathcal{R}}(\mathcal{F}|_{\widehat{Q}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_s} + \sqrt{\log(1/\delta)\ /\ n_t}\right). \tag{39}$$

$$\mathbb{E}_{(x,y)\sim Q}\mathbb{I}(f_t(x) \neq f_s(x)) \le L_{\widehat{Q},\gamma}(f_t, f_s) + \frac{2K}{\gamma}\widehat{\mathcal{R}}(\tilde{\mathcal{F}} \times \mathcal{F}|_{\widehat{Q}}) + O\left(\sqrt{\log(1/\delta)\ /\ n_t}\right). \tag{40}$$

We also have the following due to Lemma 3,

$$R(f_t) \le \frac{1 - \mathbb{E}_{(x,y)\sim\frac{1}{2}(P+Q)}\mathcal{M}(f_t(x))}{1 - 2L_f\xi}. \tag{41}$$

We use equation 39 and equation 40 as conditions. Plugging equation 38, equation 39, equation 40, and equation 41 into Theorem 1 and applying a union bound complete the proof of Theorem 2. □

A.3 Details in Section 4.2

We instantiate the domain adaptation setting in a quadratic neural network that allows us to compare various properties of the related algorithms. For a specific data distribution, we prove that (1) cycle self-training recovers target ground truth, and (2) both feature adaptation and standard self-training fail on the same distribution.

A.3.1 Setup

We study a quadratic neural network composed of a feature extractor $\phi \in \mathbb{R}^{d\times m}$ and a head $\theta \in \mathbb{R}^m$. $f_{\theta,\phi}(x) = g_\theta(h_\phi(x))$, where $g_\theta(z) = \theta^\top z$ and $h_\phi(x) = (\phi^\top x) \odot (\phi^\top x)$, $\odot$ is element-wise product. In training, we use the squared loss $\ell(f(x), y) = (f(x) - y)^2$. In testing, we map $f(x)$ to the nearest point in the output space: $\tilde{f}(x) := \arg\min_{y\in\{-1,0,1\}} |y - f(x)|$. Denote the expected error by $\mathrm{Err}_Q(\theta, \phi) := \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(\tilde{f}_{\theta,\phi}(x) \neq y)$.

Structural Covariate Shift and Label Shift. In domain adaptation, the source domain can have *multiple solutions* but we aim to learn the solution which works on the target domain [34]. Recent works also pointed out the source and the target label distributions are often different in real-world applications [74]. Following these properties, we design the underlying distributions p and q as shown in Table 6 to allow both structural covariate shift and label shift.

Table 6: Comparison of the design of the source and target.

Distribution	-1	$+1$	0
Source p	0.05	0.05	0.90
Target q	0.25	0.25	0.50

We study the following source distribution P. $x_{[1]}$ and $x_{[2]}$ are sampled *i.i.d.* from distribution p, and for $i \in [3, d]$, $x_{[i]} = \sigma_i \times x_{[2]}$. $\sigma_i \in \{\pm 1\}$ uniformly. In the target domain, $x_{[1]}$ and $x_{[2]}$ are sampled *i.i.d.* from distribution q, and for $i \in [3, d]$, $x_{[i]} = \sigma_i \times x_{[1]}$. $\sigma_i \in \{\pm 1\}$ uniformly. We also assume realizability: $y = x_{[1]}^2 - x_{[2]}^2$ for both source and target. For simplicity, we assume access to infinite *i.i.d.* examples of P ($n_s = \infty$) and n_t *i.i.d.* examples of Q. Therefore, the empirical loss and the population loss on the source domain are the same $L_P = L_{\widehat{P}}$.

Note that since $x_{[i]}^2 = x_{[2]}^2$ for all $i \in [3, d]$ in the source domain, $y = x_{[1]}^2 - x_{[i]}^2$ for all $i \in [2, d]$ are solutions to the source domain but only $y = x_{[1]}^2 - x_{[2]}^2$ works on the target domain. We visualize the setting when $d = 3$ in Figure 4.

A.3.2 Algorithms

We compare the baseline algorithms (feature adaptation and self-training) in Section 2 with the proposed CST. We study the norm-constrained versions of these algorithms.

Feature Adaptation chooses the source solution minimizing the distance between source and target feature distributions. We use total variation (TV) distance [7]: $d_{\text{TV}}(h_\sharp \widehat{P}, h_\sharp \widehat{Q}) = \sup_{E \subset \mathcal{Z}} |h_\sharp \widehat{P}(E) - h_\sharp \widehat{Q}(E)|$.

$$\hat{\theta}_{\text{FA}}, \hat{\phi}_{\text{FA}} = \arg\min_{\hat{\theta}_s, \hat{\phi}_s} \; d_{\text{TV}}(h_\sharp \widehat{P}, h_\sharp \widehat{Q}), \qquad (42)$$
$$\text{s.t. } \hat{\theta}_s, \hat{\phi}_s = \arg\min_{\theta,\phi} \|\theta\|_2^2 + \|\phi\|_F^2, \text{ s.t. } L_P(\theta, \phi) = 0.$$

Standard Self-Training first trains a source model,

$$\hat{\theta}_s, \hat{\phi}_s = \arg\min_{\theta,\phi} \|\theta\|_2^2 + \|\phi\|_F^2, \text{ s.t. } L_P(\theta, \phi) = 0. \qquad (43)$$

Then it trains the model on the source and target datasets jointly with source ground-truths and target pseudo-labels,

$$\hat{\theta}_{\text{ST}}, \hat{\phi}_{\text{ST}} = \arg\min_{\theta,\phi} \|\theta\|_2^2 + \|\phi\|_F^2, \qquad (44)$$
$$\text{s.t. } L_P(\theta, \phi) + \mathbb{E}_{x \sim \widehat{Q}} \ell(f_{\theta,\phi}(x), f_{\hat{\theta}_s, \hat{\phi}_s}(x)) = 0.$$

Cycle Self-Training. Following Section 3.1, we train the source head θ_s, and then train another head $\hat{\theta}_t(\phi)$ on the target dataset $\widehat{Q}$ with pseudo-labels generated by θ_s:

$$\hat{\theta}_t(\phi) = \arg\min_{\theta} \|\theta\|_2^2, \text{ s.t. } \mathbb{E}_{x \in \widehat{Q}} \ell(f_{\theta,\phi}(x), f_{\theta_s,\phi}(x)) = 0.$$

Finally we update the feature extractor ϕ to enforce consistent predictions of $\hat{\theta}_t(\phi)$ and θ_s on the source dataset:

$$\hat{\theta}_{\text{CST}}, \hat{\phi}_{\text{CST}} = \arg\min_{\theta_s,\phi} \|\theta_s\|_2^2 + \|\phi\|_F^2, \qquad (45)$$
$$\text{s.t. } L_P(\theta_s, \phi) + \mathbb{E}_{x \in P} \ell(g_{\theta_s}(h_\phi(x)), g_{\hat{\theta}_t(\phi)}(h_\phi(x))) = 0.$$

The following theorems show that both feature adaptation and standard self-training fail. The intuition is that the *ideal* solution that works on both source and target $y = x_{[1]}^2 - x_{[2]}^2$ has larger distance d_{TV} in the feature space than other solutions $y = x_{[1]}^2 - x_{[i]}^2$, so feature adaptation will not prefer the ideal solution. Standard self-training also fails because it will choose randomly among $y = x_{[1]}^2 - x_{[i]}^2$.

Theorem 3. *For any $\epsilon \in (0, 0.5)$, the following statements are true for feature adaptation and standard self-training:*

- *For any failure rate $\xi > 0$, and target dataset of size $n_t > \Theta(\log \frac{1}{\xi})$, with probability at least $1 - \xi$ over the sampling of target data, the source solution $\hat{\theta}_{\text{FA}}, \hat{\phi}_{\text{FA}}$ found by feature adaptation fails on the target domain:*

$$\text{Err}_Q(\hat{\theta}_{\text{FA}}, \hat{\phi}_{\text{FA}}) \geq \epsilon. \qquad (46)$$

- *With probability at least $1 - \frac{1}{d-1}$ over the training the source solution, the solution $(\hat{\theta}_{\text{ST}}, \hat{\phi}_{\text{ST}})$ of standard self-training satisfies*

$$\text{Err}_Q(\hat{\theta}_{\text{ST}}, \hat{\phi}_{\text{ST}}) \geq \epsilon. \qquad (47)$$

In comparison, we show that CST can recover the ground truth with high probability.

Theorem 4. *For any failure rate $\xi > 0$, and target dataset of size $n_t > \Theta(\log \frac{1}{\xi})$, with probability at least $1 - \xi$ over the sampling of target data, the feature extractor $\hat{\phi}_{\text{CST}}$ found by CST and the head $\hat{\theta}_{\text{CST}}$ recovers the ground truth of the target dataset:*

$$\text{Err}_Q(\hat{\theta}_{\text{CST}}, \hat{\phi}_{\text{CST}}) = 0. \qquad (48)$$

Intuitively, CST successfully learns the *transferable feature* $x_{[1]}^2 - x_{[2]}^2$ because it enforces the generalization of the head $\hat{\theta}_t(\phi)$ on the source data.

A.4 Proof of Theorem 3

We first describe the insights of the proof. As shown in Lemma 8, every source solution can be categorized into $d-1$ classes according to the coordinate l of the learned weight ϕ. Among those $d-1$ classes, only $l=2$ works on the target domain and $l \in \{3, \cdots d\}$ do not work on the target. We then show that in feature adaptation, $l \in \{3, \cdots d\}$ results in smaller distance between source and target feature distributions as a result of Lemma 9, thus feature adaptation will choose $l \in \{3, \cdots d\}$. On the other hand, standard self-training with randomly select l in the possible $d-1$ choices, but only $l=2$ works.

Lemma 8. *Under the condition of Section 4, any solution θ, ϕ to the Source Only problem*

$$\min_{\theta,\phi} \|\theta\|_2^2 + \|\phi\|_F^2, \text{ s.t. } L_P(\theta,\phi) = 0. \tag{49}$$

must have the following form: $\exists i, j \in \{2, 3, \cdots m\}$, $l \in \{2, 3, \cdots d\}$, $\phi_i = 2^{\frac{1}{6}} e_1$, $\phi_j = 2^{\frac{1}{6}} e_l$, $\phi_k = 0$ for $k \neq i, j$, and $\theta_i = \theta_j = 2^{-\frac{1}{3}}$, $\theta_k = 0$ for $k \neq i, j$.

Proof of Lemma 8. Define the symmetric matrix $A = \sum_{i=1}^m \theta_i \phi_i \phi_i^\top$, then the networks can be represented by A: $f_{\theta,\phi}(x) = x^\top A x$. We show that $A_{ij} = 0$ for $i \neq j$ if $f_{\theta,\phi}$ recovers source ground truth.

First, for $i, j > 1$, let $x_1 = \mathbf{1}$, $x_2 = \mathbf{1} - 2e_i - 2e_j$, $x_3 = \mathbf{1} - 2e_i$, and $x_4 = \mathbf{1} - 2e_j$, where $\{e_i\}$ are the standard bases. Since the source ground truth $y = x_{[1]}^2 - x_{[2]}^2$, and $x_{[k]} = \pm 1 x_{[2]}$ for $k \in \{3, 4, \cdots d\}$, $y_1 = y_2 = y_3 = y_4$.

$$\begin{aligned}
y_1 + y_2 + y_3 - y_4 =& x_1^\top A x_1 + x_2^\top A x_2 - x_3^\top A x_3 - x_4^\top A x_4 && (50)\\
=& 2\mathbf{1}^\top A \mathbf{1} + 4A_{ii} + 4A_{jj} - 4\mathbf{1}^\top A e_i - 4\mathbf{1}^\top A e_j + 8A_{ij} && (51)\\
&- (\mathbf{1}^\top A \mathbf{1} + 4A_{ii} + 4A_{jj} - 4\mathbf{1}^\top A e_i - 4\mathbf{1}^\top A e_j) && (52)\\
=& 8A_{ij} = 0. && (53)
\end{aligned}$$

We then show $A_{1,j} = 0$ for $j \in \{2, 3, \cdots d\}$ using the fact that $y_1 = y_4$.

$$\begin{aligned}
y_1 - y_4 =& x_1^\top A x_1 - x_4^\top A x_4 && (54)\\
=& \mathbf{1}^\top A \mathbf{1} - (\mathbf{1}^\top A \mathbf{1} - 4\mathbf{1}^\top A e_j + 4A_{jj}) && (55)\\
=& 4\mathbf{1}^\top A e_j - 4A_{jj} && (56)\\
=& 0. && (57)
\end{aligned}$$

From equation 53 we know $A_{ij} = 0$ if $i \neq 1$. Then we also have $A_{1j} = A_{j1} = 0$. Therefore we can write y in the following form: $y = x^\top A x = \sum_{i=1}^d A_{ii} x_{[i]}^2$ We also have the source ground truth $y = x_{[1]}^2 - x_{[2]}^2$, and $x_{[k]} = \pm 1 x_{[2]}$ for $k \in \{3, 4, \cdots d\}$. Then A must satisfy $A_{11} = 1$, $\sum_{i=2}^d A_{ii} = -1$, and all other entries of A equals to 0.

We have found the form of source ground truth matrix A. It suffices to show that the minimal norm solution of θ and ϕ subject to the form of A must be in the form of Lemma 8.

$$\begin{aligned}
\|\theta\|_2^2 + \|\phi\|_F^2 =& \sum_i^m \theta_i^2 + \frac{1}{2}\|\phi_i\|_2^2 + \frac{1}{2}\|\phi_i\|_2^2 && (58)\\
\geq& \sum_i^m 3 \cdot 2^{\frac{2}{3}} \left(|\theta_i| \|\phi_i\|_2^2\right)^{\frac{2}{3}} && (59)\\
\geq& 3 \cdot 2^{\frac{2}{3}} \left(\sum_{i:\theta_i > 0} \theta_i \|\phi_i\|_2^2\right)^{\frac{2}{3}} + 3 \cdot 2^{\frac{2}{3}} \left(\sum_{i:\theta_i \leq 0} -\theta_i \|\phi_i\|_2^2\right)^{\frac{2}{3}} && (60)\\
=& 3 \cdot 2^{\frac{2}{3}} \left(\sum_{i:A_{ii} > 0} A_{ii}\right)^{\frac{2}{3}} + 3 \cdot 2^{\frac{2}{3}} \left(\sum_{i:A_{ii} \leq 0} -A_{ii}\right)^{\frac{2}{3}} = 3 \cdot 2^{\frac{5}{3}}. && (61)
\end{aligned}$$

The first inequality holds due to AM-GM inequality, where it takes equality iff $\theta_i^2 = \frac{1}{2}\|\phi_i\|_2^2$ for all i. The second inequality holds due to Jensen inequality. The situation where both inequality take equality is exactly the form of Lemma 8. □

Lemma 9. *Suppose $\widehat{Q} = \{x_i^t\}_{i=1}^{n_t}$ are i.i.d. samples from target distribution Q, then with high probability, $\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0)$ is close to 0.5:*

$$P\left(\left|\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0) - 0.5\right| > t\right) \le e^{-2n_t t^2} \tag{62}$$

Proof of Lemma 9. Since each coordinate of x follows q, $\mathbb{I}(x_{[l]} = 0) - 0.5$ is a sub-Gaussian variable with $\sigma = 0.5$. We then apply standard Hoeffding's inequality to complete the proof. □

Proof of Theorem 3. We have the conclusion of Lemma 8. For simplicity we suppose without loss of generality that the source solution has the following form: $\exists\, l \in \{2, 3, \cdots d\}$, $\phi_1 = 2^{\frac{1}{6}}e_1$, $\phi_2 = 2^{\frac{1}{6}}e_l$, $\phi_k = 0$ for $k \in \{3, \cdots m\}$, and $\theta_1 = \theta_2 = 2^{-\frac{1}{3}}$, $\theta_k = 0$ for $k \in \{3, \cdots m\}$. Then these solutions can be categorized into two classes: (1) When $l = 2$, the source solution also works on the target, i.e. $L_Q(\theta, \phi) = 0$. (2) When $l \in \{3, \cdots d\}$, the source solution does not work on the target,

$$\begin{aligned}
\mathrm{Err}_Q(\theta, \phi) &= 1 - \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(f_{\theta,\phi}(x) = y) && (63)\\
&= 1 - \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(x_{[l]}^2 = x_{[2]}^2) && (64)\\
&= 1 - \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(x_{[l]} = 0 \text{ and } x_{[2]} = 0) - \mathbb{E}_{(x,y)\sim Q}\mathbb{I}(x_{[l]} \neq 0 \text{ and } x_{[2]} \neq 0) && (65)\\
&= 0.5 && (66)
\end{aligned}$$

To prove that feature adaptation learns the solution that does not work on the target domain, we show that with high probability, the solution belonging to situation (1) has larger total variation between source and target feature distributions $h_\sharp P$ and $h_\sharp\widehat{Q}$. In fact, the distributions of $h_\sharp P$ are the same for solutions in situation (1) and situation (2):

$$\mathbb{E}_{(x,y)\sim P}\left(h_\phi(x) = (z_1, z_2)\right) = \begin{cases} 0.81, & (z_1, z_2) = 2^{\frac{1}{3}}(0,0)\\ 0.09, & (z_1, z_2) = 2^{\frac{1}{3}}(0,1)\\ 0.09, & (z_1, z_2) = 2^{\frac{1}{3}}(1,0)\\ 0.01, & (z_1, z_2) = 2^{\frac{1}{3}}(1,1)\end{cases}$$

For the target dataset, the distribution of features is different for solutions from situation (1) and situation (2). When $l = 2$, denote by $h_{1\sharp}\widehat{Q}$ the feature distribution.

$$\mathbb{E}_{(x,y)\sim\widehat{Q}}\left(h_\phi(x) = (z_1, z_2)\right) = \begin{cases} \mathbb{E}^2_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0), & (z_1, z_2) = 2^{\frac{1}{3}}(0,0)\\ \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0)\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} \neq 0), & (z_1, z_2) = 2^{\frac{1}{3}}(0,1)\\ \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0)\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} \neq 0), & (z_1, z_2) = 2^{\frac{1}{3}}(1,0)\\ \mathbb{E}^2_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} \neq 0), & (z_1, z_2) = 2^{\frac{1}{3}}(1,1)\end{cases}$$

When $l \in \{3, \cdots d\}$, denote by $h_{2\sharp}\widehat{Q}$ the feature distribution. since $x_{[l]} = x_{[1]}$ in the target domain, (z_1, z_2) can only be $2^{\frac{1}{3}}(0,0)$ or $2^{\frac{1}{3}}(1,1)$,

$$\mathbb{E}_{(x,y)\sim\widehat{Q}}\left(h_\phi(x) = (z_1, z_2)\right) = \begin{cases} \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0), & (z_1, z_2) = 2^{\frac{1}{3}}(0,0)\\ \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} \neq 0), & (z_1, z_2) = 2^{\frac{1}{3}}(1,1)\end{cases}$$

We then instantiate Lemma 9 with $t = 0.14$: With probability at least $1 - \delta$, $0.36 < \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0) < 0.64$ for any $n_t \geq C\log\left(\frac{1}{\delta}\right)$, where $C > 26$ is a constant. Finally, we show that $d_{\mathrm{TV}}(h_\sharp P, h_{2\sharp}\widehat{Q}) < d_{\mathrm{TV}}(h_\sharp P, h_{1\sharp}\widehat{Q})$ as long as $0.36 < \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]} = 0) < 0.64$ to prove that feature adaptation will select solutions in situation (2).

$$\begin{aligned}
d_{\mathrm{TV}}(h_\sharp P, h_{1\sharp}\widehat{Q}) =& \frac{1}{2}\left|\mathbb{E}^2_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}=0)-0.81\right| + \frac{1}{2}\left|\mathbb{E}^2_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}\neq 0)-0.01\right| && (67)\\
&+\left|\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}=0)\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}\neq 0)-0.09\right| && (68)\\
=&0.81-\mathbb{E}^2_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}=0) && (69)\\
>&0.9-\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}=0) && (70)\\
=&\frac{1}{2}\left|\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}=0)-0.81\right| + \frac{1}{2}\left|\mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}\neq 0)-0.01\right| && (71)\\
=&d_{\mathrm{TV}}(h_\sharp P, h_{2\sharp}\widehat{Q}), && (72)
\end{aligned}$$

when $0.36 < \mathbb{E}_{(x,y)\sim\widehat{Q}}\mathbb{I}(x_{[l]}=0) < 0.64$, which completes the proof of feature adaptation.

In standard self-training, when training the source solution, the probability of l equalling each value in $\{2,3,\cdots d\}$ is the same, but only $l=2$ is the solution working on the source domain. Then when training on the source ground truth and target pseudo-labels, the model will make l unchanged. Thus the probability of recovering the target ground truth is only $\frac{1}{d-1}$. □

A.5 Proof of Theorem 4

Similar to the proof of Theorem 3, we use the conclusion of Lemma 8 to show that $l=2$ indicates the source solution that works on the target domain, while the solutions corresponding to $l\in\{3,\cdots d\}$ will have large error on the target domain. Then we show that only $l=2$ makes the training objective of cycle self-training $L_{\text{CST}}=0$. This is due to the fact that $l=2$ will make the spans of source and target features identical, while $l\in\{3,\cdots d\}$ makes the spans of source and target features different and thus $\hat{\theta}_t(\phi)\neq\theta_s$.

Proof of Theorem 4. We still use Lemma 8. To prove that cycle self-training recovers the target ground truth, it suffices to show that $\mathbb{E}_{x\in P}\ell(g_{\theta_s}(h_\phi(x)), g_{\hat{\theta}_t(\phi)}(h_\phi(x)))=0$ when $l=2$, and $\mathbb{E}_{x\in P}\ell(g_{\theta_s}(h_\phi(x)), g_{\hat{\theta}_t(\phi)}(h_\phi(x)))\neq 0$ when $l\in\{3,\cdots d\}$.

When $l\in\{3,\cdots d\}$, since $x^2_{[1]}=x^2_{[l]}$ in the target domain, the target pseudo-labels are all 0. Then we solve the problem $\hat{\theta}_t(\phi)=\arg\min_\theta\|\theta\|_2^2,$ s.t. $\mathbb{E}_{x\in\widehat{Q}}\ell(f_{\theta_s,\phi}(x), f_{\theta,\phi}(x))$ to get the target classifier $\hat{\theta}_t(\phi)$. Since we want the target solution with minimal norm, $\hat{\theta}_t(\phi)=0$, and we can calculate L_{CST} as follows:

$$\mathbb{E}_{x\in P}\ell(g_{\theta_s}(h_\phi(x)), g_{\hat{\theta}_t(\phi)}(h_\phi(x)))=\mathbb{E}_{(x,y)\sim P}(y-f_{\hat{\theta}_t(\phi),\phi}(x))^2=\mathbb{E}_{(x,y)\sim P}y^2=0.18. \quad (73)$$

When $l=2$, we show that $2^{\frac{1}{3}}e_1+2^{\frac{1}{3}}e_2, 2^{\frac{1}{3}}e_1, 2^{\frac{1}{3}}e_2$ and 0 all appear in the target feature set with high probability. The probability that the target feature set does not contain each one in $2^{\frac{1}{3}}e_1+2^{\frac{1}{3}}e_2, 2^{\frac{1}{3}}e_1, 2^{\frac{1}{3}}e_2$ and 0 equals to $\left(\frac{3}{4}\right)^{n_t}$. Therefore with a union bound we can show that $2^{\frac{1}{3}}e_1+2^{\frac{1}{3}}e_2, 2^{\frac{1}{3}}e_1, 2^{\frac{1}{3}}e_2$ and 0 all appear in the target feature set with probability at least $1-4\left(\frac{3}{4}\right)^{n_t}$. In this case, $\hat{\theta}_t(\phi)=\arg\min_\theta\|\theta\|_2^2,$ s.t. $\mathbb{E}_{x\in\widehat{Q}}\ell(f_{\theta_s,\phi}(x), f_{\theta,\phi}(x))$ results in $\hat{\theta}_t(\phi)=\theta_s$, which means if $n_t>\Theta(\log\frac{1}{\xi})$, with probability at least $1-\xi$ over the sampling of target data,

$$L_{\text{CST}}=\mathbb{E}_{x\in P}\ell(g_{\theta_s}(h_\phi(x)), g_{\hat{\theta}_t(\phi)}(h_\phi(x)))=0. \quad (74)$$

□

B Implementation Details

We use PyTorch [44] and run each experiment with 2080Ti GPUs. CBST, KLD, and IA results are from their original papers. We use the highest results in the literature for DANN, MCD, CDAN, and MDD. VAT, FixMatch, MixMatch and DIRT-T are adapted to our datasets from the official code. We adopt the pre-trained ResNet models provided in torchvision. For BERT implementation, we use the official checkpoint and PyTorch code from `https://github.com/huggingface/transformers`.

B.1 Dataset Details

OfficeHome [64] `https://www.hemanthdv.org/officeHomeDataset.html` is an object recognition dataset which contains images from 4 domains. It has about 15500 images organized into 65 categories. The dataset was collected using a python web-crawler that crawled through several search engines and online image directories. The authors provided a Fair Use Notice on their website.

VisDA-2017 [45] `https://github.com/VisionLearningGroup/taskcv-2017-public/tree/master/classification` uses synthetic object images rendered from CAD models as the training domain and real object images cropped from the COCO dataset as the validation domain. The authors provided a Term of Use on the website.

DomainNet [46] `http://ai.bu.edu/M3SDA/#dataset` contains images from clipart, infograph, painting, real, and sketch domains collected by searching a category name combined with a domain name from searching engines. The authors provided a Fair Use Notice on their website.

Amazon Review [10] `https://www.cs.jhu.edu/~mdredze/datasets/sentiment/` contains product reviews taken from Amazon.com from many product types (domains). Some domains (books and dvds) have hundreds of thousands of reviews. Others (musical instruments) have only a few hundred. Reviews contain star ratings (1 to 5 stars) that can be converted into binary labels if needed.

B.2 Bi-level Optimization

In Section 3.1, we highlight that the optimization of CST involves bi-level optimization. In the inner loop (equation 4), we train the target classifier $\theta_t(\phi)$ on top of the shared representations ϕ, thus $\theta_t(\phi)$ is a function of ϕ. Moreover, the target classifier $\theta_t(\phi)$ is trained with target pseudo-labels y', which are the sharpened version of the outputs of the source classifier θ_s on top of the shared representations ϕ. In this sense, $\theta_t(\phi)$ relies on θ_s and ϕ through y' implicitly, too. In the outer loop (equation 5), we update the shared representations ϕ and the source classifier θ_s to make both the source classifier θ_s and the target classifier $\theta_t(\phi)$ perform well on the source domain. Since $\theta_t(\phi)$ relies on ϕ and θ_s, the objective of equation 5 is a bi-level optimization problem. We can derive the gradient of the loss w.r.t. ϕ and θ_s as follows:

$$\begin{aligned}&\nabla_\phi[L_{\widehat{P}}(\theta_s,\phi)+L_{\widehat{P}}(\hat\theta_t(\phi),\phi)] \\ &=\nabla_\phi L_{\widehat{P}}(\theta_s,\phi)+\frac{\partial L_{\widehat{P}}(\hat\theta_t(\phi),\phi)}{\partial\phi}+\frac{\partial L_{\widehat{P}}(\hat\theta_t(\phi),\phi)}{\partial\hat\theta_t(\phi)}\frac{\mathrm{d}\hat\theta_t(\phi)}{\mathrm{d}\phi} \\ &=\nabla_\phi L_{\widehat{P}}(\theta_s,\phi)+\frac{\partial L_{\widehat{P}}(\hat\theta_t(\phi),\phi)}{\partial\phi}+\frac{\partial L_{\widehat{P}}(\hat\theta_t(\phi),\phi)}{\partial\hat\theta_t(\phi)}\left[\frac{\partial\hat\theta_t(\phi)}{\partial\phi}+\frac{\partial\hat\theta_t(\phi)}{\partial y'}\frac{\partial y'}{\partial\phi}\right].\end{aligned} \tag{75}$$

$$\nabla_{\theta_s}[L_{\widehat{P}}(\theta_s,\phi)+L_{\widehat{P}}(\hat\theta_t(\phi),\phi)]=\nabla_{\theta_s}L_{\widehat{P}}(\theta_s,\phi)+\frac{\partial L_{\widehat{P}}(\hat\theta_t(\phi),\phi)}{\partial\hat\theta_t(\phi)}\frac{\partial\hat\theta_t(\phi)}{\partial y'}\frac{\partial y'}{\partial\theta_s}. \tag{76}$$

However, following the standard practice in self-training, we use label-sharpening to obtain target pseudo-labels y', i.e. $y'=\arg\max_i\{f_{\theta_s,\phi}(x)_{[i]}\}$. Thus, y' is not differentiable w.r.t. θ_s and ϕ. We treat the gradient of y' w.r.t. θ_s and ϕ as 0 in equation 75 and equation 76, making optimization easier. This modification leads to exactly equation 7 and equation 8 in Algorithm 1 together with the Tsallis entropy loss.

Speeding up bi-level optimization with MSE loss. Standard methods of bi-level optimization backpropagate through the inner loop algorithm, which requires computing the second-order derivative (Hessian-vector products) and can be unstable. We propose to use MSE loss instead of cross entropy in the inner loop when training the head $\hat{\theta}(\phi)$ to calculate the analytical solution with least square and directly back-propagate to the outer loop without calculating second-order derivatives. The framework is as fast as training the two heads jointly. To adopt MSE loss in multi-class classification, we use the one-hot embedding as the output and train a multi-variate regressor following the protocol of Arora et al. [3]. We calculate the least square solution of $\theta_t(\phi)$ based on one minibatch following the protocol of Bertinetto et al. [9]. We also provide results of varying batchsize to verify the performance of this approximation in Table 7. Results indicate that the performance of CST is stable in a wide range of batchsizes.

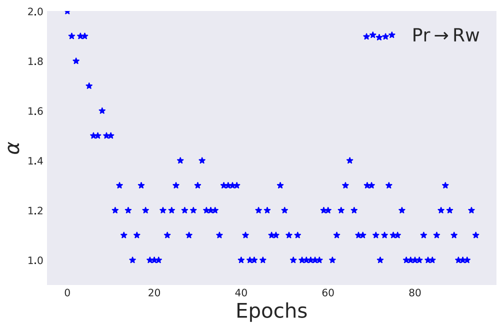
Table 7: Accuracy (%) on VisDA-2017 with ResNet-50

Method	Accuracy
CST (batchzize 32)	79.9 $\pm$ 0.6
CST (batchzize 64)	79.9 $\pm$ 0.5
CST (batchzize 128)	79.6 $\pm$ 0.4
CST (batchzize 256)	79.0 $\pm$ 0.6

B.3 Selection of α

We can also update α with gradient methods auto-differentiation tools as we treat ϕ. However, since α has only one parameter but many other parameters ($\theta_{s,\alpha}$ and $\theta_{t,\alpha}$) rely on it, using gradient methods is costly. To ease the computational cost, we choose to discretize the feasible region of $\alpha \in [1, 2]$ with $\alpha \in \{1.0, 1.1, \cdots 1.9, 2.0\}$, and train $\theta_{s,\alpha}$ with each $\alpha \in \{1.0, 1.1, \cdots 1.9, 2.0\}$ to generate pseudo-labels and train $\hat{\theta}_{t,\alpha}$ on pseudo-labels corresponding to each value of α. Then we select the $\alpha \in \{1.0, 1.1, \cdots 1.9, 2.0\}$ with best performance on the source dataset following equation 10. We also update α at the start of each epoch, since we found more frequent update leads to no performance gain. Since we only need to select α once at the start of each epoch, the resulting additional computational cost only relates to training the linear head on the source and target datasets for additional 11 times per epoch, which is negligible compared to training the backbone.

We plot the change of α throughout training in Figure 6. α converges to smaller value at the end of training, indicating that the penalization on uncertainty is increasing. Also note that α tends decrease slower for "heuristically distant" source and target domains. This corroborates the intuition that we need to penalize uncertain predictions mildly especially when the domain gap is large.



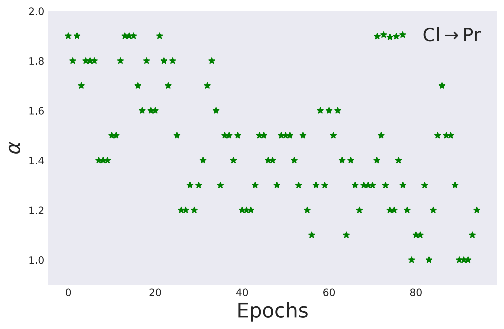


Figure 6: **Change of α during training.**

Difference between θ_s and $\theta_{s,\alpha}$. We use θ_s to update the feature extractor ϕ and test on the target domain after training. $\theta_{s,\alpha}$ is only used to search the optimal α, and the gradient does not back-propagate to ϕ.

C Additional Experiment Details

C.1 Additional Details of Section 2.1

In Figure 2 (Left), we use VisDA-2017 with 12 classes. To simulate the i.i.d., covariate shift and label shift setup, we resample the original Synthetic (source) and Real (target) datasets. In the i.i.d. setting, the labeled dataset consists of 1000 random samples per class from Real, and the unlabeled dataset consists of 1000 random samples (no overlapping with the labeled dataset) per class from Real. In the covariate shift setting, the labeled dataset consists of 1000 random samples per class from Synthetic, and the unlabeled dataset consists of 1000 random samples per class from Real. In the label shift setting, the number of examples is $[1800, 1440, 1152, 922, 737, 590, 472, 377, 302, 240, 193, 154]$ for each class of the labeled dataset and 1000 for each class of the unlabeled dataset, with both labeled and unlabeled datasets sampled randomly from Real. We train the model on labeled data until convergence to generate pseudo-labels for the unlabeled data. Then we calculate the ratio of classes in pseudo-labels and ground truth.

In Figure 2 (Middle), we also visualize the change of pseudo-label accuracy and the distance d_{TV} throughout standard self-training on original Synthetic and Real datasets. Here standard self-training refers to equation 2 with label-sharpening.

In both Figure 2 (Right) and this subsection, confidence refers to the maximum soft-max output value, and entropy is defined as $\sum_i -y_i \log(y_i)$. We change the confidence threshold from 0 to 1 and entropy threshold from 0 to log`(numclasses)`. Then we plot the point (False Positive Rate, True Positive Rate) in the plane.

In Section 2.1, we measure the quality of pseudo-labels with the total variation between pseudo-label distribution and ground-truth distribution. We show this quantity is upper-bounded by the accuracy of pseudo-labels. Intuitively, when the pseudo-label distribution and ground-truth distribution are the same, the output can still be incorrect. (e.g., in a binary problem, $P(Y = 1) = P(Y = 0) = 0.5$, and $\hat{Y} \sim \text{uniform}[0, 1]$ but is independent of X.) Recall that $d_{\text{TV}}(Y, \hat{Y}) = \sup_{E \subset [C]} |P(Y \in E) - P(\hat{Y} \in E)|$. Suppose the supremum is reached by $\hat{E}$. Without loss of generality, assume $P(Y = c) > P(\hat{Y} = c)$ for all $c \in \hat{E}$. Then $P(Y = c) \leq P(\hat{Y} = c)$ for all $c \notin \hat{E}$.

$$
\begin{aligned}
P(Y \neq \hat{Y}) &= \sum_{c=1}^{C} P(Y = c,\ \hat{Y} \neq c) \\
&= \sum_{c \in \hat{E}} P(Y = c,\ \hat{Y} \neq c) + \sum_{c \notin \hat{E}} P(Y = c,\ \hat{Y} \neq c) \\
&\geq \sum_{c \in \hat{E}} P(Y = c) - P(\hat{Y} = c) \\
&= d_{\text{TV}}(Y, \hat{Y}).
\end{aligned}
$$

In the equality, we use $P(Y = c) > P(\hat{Y} = c)$ when $c \in \hat{E}$, so $P(Y = c,\ \hat{Y} \neq c) \geq P(Y = c) - P(\hat{Y} = c)$ if $c \in \hat{E}$. Also note that $P(Y = c,\ \hat{Y} \neq c) \geq 0$ if $c \notin \hat{E}$.

In this subsection, we provide additional results of Section 2.1. We visualize the distributions of pseudo-labels and ground-truth with ResNet-50 backbones on Art→Clipart, Product→Art, Clipart→Real World, Art→Real World and Real World →Product tasks (without resampling) in Figures 7, 8, 9, 10, and 11 respectively. We also visualize the ROC curve of pseudo-label selection with confidence threshold. Results on Art→Clipart, Product→Art, Clipart→Real World, Art→Real World and Real World →Product are similar to VisDA-2017. When the pseudo-labels are generated from models trained on different distributions, they can become especially unreliable in that examples of several classes are almost misclassified into other classes. Domain shift also makes the selection of correct pseudo-labels more difficult than standard semi-supervised learning.

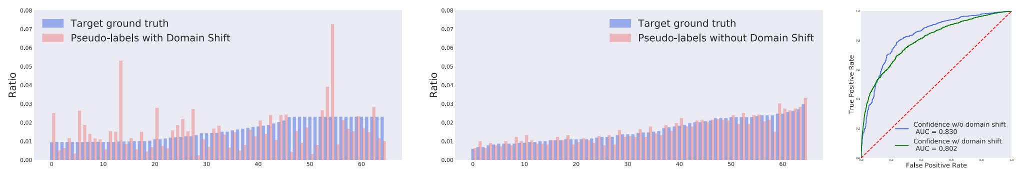


Figure 7: **Analysis of pseudo-labels under domain shift on Art→Clipart.** Left: Comparison of pseudo-label distributions with and without domain shift. Right: Comparison of pseudo-label selection with and without domain shift.

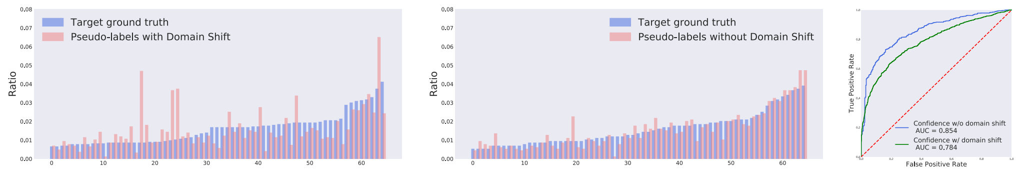


Figure 8: **Analysis of pseudo-labels under domain shift on Product→Art.** Left: Comparison of pseudo-label distributions with and without domain shift. Right: Comparison of pseudo-label selection with and without domain shift.

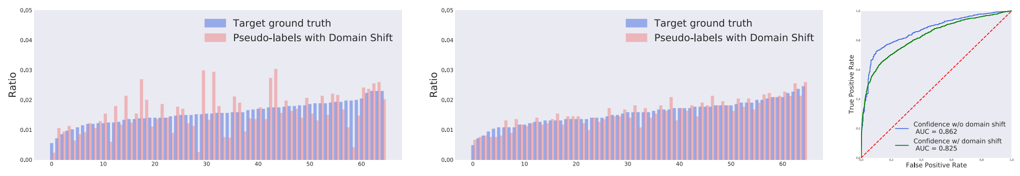


Figure 9: **Analysis of pseudo-labels under domain shift on Clipart→Real World.** Left: Comparison of pseudo-label distributions with and without domain shift. Right: Comparison of pseudo-label selection with and without domain shift.

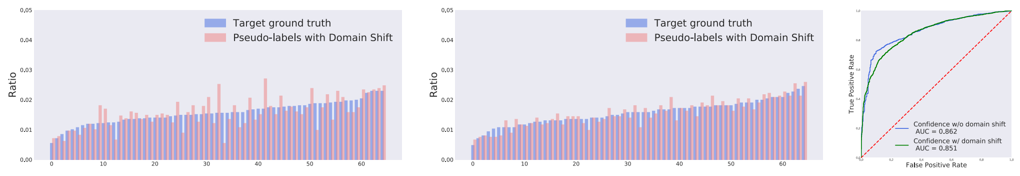


Figure 10: **Analysis of pseudo-labels under domain shift on Art→Real World.** Left: Comparison of pseudo-label distributions with and without domain shift. Right: Comparison of pseudo-label selection with and without domain shift.

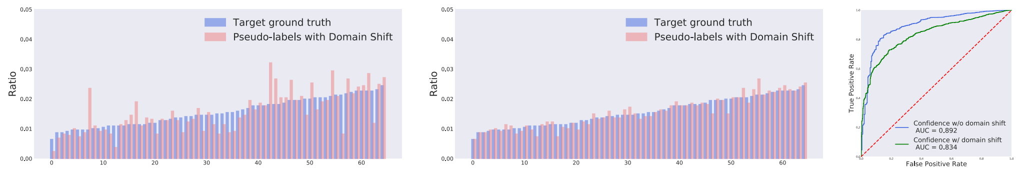


Figure 11: **Analysis of pseudo-labels under domain shift on Real World→Product.** Left: Comparison of pseudo-label distributions with and without domain shift. Right: Comparison of pseudo-label selection with and without domain shift.

C.2 Results on digit datasets

We provide results on the digit datasets to test the performance of the proposed method without using pre-training. We use DTN architecture following Long et al. [37]. Results in Table 8 indicate that CST achieve comparable performance to state-of-the-art.

Table 8: Accuracy (%) on digits datasets with DTN

Method	MNIST→USPS	SVHN→MNIST
CDAN	95.6 ± 0.2	96.9 ± 0.2
RWOT (CVPR 2020)	98.5 ± 0.2	97.5 ± 0.2
CST	98.5 ± 0.2	**98.2** ± 0.2

C.3 Results on DomainNet

We test the performance of the proposed method on the 40-class DomainNet [46] subset following the protocol of Tan et al. [60]. Results in Table 9 indicate that CST outperforms MDD by a large margin.

Table 9: Accuracy (%) on DomainNet for unsupervised domain adaptation (`ResNet-50`).

Method	R-C	R-P	R-S	C-R	C-P	C-S	P-R	P-C	P-S	S-R	S-C	S-P	Avg.
DANN [22]	63.4	73.6	72.6	86.5	65.7	70.6	86.9	73.2	70.2	85.7	75.2	70.0	74.5
COAL [60]	73.9	75.4	70.5	89.6	70.0	71.3	89.8	68.0	70.5	88.0	73.2	70.5	75.9
MDD [73]	77.6	75.7	74.2	89.5	74.2	75.6	90.2	76.0	74.6	86.7	72.9	73.2	78.4
CST	**83.9**	**78.1**	**77.5**	**90.9**	**76.4**	**79.7**	**90.8**	**82.5**	**76.5**	**90.0**	**82.8**	**74.4**	**82.0**

C.4 Standard deviations of Tables

We visualize the performance of CST and best baselines in Table 3 with standard deviations. Results indicate that the improvement of CST over previous methods is significant.

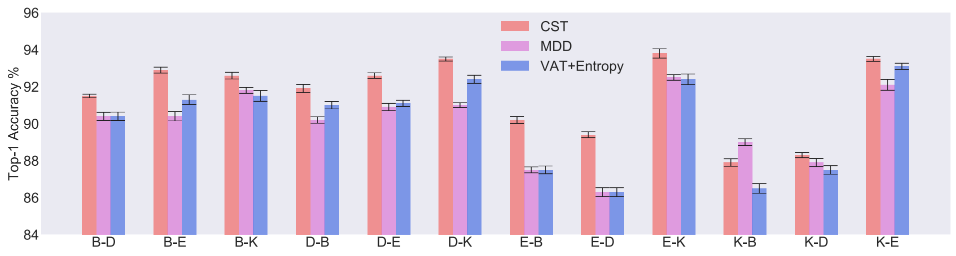


Figure 12: **Visualization of standard deviations of CST and baselines.** CST outperforms baselines significantly on all tasks except K→B.

D Limitations of CST and Future Directions

CST overcomes the drawbacks of standard pseudo-labeling in domain adaptation by dealing with the domain discrepancy explicitly with the cycle loss. However, pseudo-labeling is only one main direction of semi-supervised learning. Consistency regularization [40] and self-ensembling [4] are also important paradigms in semi-supervised learning. How to apply them to the setting with distribution shift and achieve consistent performance gain is still an open question. More recently, Carlini [12] investigated the effect of adversarial unlabeled data poisoning to self-training. Future works can extend CST to this setting and extend consistency regularization as a potential way of defense.

E Broader Impact

This work studies and improves self-training in the unsupervised domain adaptation setting. When deployed in real-world applications, distribution shift between labeled and unlabeled data can come in various ways. Although the quality of labeled datasets can be monitored, enabling the mitigation of bias in pre-processing, bias in unlabeled datasets can be intractable. Self-training with biased unlabeled data is highly risky since it may potentially amplify the biased models predictions. This work explores how to mitigate the effect of dataset bias in unlabeled data, and can potentially promote fair self-training systems.