Understanding Self-Training for Gradual Domain Adaptation

Ananya Kumar [1] **Tengyu Ma** [1] **Percy Liang** [1]

Abstract

Machine learning systems must adapt to data distributions that evolve over time, in applications ranging from sensor networks and self-driving car perception modules to brain-machine interfaces. Traditional domain adaptation is only guaranteed to work when the distribution shift is small; empirical methods combine several heuristics for larger shifts but can be dataset specific. To adapt to larger shifts we consider gradual domain adaptation, where the goal is to adapt an initial classifier trained on a source domain given only unlabeled data that shifts gradually in distribution towards a target domain. We prove the first non-vacuous upper bound on the error of self-training with gradual shifts, under settings where directly adapting to the target domain can result in unbounded error. The theoretical analysis leads to algorithmic insights, highlighting that regularization and label sharpening are essential even when we have infinite data. Leveraging the gradual shift structure leads to higher accuracies on a rotating MNIST dataset, a forest Cover Type dataset, and a realistic Portraits dataset.

1. Introduction

Machine learning models are typically trained and tested on the same data distribution. However, when a model is deployed in the real world, the data distribution typically evolves over time, leading to a drop in performance. This problem is widespread: sensor measurements drift over time due to sensor aging (Vergara et al., 2012), self-driving car vision modules have to deal with evolving road conditions (Bobu et al., 2018), and neural signals received by brain-machine interfaces change within the span of a day (Farshchian et al., 2019). Repeatedly gathering large sets of labeled examples to retrain the model can be impractical, so we would like to leverage unlabeled examples to adapt the model to maintain high accuracy (Farshchian et al., 2019; Sethi and Kantardzic, 2017).

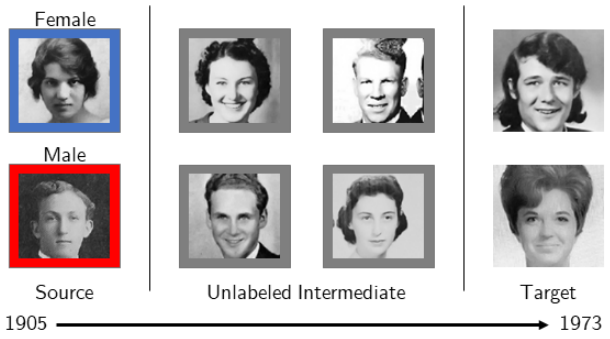


Figure 1. In gradual domain adaptation we are given labeled data from a source domain, and unlabeled data from intermediate domains that shift gradually in distribution towards a target domain. Here, blue = female, red = male, and gray = unlabeled data.

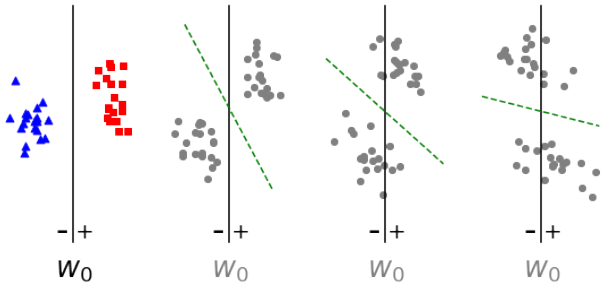
The traditional solution to adapt to the distribution shift is unsupervised domain adaptation, but existing methods are only guaranteed to work when the distribution shift is small—when the source and target distributions cannot be easily distinguished from each other (Zhao et al., 2019). To adapt to larger shifts, recent empirical papers propose combining several heuristics (Hoffman et al., 2018; Shu et al., 2018), but these methods can be brittle, working well on some datasets but not on others (Peng et al., 2019) and requiring substantial tuning for new domains.

In many real applications the domain shift does not happen at one time, but happens gradually, although this structure is ignored by most domain adaptation methods. *We show that the gradual shift structure allows us to reliably adapt to very different distributions, both in theory and practice.* We analyze self-training (also known as pseudolabeling), a method for semi-supervised learning (Chapelle et al., 2006) that has led to state-of-the-art results on ImageNet (Xie et al., 2020) and adversarial robustness on CIFAR-10 (Uesato et al., 2019; Carmon et al., 2019; Najafi et al., 2019).

Intuitively, it is easier to handle smaller shifts, but for each shift we can incur some error so the more steps, the more degradation—making it unclear whether leveraging the gradual shift structure is better than directly adapting to the tar-

[1]Stanford University, Stanford, California, USA. Correspondence to: Ananya Kumar <ananya@cs.stanford.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.



(a) t = 0 (b) t = 1 (c) t = 2 (d) t = 3

Figure 2. The source classifier w_0 gets 100% accuracy on the source domain (Figure 2a), where we have labeled data. But after 3 time steps (Figure 2d) the source classifier is stale, classifying most examples incorrectly. Now, we cannot correct the classifier using unlabeled data from the target domain, which corresponds to traditional domain adaptation directly to the target. Given *unlabeled* data in an intermediate domain (Figure 2b) where the shift is gradual, the source classifier pseudolabels most points correctly, and self-training learns an accurate classifier (show in green) that separates the classes. Successively applying self-training learns a good classifier on the target domain (green classifier in Figure 2d).

get. In this paper, *we provide the first theoretical analysis showing that gradual domain adaptation improves over the traditional approach of direct domain adaptation*, allowing us to adapt to very different target distributions.

As a concrete example of our setting, the Portraits dataset (Ginosar et al., 2017) contains photos of high school seniors taken across many years, labeled by gender (Figure 1). We use the first 2000 images (1905 - 1935) as the source, next 14000 (1935 - 1969) as intermediate domains, and next 2000 images as the target (1969 - 1973). A model trained on labeled examples from the source gets 98% accuracy on held out examples in the same years, but only 75% accuracy on the target domain. Assuming access to *unlabeled* images from intermediate domains, our goal is to adapt the model to do well on the target domain. Direct adaptation to the target with self-training only improves the accuracy a little, from 75% to 77%.

The gradual self-training algorithm begins with a classifier w_0 trained on labeled examples from the source domain (Figure 2a). For each successive domain P_t, the algorithm generates pseudolabels for unlabeled examples from that domain, and then trains a regularized supervised classifier on the pseudolabeled examples. The intuition, visualized in Figure 2, is that after a single gradual shift, most examples are pseudolabeled correctly so self-training learns a good classifier on the shifted data, but the shift from the source to the target can be too large for self-training to correct. *We find that gradual self-training on the Portraits dataset improves upon direct target adaptation (77% to 84% accuracy).*

Our results: We analyze gradual domain adaptation in two settings. The key challenge for domain adaptation theory is dealing with source and target domains that are very different, for example where the source and target can be easily discriminated / distinguished (Zhao et al., 2019; Shu et al., 2018), which are typical in the modern high-dimensional regime. The gradual shift structure inherent in many applications provides us with leverage to handle adapting to target distributions that are very different.

Our first setting, the margin setting, is distribution-free—we only assume that at every point in time there exists some Lipschitz classifier that can classify most of the data correctly with a margin, where the classifier may be different at each time step (so this is more general than covariate shift), and that the shifts are small in Wasserstein-infinity distance. The classifier can be non-linear. A simple example (as in Figure 2) shows that a classifier that gets 100% accuracy can get 0% accuracy after a constant number of time steps. Directly adapting to the final target domain also gets 0% accuracy. Gradual self-training does better, letting us bound the error after T steps: $\text{err}_T \leq e^{cT}(\alpha_0 + O(1/\sqrt{n}))$, where α_0 is the error of the classifier on the source domain, and n is the number of unlabeled examples in each intermediate domain. While this bound is exponential in T, this bound is non-vacuous for small α_0, and we show that this bound is tight for gradual self-training.

In the second setting, stronger distributional assumptions allow us to do better—we assume that $P(X \mid Y = y)$ is a d-dimensional isotropic Gaussian for each y. Here, we show that if we begin with a classifier w_0 that is nearly Bayes optimal for the initial distribution, we can recover a classifier w_T that is Bayes optimal for the target distribution with infinite unlabeled data. This is an idealized setting to understand what properties of the data might allow self-training to do better than the exponential bound.

Our theory leads to practical insights, showing that regularization—even when we have infinite data—and label sharpening are essential for gradual self-training. Without regularization, the accuracy of gradual self-training drops from 84% to 77% on Portraits and 88% to 46% on rotating MNIST. Even when we self-train with more examples, the accuracy gap between regularized and unregularized models stays the same—unlike in supervised learning where the benefit of regularization diminishes with more examples.

Finally, our theory suggests that the gradual shift structure helps when the shift is small in Wasserstein-infinity distance as opposed to other distance metrics like the KL-divergence. For example, one way to interpolate between the source and target domains is to gradually introduce more images from the target, but this shift is large in Wasserstein-infinity distance—we see experimentally that gradual self-training does not help in this setting. We hope this gives practitioners

some insight into when gradual self-training can work.

2. Setup

Gradually shifting distributions: Consider a binary classification task of predicting labels $y \in \{-1, 1\}$ from input features $x \in \mathbb{R}^d$. We have joint distributions over the inputs and labels, $\mathbb{R}^d \times \{-1, 1\}$: $P_0, P_1, \ldots, P_T$, where P_0 is the source domain, P_T is the target domain, and $P_1, \ldots, P_{T-1}$ are intermediate domains. We assume the shift is gradual: for some $\epsilon > 0$, $\rho(P_t, P_{t+1}) < \epsilon$ for all $0 \le t < T$, where $\rho(P, Q)$ is some distance function between distributions P and Q. We have n_0 labeled examples $S_0 = \{x_i^{(0)}, y_i^{(0)}\}_{i=1}^{n_0}$ sampled independently from the source P_0 and n unlabeled examples $S_t = \{x_i^{(t)}\}_{i=1}^{n}$ sampled independently from P_t for each $1 \le t \le T$.

Models and objectives: We have a model family Θ, where a model $M_\theta : \mathbb{R}^d \to \mathbb{R}$, for each $\theta \in \Theta$, outputs a score representing its confidence that the label y is 1 for the given example. The model's prediction for an input x is $\text{sign}(M_\theta(x))$, where $\text{sign}(r) = 1$ if $r \ge 0$ and $\text{sign}(r) = -1$ if $r < 0$. We evaluate models on the fraction of times they make a wrong prediction, also known as the 0-1 loss:

$$\text{Err}(\theta, P) = \mathop{\mathbb{E}}_{X,Y \sim P}[\text{sign}(M_\theta(X)) \neq Y] \tag{1}$$

The goal is to find a classifier θ that gets high accuracy on the target domain P_T—that is, low $\text{Err}(\theta, P_T)$. In an online setting we may care about the accuracy at the current P_t for every time t, and our analysis works in this setting as well.

Baseline methods: We select a loss function $\ell : \mathbb{R} \times \{-1, 1\} \to \mathbb{R}^+$ which takes a prediction and label, and outputs a non-negative loss value, and we begin by training a source model θ_0 that minimizes the loss on labeled data in the source domain:

$$\theta_0 = \mathop{\arg\min}_{\theta' \in \Theta} \frac{1}{n_0} \sum_{(x_i, y_i) \in S_0} \ell(M_{\theta'}(x_i), y_i) \tag{2}$$

The *non-adaptive baseline* is to use θ_0 on the target domain, which incurs error $\text{Err}(\theta_0, P_T)$. *Self-training* uses unlabeled data to adapt a model. Given a model θ and unlabeled data S, $\text{ST}(\theta, S)$ denotes the output of self-training. Self-training pseudolabels each example in S using M_θ, and then selects a new model θ' that minimizes the loss on this pseudolabeled dataset. Formally,

$$\text{ST}(\theta, S) = \mathop{\arg\min}_{\theta' \in \Theta} \frac{1}{|S|} \sum_{x_i \in S} \ell(M_{\theta'}(x_i), \text{sign}(M_\theta(x_i))) \tag{3}$$

Here, self-training uses "hard" labels: we pseudolabel examples as either -1 or 1, based on the output of the classifier, instead of a probabilistic label based on the model's confidence—we refer to this as *label sharpening*. In our theoretical analysis, we sometimes want to describe the behavior of self-training when run on infinite unlabeled data from a probability distribution P:

$$\text{ST}(\theta, P) = \mathop{\arg\min}_{\theta' \in \Theta} \mathop{\mathbb{E}}_{X \sim P}[\ell(M_{\theta'}(X), \text{sign}(M_\theta(X)))] \tag{4}$$

The *direct adaptation to target* baseline takes the source model θ_0 and self-trains on the target data S_T, and is denoted by $\text{ST}(\theta_0, S_T)$. Prior work often chooses to repeat this process of self-training on the target k times, which we denote by $\text{ST}_k(\theta_0, S_T)$.

Gradual self-training: In gradual self-training, we self-train on the finite unlabeled examples from each domain successively. That is, for $i \ge 1$, we set:

$$\theta_i = \text{ST}(\theta_{i-1}, S_i) \tag{5}$$

Let $\text{ST}(\theta_0, (S_1, \ldots, S_T)) = \theta_T$ denote the output of gradual self-training, which we evaluate on the target distribution P_T. As defined here, gradual self-training uses more data than directly adapting to the target, but we account for this in our theory and experiments.

3. Theory for the margin setting

We show that gradual self-training does better than directly adapting to the target, where we assume that at each time step there exists some Lipschitz classifier—which can be different at each step—that can classify most of the data correctly with a margin (a standard assumption in learning theory), and that the shifts are small. Our main result (Theorem 3.2) bounds the error of gradual self-training. We show that our analysis is tight for gradual self-training (Example 3.4), and explain why regularization, label sharpening, and the ramp loss, are key to our bounds. Proofs are in Appendix A.

3.1. Assumptions

Models: We consider a model family Θ_R where each model M_θ, for $\theta \in \Theta_R$, is R-Lipschitz in the input in ℓ_2 norm for some fixed $R > 0$. That is, for all $x, x' \in \mathbb{R}^d$:

$$|M_\theta(x) - M_\theta(x')| \le R\|x - x'\|_2 \tag{6}$$

An example is the set of regularized linear models that have weights with bounded ℓ_2 norm:

$$\Theta_R^L = \{(w, b) : w \in \mathbb{R}^d, b \in \mathbb{R}, \|w\|_2 \le R\} \tag{7}$$

In this case, given $(w, b) \in \Theta_R^L$, the model's output is $M_{w,b}(x) = w^\top x + b$. Our theory applies to non-linear models that are Lipschitz, but it may help the reader to think of linear models on a first reading.

Losses: We consider margin loss functions such as the hinge and ramp losses. Intuitively, a margin loss encourages a model to classify points correctly and confidently—by keeping correctly classified points far from the decision boundary. We consider the hinge function h and ramp function r:

$$h(m) = \max(1 - m, 0) \tag{8}$$
$$r(m) = \min(h(m), 1) \tag{9}$$

The ramp loss is $\ell_r(\hat{y}, y) = r(y\hat{y})$, where $\hat{y} \in \mathbb{R}$ is a model's prediction, and $y \in \{-1, 1\}$ is the true label. The hinge loss is the standard way to enforce margin, but the ramp loss is more robust towards outliers because it is bounded above—no single point contributes too much to the loss. We will see that the ramp loss is key to the theoretical guarantees for gradual self-training because of its robustness. We denote the population ramp loss as:

$$L_r(\theta, P) = \mathop{\mathbb{E}}_{X,Y\sim P}[\ell_r(M_\theta(X), Y)] \tag{10}$$

Given a finite sample S, the empirical loss is:

$$L_r(\theta, S) = \frac{1}{|S|} \sum_{x,y\in S} \ell_r(M_\theta(x), y) \tag{11}$$

Distributional distance: Our notion of distance is W_∞, the Wasserstein-infinity distance. Intuitively, W_∞ moves points from distribution P to Q by distance at most ϵ to match the distributions. For ease of exposition we consider the Monge form of W_∞, although the results can be extended to the Kantarovich formulation as well. Formally, given probability measures P, Q on $\mathcal{X}$:

$$W_\infty(P, Q) = \inf\{ \sup_{x\in\mathbb{R}^d} ||f(x) - x||_2 : f : \mathbb{R}^d \to \mathbb{R}^d, f_\# P = Q\} \tag{12}$$

As usual, $\#$ denotes the push-forward of a measure, that is, for every set $A \subseteq \mathbb{R}^d$, $f_\# P(A) = P(f^{-1}(A))$.

In our case, we require that the conditional distributions do not shift too much. Given joint probability measures P, Q on the inputs and labels $\mathbb{R}^d \times \{-1, 1\}$, the distance is:

$$\rho(P, Q) = \max(W_\infty(P_{X|Y=1}, Q_{X|Y=1}), W_\infty(P_{X|Y=-1}, Q_{X|Y=-1})). \tag{13}$$

α^*-low-loss assumption: Assume every domain admits a classifier with low loss α^*, that is there exists $\alpha^* \geq 0$ and for every domain P_t, there exists some $\theta_t \in \Theta_R$ with $L_r(\theta_t, P_t) \leq \alpha^*$ (θ_t can be different for each domain).

Gradual shift assumption: For some $\rho < \frac{1}{R}$, assume $\rho(P_t, P_{t+1}) \leq \rho$ for every consecutive domain, where $\frac{1}{R}$ can be interpreted as the regularization strength of the model class Θ_R or as the geometric margin (distance from decision boundary to data) the model is trying to enforce.

Bounded model complexity assumption: For finite sample guarantees, we assume that the Rademacher complexity of the model family, $\mathcal{R}_n(\Theta_R; P)$, is bounded for all distributions $P_0, \ldots, P_T$. That is, for some fixed $B > 0$:

$$\mathcal{R}_n(\Theta_R; P) \leq \frac{B}{\sqrt{n}}, \quad \forall P = P_0, \ldots, P_T \tag{14}$$

where $\mathcal{R}_n(\Theta_R; P)$ is defined as usual as:

$$\mathcal{R}_n(\Theta_R; P) = \mathbb{E}\left[\sup_{\theta\in\Theta_R} \frac{1}{n} \sum_{i=1}^{n} \sigma_i M_\theta(x_i)\right] \tag{15}$$

where the expectation is taken over $x_i \sim P$ and $\sigma_i \sim \text{Uniform}(\{-1, 1\})$ sampled independently for $i = 1, \ldots, n$, so $\sigma_i = -1$ or $\sigma_i = 1$ with equal probability 0.5.

An example is the set of regularized linear models, Θ_R^L, when the data is not too large on average: $\mathbb{E}_{X\sim P}[||X||_2^2] \leq \beta^2$ where $\beta > 0$. In this case a standard result, e.g. Theorem 11, page 82 in (Liang, 2016), is that $\mathcal{R}_n(\Theta_R^L; P) \leq \beta R/\sqrt{n}$, so $B = \beta R$ is the desired bound on the model complexity.

No label shift assumption: Assume that the fraction of $Y = 1$ labels does not change: $P_t(Y)$ is the same for all t.

3.2. Domain shift: baselines fail

While the distribution shift from P_t to P_{t+1} is small, the distribution shift from the source P_0 to the target P_T can be large, as visualized in Figure 2. A classifier that gets 100% accuracy on P_0, might classify every example wrong on P_T, even if $T \geq 2$. In this case, directly adaptating to P_T would not help. The following example formalizes this:

Example 3.1. *Even under the α^*-low-loss, no label shift, gradual shift, and bounded model complexity assumptions, there exists distributions P_0, P_1, P_2 and a source model $\theta \in \Theta_R^L$ that gets 0 loss on the source ($L_r(\theta, P_0) = 0$), but high loss on the target: $L_r(\theta, P_2) = 1$. Self-training directly on the target does not help: $L_r(\text{ST}(\theta, P_2), P_2) = 1$. This holds true even if every domain is separable, so $\alpha^* = 0$.*

Other methods: Our analysis focuses on self-training, but other bounds do not apply in this setting because they either assume that the density ratio between the target and source exists and is not too small (Jiayuan et al., 2006), or that the source and target are similar enough that we cannot discriminate between them (Ben-David et al., 2010).

3.3. Gradual self-training improves error

We show that gradual self-training helps over direct adaptation. For intuition, consider a simple example where $\alpha^* = 0$

and θ_0 classifies every example in P_0 correctly with geometric margin $\gamma = \frac{1}{R}$. If each point shifts by distance $< \gamma$, θ_0 gets every example in the new domain P_1 correct. If we had infinite unlabeled data from P_1, we can learn a model θ' that classifies every example in the new domain P_1 correctly with margin γ since $\alpha^* = 0$. Repeating the process for $P_2, \ldots, P_T$, we get every example in P_T correct.

But what happens when we start with a model that has some error, for example because the data cannot be perfectly separated, and have only finite unlabeled samples? We show that self-training still does better than adapting to the target domain directly, or using the non-adaptive source classifier.

The first main result of the paper says that if we have a model θ that gets low loss and the distribution shifts slightly, self-training gives us a model θ' that does not do too badly on the new distribution.

Theorem 3.2. *Given P, Q with $\rho(P, Q) = \rho < \frac{1}{R}$ and marginals on Y are the same so $P(Y) = Q(Y)$. Assuming Θ_R has bounded model complexity with respect to P and Q, if we have initial model θ, and n unlabeled samples S from Q, and we set $\theta' = \mathrm{ST}(\theta, S)$, then with probability at least $1 - \delta$ over the sampling of S, letting $\alpha^* = \min_{\theta^* \in \Theta_R} L_r(\theta^*, Q)$:*

$$L_r(\theta', Q) \leq \frac{2}{1 - \rho R} L_r(\theta, P) + \alpha^* + \frac{4B + \sqrt{2 \log 2/\delta}}{\sqrt{n}} \tag{16}$$

The proof of this result is in Appendix A, but we give a high level sketch here. There exists some classifier that gets accuracy α^* on Q, so if we had access to n *labeled* examples from Q then empirical risk minimization gives us a classifier that is accurate on the population—from a Rademacher complexity argument we get a classifier θ' with loss at most $\alpha^* + O(B/\sqrt{n})$, the second and third term in the RHS of the bound.

Since we only have *unlabeled* examples from Q, self-training uses θ to pseudolabel these n examples and then trains on this generated dataset. Now, if the distribution shift ρ is small relative to the geometric margin $\gamma = \frac{1}{R}$, then we can show that the original model θ labels most examples in the new distribution Q correctly—that is, $\mathrm{Err}(\theta, Q)$ is small if $L_r(\theta, P)$ is small. Finally, if most examples are labeled correctly we show that because there exists some classifier θ^* with low margin loss, self-training will also learn a classifier θ' with low margin loss $L_r(\theta', Q)$, which completes the proof.

We apply this argument inductively to show that after T time steps, the error of gradual self-training is $\lesssim \exp(cT)\alpha_0$ for some constant c, if the original error is α_0.

Corollary 3.3. *Under the α^*-low-loss, no label shift, gradual shift, and bounded model complexity assumptions, if the source model θ_0 has low loss $\alpha_0 \geq \alpha^*$ on P_0 (i.e. $L_r(\theta_0, P_0) \leq \alpha_0$) and θ is the result of gradual self-training: $\theta = \mathrm{ST}(\theta_0, (S_1, \ldots, S_n))$, letting $\beta = \frac{2}{1 - \rho R}$:*

$$L_r(\theta, P_T) \leq \beta^{T+1}\Big(\alpha_0 + \frac{4B + \sqrt{2 \log 2T/\delta}}{\sqrt{n}}\Big). \tag{17}$$

Corollary 3.3 says that the gradual structure allows some control of the error unlike direct adaptation where the accuracy on the target domain can be 0% if $T \geq 2$. Note that if the classes are separable and we have infinite data, then gradual self-training maintains 0 error.

Our next example shows that our analysis for gradual self-training in this setting is tight—if we start with a model with loss α_0, then the error can in fact increase exponentially even with infinite unlabeled examples. Intuitively, at each step of self-training the loss can increase by a constant factor, which leads to an exponential growth in the error.

Example 3.4. *Even under the α^*-low-loss, no label shift, gradual shift, and bounded model complexity assumptions, given $0 < \alpha_0 \leq \frac{1}{4}$, for every T there exists distributions $P_0, \ldots, P_{2T}$, and $\theta_0 \in \Theta_R^L$ with $L_r(\theta_0, P_0) \leq \alpha_0$, but if $\theta' = \mathrm{ST}(\theta_0, (P_1, \ldots, P_{2T}))$ then $L_r(\theta', P_{2T}) \geq \min(0.5, \frac{1}{2} 2^T \alpha_0)$. Note that L_r is always in $[0, 1]$.*

This suggests that if we want sub-exponential bounds we either need to make additional assumptions on the data distributions, or devise alternative algorithms to achieve better bounds (which we believe is unlikely).

3.4. Essential ingredients for gradual self-training

In this section, we explain why regularization, label sharpening, and the ramp loss are essential to bounding the error of gradual self-training (Theorem 3.2).

Regularization: Without regularization there is no incentive for the model to change when self-training—if we self-train without regularization an optimal thing to do is to output the original model. The intuition is that since the model $\theta = (w, b)$ is used to pseudolabel examples, θ gets every pseudolabeled example correct. The scaled classifier $\theta' = (\alpha w, \alpha b)$ for large α then gets optimal loss, but θ' and θ make the same predictions for every example. We use $\mathrm{ST}'(\theta, S)$ to denote the *set* of possible θ' that minimize the loss on the pseudolabeled distribution (Equation (3)):

Example 3.5. *Given a model $\theta \in \Theta_\infty^L$ (in other words $R = \infty$) and unlabeled examples S where for all $x \in S$, $M_\theta(x) \neq 0$, there exists $\theta' \in \mathrm{ST}'(\theta, S)$ such that for all $x \in \mathbb{R}^d$, $M_\theta(x) = M_{\theta'}(x)$.*

More specific to our setting, our bounds require regularized models because regularized models classify the data cor-

rectly with *a margin*, so even after a mild distribution shift we get most new examples correct. Note that in traditional supervised learning, regularization is usually required when we have few examples for better generalization to the population, whereas in our setting regularization is important for maintaining a margin even with infinite data.

Label sharpening: When self-training, we pseudolabel examples as -1 or 1, based on the output of the classifier. Prior work sometimes uses "soft" labels (Najafi et al., 2019), where for each example they assign a probability of the label being -1 or 1, and train using a logistic loss. The loss on the soft-pseudolabeled distribution is defined as:

$$L_{\sigma,\theta}(\theta') = \mathop{\mathbb{E}}_{X\sim P}[ll(\sigma(M_\theta(X)), \sigma(M_{\theta'}(X)))] \quad (18)$$

, where σ is the sigmoid function, and ll is the log loss:

$$ll(p, p') = p \log p' + (1-p) \log (1-p') \quad (19)$$

Self-training then picks $\theta' \in \Theta_R$ minimizing $L_{\sigma,\theta}(\theta')$. A simple example shows that this form of self-training may never update the parameters because θ minimizes $L_{\sigma,\theta}$:

Example 3.6. *For all Θ_R and $\theta \in \Theta_R$, θ is a minimizer of $L_{\sigma,\theta}$, that is, for all $\theta' \in \Theta_R$, $L_{\sigma,\theta}(\theta) \le L_{\sigma,\theta}(\theta')$.*

This suggests that we "sharpen" the soft labels to encourage the model to update its parameters. Note that this is true even on finite data: set P to be the empirical distribution.

Ramp versus hinge loss: We use the ramp loss, but does the more popular hinge loss L_h work? Unfortunately, the next example shows that we cannot control the error of gradual self-training with the hinge loss even if we had infinite examples, so the ramp loss is important for Theorem 3.2.

Example 3.7. *Even under the α^*-low-loss, no label shift, and gradual shift assumptions, given $\alpha_0 > 0$, there exists distributions P_0, P_1, P_2 and $\theta_0 \in \Theta_R^L$ with $L_h(\theta_0, P_0) \le \alpha$, but if $\theta' = \text{ST}(\theta_0, (P_1, P_2))$ then $L_h(\theta', P_2) \ge Err(\theta', P_2) = 1$ (θ' gets every example in P_2 wrong), where we use the hinge loss in self-training.*

We only analyzed the statistical effects here—the hinge loss tends to work better in practice because it is much easier to *optimize* and is convex for linear models.

3.5. Self-training without domain shift

Example 3.4 showed that when the distribution shifts, the loss of gradual self-training can grow exponentially (though the non-adaptive baseline has unbounded error). Here we show that if we have no distribution shift, the error can only grow linearly: if $P_0 = \ldots = P_T$, given a classifier with loss α_0, if we do gradual self-training the loss is at most $\alpha_0 T$.

Proposition 3.8. *Given $\alpha_0 > 0$, distributions $P_0 = \ldots = P_T$, and model $\theta_0 \in \Theta_R$ with $L_r(\theta_0, P_0) \le \alpha_0$, $L_r(\theta', P_T) \le \alpha_0(T+1)$ where $\theta' = \text{ST}(\theta_0, (P_1, \ldots, P_T))$*

In Appendix A, we show that self-training can indeed hurt without domain shift: given a classifier with loss α on P, self-training on P can increase the classifier's loss on P to 2α, but here the non-adaptive baseline has error α.

4. Theory for the Gaussian setting

In this section we study an idealized Gaussian setting to understand conditions under which self-training can have better than exponential error bounds: we show that if we begin with a good classifier, the distribution shifts are not too large, and we have infinite *unlabeled* data, then gradual self-training maintains a good classifier.

4.1. Setting

We assume $P_t(X \mid Y = y)$ is an isotropic Gaussian in d-dimensions for each $y \in \{-1, 1\}$. We can shift the data to have mean 0, so we suppose:

$$P_t(X|Y = y) = \mathcal{N}(y\mu_t, \sigma_t^2 I) \quad (20)$$

Where $\mu_t \in \mathbb{R}^d$ and $\sigma_t > 0$ for each t. As usual, we assume the shifts are gradual: for some $B > 0$, $\|\mu_{t+1} - \mu_t\|_2 \le \frac{B}{4}$. We assume that the means of the two classes do not get closer than the shift, or else it would be impossible to distinguish between no shift, and the distributions of the two classes swapping: so $\|\mu_t\|_2 \ge B$ for all t. We assume infinite unlabeled data (access to $P_t(X)$) in our analysis.

Given labeled data in the source, we use the objective:

$$L(w, P) = \mathop{\mathbb{E}}_{X,Y\sim P}[\phi(Y(w^\top X))] \quad (21)$$

For unlabeled data, self-training performs descent steps on an underlying objective function (Amini and Gallinari, 2003), which we focus on:

$$U(w, P) = \mathop{\mathbb{E}}_{X\sim P}[\phi(|w^\top X|)] \quad (22)$$

We assume $\phi : \mathbb{R} \to \mathbb{R}^+$ is a continuous, non-increasing function which is strictly decreasing on $[0, 1]$: these are regularity conditions which the hinge, ramp, and logistic losses satisfy. If $w' = \text{ST}(w, P)$ then $U(w', P) \le U(w, P)$ (Amini and Gallinari, 2003).

The algorithm we analyze begins by choosing w_0 from labeled data in P_0, and then updates the parameters with unlabeled data from P_t for $1 \le t \le T$:

$$w_t = \mathop{\arg\min}_{\|w\|_2\le 1, \|w-w_{t-1}\|_2\le \frac{1}{2}} U(w, P_t) \quad (23)$$

Note that we do not show that self-training actually converges to the constrained minimum of U in Equation (23) and prior work only shows that self-training descends on U—we leave this optimization analysis to future work.

4.2. Analysis

Let $w^*(\mu) = \frac{\mu}{\|\mu\|_2}$ where $\|\mu\| \geq B > 0$. Note that $w^*(\mu_t)$ minimizes the 0-1 error on P_t. Our main theorem says that if we start with a regularized classifier w_0 that is near $w^*(\mu_0)$, which we can learn from labeled data, and the distribution shifts $\|\mu_{t+1} - \mu_t\|_2$ are not too large, then we recover the optimal $w_T = w^*(\mu_T)$. The key challenge is that the unlabeled loss U in d dimensions is non-convex, with multiple local minima, so directly minimizing U does not guarantee a solution that minimizes the labeled loss L.

Theorem 4.1. *Assuming the Gaussian setting, if* $\|w_0 - w^*(\mu_0)\|_2 \leq \frac{1}{4}$, *then we recover* $w_T = w^*(\mu_T)$.

Proving this reduces to proving the *single-step* case. At each step $t+1$, if we have a classifier w_t that was close to $w^*(\mu_t)$, then we will recover $w_{t+1} = w^*(\mu_{t+1})$. We give intuition here and the formal proof in Appendix B.

We first show that if μ changes by a small amount, the optimal parameters (for the labeled loss) does not change too much. Then since w_t is close to $w^*(\mu_t)$, w_t is not too far away from $w^*(\mu_{t+1})$. The key step in our argument is showing that the unique minimum of the unlabeled loss $U(w, P_{\mu_{t+1}})$ in the neighborhood of w_t, is $w^*(\mu_t)$—looking for a minimum *nearby* is important because if we deviate too far we might select other "bad" minima. We consider arbitrary w near $w^*(\mu_{t+1})$ and construct a pairing of points (a, b) in $\mathbb{R}^d$, using a convexity argument to show that (a, b) contributes more to the loss of w than $w^*(\mu_{t+1})$.

5. Experiments

Our theory leads to practical insights—we show that regularization and label sharpening are important for gradual self-training, that leveraging the gradual shift structure improves target accuracy, and give intuition for when the gradual shift assumption may not help. We run experiments on three datasets (see Appendix C for more details):

Rotating MNIST: Rotating MNIST is a semi-synthetic dataset where we rotate each MNIST image by an angle between 0 and 60 degrees. We split the 50,000 MNIST training set images into a source domain (images rotated between 0 and 5 degrees), intermediate domain (rotations between 5 and 60 degrees), and a target domain (rotations between 55 degrees and 60 degrees). Note that each image is seen at exactly one angle, so the training procedure cannot track a single image across different angles.

Cover Type: A dataset from the UCI repository where the goal is to predict the forest cover type at a particular location given 54 features (Blackard and Dean, 1999). We sort the examples by increasing distance to water body, splitting the data into a source domain (first 50K examples), intermediate domain (next 400K examples), and a target domain (final 50K examples).

Portraits: A real dataset comprising photos of high school seniors across years (Ginosar et al., 2017). The model's goal is to classify gender. We split the data into a source domain (first 2000 images), intermediate domain (next 14000 images), and target domain (next 2000 images).

In Appendix C we also include synthetic experiments on a mixture of Gaussians dataset which resembles the Gaussian setting of our theory but the covariance matrices are not isotropic, and the number of labeled and unlabeled samples is finite and on the order of the dimension d.

5.1. Leveraging gradual shifts improves adaptation

Our goal is to see if adapting to the gradual shift sequentially helps compared to directly adapting to the target. We evaluate four methods: *Source*: simply train a classifier on the labeled source examples. *Target self-train*: repeatedly self-train on the unlabeled target examples ignoring the intermediate examples. *All self-train*: pool all the unlabeled examples from the intermediate and target domains, and repeatedly self-train on this pooled dataset to adapt the initial source classifier. *Gradual self-train*: sequentially self-train on unlabeled data in each successive intermediate domain, and finally self-train on unlabeled data on the target domain, to adapt the initial source classifier.

For the rotating MNIST datasets, we ensured that the target self-train method sees as many unlabeled target examples as gradual self-train sees across all the intermediate examples. Since Portraits and Cover Type are real datasets we cannot synthesize more examples from the target, so target self-train uses fewer unlabeled examples here. However, the all self-train baseline uses all of the unlabeled examples from all domains.

For rotating MNIST and Portraits we used a 3-layer convolutional network with dropout(0.5) and batchnorm on the last layer, that was able to achieve $97\% - 98\%$ accuracy on held out examples in the source domain. For the CoverType dataset we used a 2 hidden layer feedforward neural network with dropout(0.5) and batchnorm on the last layer which got higher accuracies than logistic regression. For each step of self-training, we filter out the 10% of images where the model's prediction was least confident—Appendix C shows similar findings without this filtering. To account for variance in initialization and optimization, we ran each method 5 times and give 90% confidence intervals. More experimental details are in Appendix C.

Table 1 shows that leveraging the gradual structure leads to improvements over baselines on all 3 datasets, and closes over half the gap between the source and oracle classifiers.

Table 1. Percentage classification accuracies for gradual self-training (ST) and baselines on three datasets, with 90% standard errors for the mean over 5 runs in parantheses. Gradual ST closes about half the gap between the source and oracle classifiers on all three datasets, and does better than self-training directly on the target or self-training on all the unlabeled data pooled together.

	ROT MNIST	COVER TYPE	PORTRAITS
SOURCE MODEL	31.9 (±1.7)	62.8 (±5.0)	75.3 (±1.6)
TARGET ST	+1.0 (±0.5)	+0.7 (±1.5)	+1.6 (±1.2)
ALL ST	+6.1 (±1.1)	+1.5 (±2.2)	+3.5 (±1.8)
GRADUAL ST	**+56.0 (±1.5)**	**+7.6 (±3.7)**	**+8.5 (±0.9)**
ORACLE	+59.6 (±1.2)	+16.8 (±3.7)	+17.0 (±0.8)

5.2. Important ingredients for gradual self-training

Our theory suggests that regularization and label sharpening are important for gradual self-training, because without regularization and label sharpening there is no incentive for the model to change (Section 3.4). However, prior work suggests that overparameterized neural networks trained with stochastic gradient methods have strong implicit regularization (Zhang et al., 2017; Hardt et al., 2016)—in the supervised setting they perform well without explicit regularization even though the number of parameters is much larger than the number of data points—is this implicit regularization enough for gradual self-training?

In our experiments, we see that even without explicit regularization, or with 'soft' probabilistic labels, gradual self-training does slightly better than the non-adaptive source classifier, suggesting that this implicit regularization may have some effect. However, explicit regularization and 'hard' labeling gives a much larger accuracy boost.

Regularization is important: We repeat the same experiment as Section 5.1, comparing gradual self-training with or without regularization—that is, disabling dropout and batchnorm (Ioffe and Szegedy, 2015) in the neural network experiments. In both cases, we first train an *unregularized* model on labeled examples in the source domain. Then, we either turn on regularization during self-training, or keep the model unregularized. We control the original model to be the same in both cases to see if regularization helps in the self-training process, as opposed to in learning a better supervised classifier. Table 2 shows that accuracies are significantly better with regularization, even though unregularized performance is still better than the non-adaptive source classifier.

Soft labeling hurts: We ran the same experiment as Section 5.1, comparing gradual self-training with hard labeling versus using probabilistic labels output by the model. Table 2 shows that accuracies are better with hard labels. Note that in datasets with more intrinsic uncertainty, soft labeling may work well (Mey and Loog, 2016).

Regularization is still important with more data: In supervised learning, the importance of regularization diminishes as we have more training examples—if we had access to infinite data (the population), we don't need regularization. On the other hand, for gradual domain adaptation, the theory says regularization is needed to adapt to the dataset shift even with infinite data, and predicts that regularization remains important even if we increase the sample size.

Table 2. Classification accuracies for gradual self-train with explicit regularization and hard labels (Gradual ST), without regularization but with hard labels (No Reg), and with regularization but with soft labels (Soft Labels). Gradual self-train does best with explicit regularization and hard labels, as our theory suggests, even for neural networks with implicit regularization.

	ROT MNIST	COV TYPE	PORTRAITS
SOFT LABELS	44.1±2.3	63.2±8.5	80.1±1.8
NO REG	45.8±2.5	70.7±1.8	76.5±1.0
GRADUAL ST	**83.8±2.5**	**73.5±1.6**	**82.6±0.8**

To test this hypothesis, we construct a rotating MNIST dataset where we increase the sample sizes. The source domain P_0 consists of $N \in \{2000, 5000, 20000\}$ images on MNIST. P_t then consists of these *same* N images, rotated by angle $3t$, for $0 \leq t \leq 20$. The goal is to get high accuracy on P_{20}: these images rotated by 60 degrees—the model doesn't have to generalize to unseen images, but to seen images at different angles. We compare using regularization versus not using regularization during gradual self-training. Table 3 shows that regularization is still important here, and the gap between regularized and unregularized gradual self-training does not shrink much with more data.

5.3. When does gradual shift help?

Our theory in Section 3 says that gradual self-training works well if the shift between domains is small in Wasserstein-infinity distance, but it may not be enough for the total variation or KL-divergence between P and Q to be small.

To test this, we run an experiment on a modified version of the rotating MNIST dataset. We keep the source and target

Table 3. Classification accuracies for gradual self-train on rotating MNIST as we vary the number of samples. Unlike in previous experiments, here the same N samples are rotated, so the models do not have to generalize to unseen images, but seen images at different angles. The gap between regularized and unregularized gradual self-training does not shrink much with more data.

	N=2000	N=5000	N=20,000
SOURCE	28.3±1.4	29.9±2.5	33.9±2.6
NO REG	55.7±3.9	53.6±4.0	55.1±3.9
REG	**93.1±0.8**	**91.7±2.4**	**87.4±3.1**

domains the same as before, but change the intermediate domains. In Table 1 we saw that gradual self-training works well if we have intermediate images rotated by gradually increasing rotation angles. Another type of gradual transformation is to gradually introduce more examples rotated by 55 to 60 degrees. That is, in the i-th domain, $(20-i)/20$ fraction of the examples are MNIST images rotated by 0 to 5 degrees, and $i/20$ of the examples are MNIST images rotated by 55 to 60 degrees, where $1 \leq i \leq 20$. Here the total-variation distance between successive domains is small, but intuitively the Wasserstein distance is large because each image undergoes a large (≈ 55 degrees) rotation.

As the theory suggests, here gradual self-training does not outperform directly self-training on the target—gradual self-training gets $33.5 \pm 1.5\%$ accuracy on the target, while direct adaptation to the target gets $33.0 \pm 2.2\%$ over 5 runs. Intuitively, gradual self-training helps when most of the distribution shifts by a small amount, and it may not be sufficient if only a small fraction of the distribution shifts but by a large amount. We hope this gives practitioners some insight into when gradual self-training helps.

6. Related work

Self-training is a popular method in semi-supervised learning (Lee, 2013; Sohn et al., 2020) and domain adaptation (Long et al., 2013; Zou et al., 2019; Inoue et al., 2018), and is related to entropy minimization (Grandvalet and Bengio, 2005). Recent work shows that a robust variant of self-training can mitigate the tradeoff between standard and adversarial accuracy (Raghunathan et al., 2020). Related to self-training is co-training (Blum and Mitchell, 1998), which assumes that the input features can be split into two or more views that are conditionally independent on the label. Other theory in semi-supervised learning (Rigollet, 2007; Singh et al., 2008; Ben-David et al., 2008) does not analyze domain shift.

Unsupervised **domain adaptation**, where the goal is to directly adapt from a labeled source domain to an unlabeled target domain, is widely studied (Quiñonero-Candela et al., 2009). The key challenge for domain adaptation is when the source and target domains are very different, when it is easy to discriminate between the two domains and their supports do not overlap (Zhao et al., 2019; Shu et al., 2018), which is typical in the modern high-dimensional regime. *Importance weighting* based methods (Shimodaira, 2000; Sugiyama et al., 2007; Jiayuan et al., 2006) assume the domains are close, with bounds depending on the expected density ratios between the source and target. In practice, even if the domains overlap, the density ratio often scales exponentially in the dimension in which case these methods perform poorly. These methods also assume that $P(Y \mid X)$ is the same for the source and target which we do not require. *The theory of $H\Delta H$-divergence* (Ben-David et al., 2010; Mansour et al., 2009) gives conditions for when a model trained on the source does well on the target *without any adaptation*. Empirical methods aim to learn domain invariant representations (Tzeng et al., 2014; Ganin and Lempitsky, 2015; Tzeng et al., 2017) but there are no theoretical guarantees for these methods (Zhao et al., 2019). These methods require several additional heuristics (Hoffman et al., 2018), and work well on some tasks but not others (Bobu et al., 2018; Peng et al., 2019).

Hoffman et al. (2014); Michael et al. (2018); Markus et al. (2018); Bobu et al. (2018) among others propose approaches for **gradual domain adaptation**. This setting differs from online learning (Shalev-Shwartz, 2007), lifelong learning (Silver et al., 2013), and concept drift (Kramer, 1988; Bartlett, 1992; Bartlett et al., 1996), since we only have unlabeled data from shifted distributions. To the best of our knowledge, we are the first to develop a theory for gradual domain adaptation, and investigate when and why the gradual structure helps.

7. Conclusion and Future Work

Our work suggests that the gradual shift structure, which appears often in applications, enables us to reliably adapt to very different target distributions. There are many exciting avenues for future work:

1. **Better algorithms for gradual shifts**: Can we develop better algorithms and more general theory for gradual domain adaptation?

2. **Discovering gradual shifts**: Can we apply gradual self-training to spatial data: datapoints close in space may be similar? More generally, can we learn which datapoints are similar and do gradual self-training?

3. **Structured domain adaptation**: Are there other structures in real applications we can leverage to reliably adapt to very different target distributions?

Acknowledgements. The authors would like to thank the Open Philantropy Project and the Stanford Graduate Fellowship program for funding. This work is also partially supported by the Stanford Data Science Initiative and the Stanford Artificial Intelligence Laboratory.

We are grateful to the anonymous reviewers, and to Rui Shu, Robin Jia, Pang Wei Koh, Michael Kim, Stephen Mussman, Csaba Szepesvari, Judy Hoffman, Shai Ben-David, Lin Yang, Michael Xie, Aditi Raghunathan, Sanjana Srivastava, Nelson Liu, Rachel Holladay, Megha Srivastava, and Albert Gu for insightful comments.

Reproducibility. All code, data, and experiments can be found on CodaLab at https://bit.ly/gradual-shift-codalab, code is also on GitHub at https://github.com/p-lambda/gradual_domain_adaptation.

References

A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Journal of the American Statistical Association*, -1:320–329, 2012.

A. Bobu, E. Tzeng, J. Hoffman, and T. Darrell. Adapting to continuously shifting domains. In *International Conference on Learning Representations Workshop (ICLR)*, 2018.

A. Farshchian, J. A. Gallego, J. P. Cohen, Y. Bengio, L. E. Miller, and S. A. Solla. Adversarial domain adaptation for stable brain-machine interfaces. In *International Conference on Learning Representations (ICLR)*, 2019.

T. S. Sethi and M. Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, 82:77–99, 2017.

H. Zhao, R. T. des Combes, K. Zhang, and G. J. Gordon. On learning invariant representation for domain adaptation. In *International Conference on Machine Learning (ICML)*, 2019.

J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

R. Shu, H. H. Bui, H. Narui, and S. Ermon. A DIRT-T approach to unsupervised domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2018.

X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *International Conference on Computer Vision (ICCV)*, 2019.

O. Chapelle, A. Zien, and B. Scholkopf. *Semi-Supervised Learning*. MIT Press, 2006.

Q. Xie, M. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. *arXiv*, 2020.

J. Uesato, J. Alayrac, P. Huang, R. Stanforth, A. Fawzi, and P. Kohli. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A. Najafi, S. Maeda, M. Koyama, and T. Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

S. Ginosar, K. Rakelly, S. M. Sachs, B. Yin, C. Lee, P. Krähenbühl, and A. A. Efros. A century of portraits: A visual historical record of american high school yearbooks. *IEEE Transactions on Computational Imaging*, 3, 2017.

Percy Liang. Statistical learning theory. https://web.stanford.edu/class/cs229t/notes.pdf, 2016.

H. Jiayuan, S. A. J., G. Arthur, B. K. M., and S. Bernhard. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2006.

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

M. Amini and P. Gallinari. Semi-supervised learning with explicit misclassification modeling. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. In *Computers and Electronics in Agriculture*, 1999.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, pages 1225–1234, 2016.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

A. Mey and M. Loog. A soft-labeled self-training approach. In *d International Conference on Pattern Recognition*, 2016.

D. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

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

M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang. Confidence regularized self-training. *arXiv preprint arXiv:1908.09822*, 2019.

N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

Y. Grandvalet and Y. Bengio. Entropy regularization. In *Semi-Supervised Learning*, 2005.

A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2020.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Learning Theory (COLT)*, 1998.

P. Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research (JMLR)*, 8:1369–1392, 2007.

A. Singh, R. Nowak, and J. Zhu. Unlabeled data: Now it helps, now it doesn't. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

S. Ben-David, T. Lu, and D. Pal. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *Conference on Learning Theory (COLT)*, 2008.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

M. Sugiyama, M. Krauledat, and K. Muller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)*, 8:985–1005, 2007.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*, 2009.

E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

J. Hoffman, T. Darrell, and K. Saenko. Continuous manifold based adaptation for evolving visual domains. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

G. Michael, E. Dennis, K. B. Mara, B. Peter, and M. Dorit. Gradual domain adaptation for segmenting whole slide images showing pathological variability. In *Image and Signal Processing*, 2018.

W. Markus, B. Alex, and P. Ingmar. Incremental adversarial domain adaptation for continually changing environments. In *International Conference on Robotics and Automation (ICRA)*, 2018.

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

D. L. Silver, Q. Yang, and L. Li. Lifelong machine learning systems: Beyond learning algorithms. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 13, 2013.

A. H. Kramer. Learning despite distribution shift. In *Connectionist Models Summer School*, 1988.

P. L. Bartlett. Learning with a slowly changing distribution. In *Conference on Learning Theory (COLT)*, 1992.

P. L. Bartlett, S. Ben-David, and S. R. Kulkarni. Learning changing concepts by exploiting the structure of change. *Machine Learning*, 41, 1996.

A. Proofs for Section 3

Restatement of Example 3.1. *Even under the α^*-low-loss, no label shift, gradual shift, and bounded model complexity assumptions, there exists distributions P_0, P_1, P_2 and a source model $\theta \in \Theta_R^L$ that gets 0 loss on the source ($L_r(\theta, P_0) = 0$), but high loss on the target: $L_r(\theta, P_2) = 1$. Self-training directly on the target does not help: $L_r(\mathrm{ST}(\theta, P_2), P_2) = 1$. This holds true even if every domain is separable, so $\alpha^* = 0$.*

Proof. We construct an example in 2-D, where we consider the set of regularized linear models Θ_R^L, where $R = 1$. Such a classifier is parametrized by (w, b) where $w \in \mathbb{R}^2$ with $||w||_2 \leq 1$, and $b \in \mathbb{R}$. The output of the model is $M_{w,b}(x) = w^T x + b$, and the predicted label is $\mathrm{sign}(w^T x + b)$.

We first define the source distribution P_0:

$$P_0(X = (1, 1) \wedge Y = 1) = 0.5 \tag{24}$$

$$P_0(X = (-1, -1) \wedge Y = -1) = 0.5 \tag{25}$$

Consider the source classifier $w_0 = (0, 1)$. The classifier classifies all examples correctly, in particular $\mathrm{sign}(w_0^T(1, 1)) = 1$, and $\mathrm{sign}(w_0^T(-1, -1)) = -1$. In addition, the ramp loss is 0, that is:

$$\mathop{\mathbb{E}}_{X,Y \sim P_0}[r(Y(w_0^T X))] = 0 \tag{26}$$

We now construct distributions P_1 and P_2:

$$P_1(X = (1, 1/3) \wedge Y = 1) = 0.5 \tag{27}$$

$$P_1(X = (-1, -1/3) \wedge Y = -1) = 0.5 \tag{28}$$

$$P_2(X = (1, -1/3) \wedge Y = 1) = 0.5 \tag{29}$$

$$P_2(X = (-1, 1/3) \wedge Y = -1) = 0.5 \tag{30}$$

Basically, the second-coordinate starts at 1 and decreases over time when the label is $Y = 1$, and starts at -1 and increases over time when the label is $Y = -1$. We note that $\rho(P_0, P_1) = \rho(P_1, P_2) = \frac{2}{3} \leq \frac{1}{R}$.

Now, w_0, b_0 classifies everything incorrectly in P_2. $\mathrm{sign}(w_0^T(1, -1/3)) = -1$, and $\mathrm{sign}(w_0^T(-1, 1/3)) = 1$ but the corresponding labels in P_2 are 1 and -1 respectively. Accordingly, the ramp loss $L_r(M_{w_0,b_0}, P_2) = 1$.

Self-training on P_2 cannot fix the problem. w_0, b_0 gets every example incorrect, so all the pseudolabels are incorrect. In particular, let Y' be the pseudolabels produced using w_0, b_0—we have, $Y' \mid [X = (1, -1/3)] = -1$ and $Y' \mid [X = (-1, 1/3)] = 1$. Self-training on this is now a convex optimization problem, which attains 0 loss, for example using the classifier $w' = (-1, 0)$, $b' = 0$, but any such classifier also gets all the examples incorrect. Note that the max-margin classifier on the source also exhibits the same issue (that is, it can get all the examples wrong after the dataset shift), from a simple extension of this example.

Finally, the classifier $w^* = (1, 0)$, $b^* = 0$, gets every label correct in *all distributions*, P_0, P_1, P_2.

□

Restatement of Theorem 3.2. *Given P, Q with $\rho(P, Q) = \rho < \frac{1}{R}$ and marginals on Y are the same so $P(Y) = Q(Y)$. Assuming Θ_R has bounded model complexity with respect to P and Q, if we have initial model θ, and n unlabeled samples S from Q, and we set $\theta' = \mathrm{ST}(\theta, S)$, then with probability at least $1 - \delta$ over the sampling of S, letting $\alpha^* = \min_{\theta^* \in \Theta_R} L_r(\theta^*, Q)$:*

$$L_r(\theta', Q) \leq \frac{2}{1 - \rho R} L_r(\theta, P) + \alpha^* + \frac{4B + \sqrt{2 \log 2/\delta}}{\sqrt{n}} \tag{31}$$

We begin by stating and proving some lemmas that formalize the proof outline in the main paper. We begin with a standard lemma that says if a model family Θ_R has bounded Rademacher complexity, and we learn a model from n labeled examples from a distribution P, then the classifier is almost as good as the optimal classifier in Θ_R on P, and the classifier gets closer to optimal as n increases.

Lemma A.1. *Given n samples S from a joint distribution P over inputs $\mathbb{R}^d$ and labels $\{-1, 1\}$, and suppose $\mathcal{R}_n(\Theta_R; P) \leq B/\sqrt{n}$. Let $\hat{f}$ and f be the empirical and population minimizers of the ramp loss respectively:*

$$\hat{f} = \mathop{\arg\min}_{f \in \Theta_R} L_r(f, S) \tag{32}$$

$$f^* = \mathop{\arg\min}_{f \in \Theta_R} L_r(f, P) \tag{33}$$

Then with probability at least $1 - \delta$,

$$L_r(\hat{f}) - L_r(f^*) \leq \frac{4B + \sqrt{2 \log 2/\delta}}{\sqrt{n}} \tag{34}$$

Proof. We begin with a standard bound (see e.g. Theorem 9, page 70 in (Liang, 2016)), where the generalization error on the left is bounded by the Rademacher complexity:

$$L_r(\hat{f}) - L_r(f^*) \leq 4\mathcal{R}_n(A; P) + \sqrt{\frac{2 \log 2/\delta}{n}} \tag{35}$$

Here, $A = \{(x, y) \mapsto \ell_r(M_\theta(x), y) : \theta \in \Theta_R\}$ is the composition of the loss with the model family, and $\mathcal{R}_n$ is the Rademacher complexity. It now suffices to bound the $\mathcal{R}_n$ term.

We first use Talagrand's lemma, which says that if $\phi : \mathbb{R} \to \mathbb{R}^+$ is an L-Lipschitz function (that is, $|\phi(b) - \phi(a)| \leq L|b-a|$ for all a, b), then:

$$\mathcal{R}_n(\phi \circ F; P) \leq L\mathcal{R}_n(F; P) \tag{36}$$

In our case, we let $F = \{(x, y) \mapsto yM_\theta(x) : \theta \in \Theta_R\}$, in which case $A = r \circ F$ where r is the ramp loss. The Lipschitz constant of the ramp loss r is 1, so $\mathcal{R}_n(A; P) \leq \mathcal{R}_n(F; P)$.

This gives us the desired result since from a simple calculation we can show $\mathcal{R}_n(F; P) = \mathcal{R}_n(\Theta_R; P)$ (the y cancels out since it gets multiplied by a Rademacher random variable which is -1 and 1 with equal probability), so we have:

$$\mathcal{R}_n(F; P) \leq \frac{B}{\sqrt{n}} \tag{37}$$

□

The next lemma shows that the error (0-1 loss) of M_θ is low on Q, even though the margin loss may be high. Intuitively, M_θ classifies most points in P correctly with geoemtric margin $\frac{1}{R}$, so after a small distribution shift $< \frac{1}{R}$, these points are still correctly classified since the margin acts as a 'buffer' protecting us from misclassification.

Lemma A.2. *If $\theta \in \Theta_R$, $\rho(P, Q) = \rho < \frac{1}{R}$, and the marginals on Y are the same so $P(Y) = Q(Y)$, then $Err(M_\theta, Q) \leq \frac{2}{1-\rho R} L_r(M_\theta, P)$*

Proof. Intuitively, if the ramp loss for an R-Lipschitz model is low, then most points are classified correctly with high geometric margin (distance to decision boundary). Formally, we first show (using basically Markov's inequality) that $P(YM_\theta(X) \leq \rho R) \leq \frac{1}{1-\rho R} L_r(\theta, P)$, where we recall that $r : \mathbb{R} \to [0, 1]$ is the ramp loss which is bounded between 0 and 1:

$$\begin{aligned}
L_r(\theta, P) &= \mathop{\mathbb{E}}_{X,Y\sim P}[r(YM_\theta(X))] \\
&\geq \mathop{\mathbb{E}}_{X,Y\sim P}[r(YM_\theta(X))\mathbb{I}_{YM_\theta(X)\leq\rho R}] \\
&\geq \mathop{\mathbb{E}}_{X,Y\sim P}[(1-\rho R)\mathbb{I}_{YM_\theta(X)\leq\rho R}] \\
&= (1-\rho R)P(YM_\theta(X) \leq \rho R)
\end{aligned}$$

Here, the inequality on the third line follows because if $YM_\theta(X) \leq \rho R$ where $0 < \rho R \leq 1$, then $r(YM_\theta(X)) \geq 1 - \rho R$, from the definition of the ramp loss.

This gives us:

$$P(YM_\theta(X) \leq \rho R) \leq \frac{1}{1-\rho R} L_r(\theta, P) \tag{38}$$

The high level intuition of the next step is that since the shift is small, only points x, y with $yM_\theta(x) \leq \rho R$ can be misclassified after the distribution shift, and from the previous step since there aren't too many of these the error of θ on Q is small.

Formally, fix $\epsilon > 0$ with $\rho + \epsilon < \frac{1}{R}$, and let $f_y : \mathbb{R}^d \to \mathbb{R}^d$ be a mapping such that for all measurable $A \subseteq \mathbb{R}^d$, $P(f^{-1}(A) \mid Y = y) = Q(A|Y = y)$, with $\sup_{x\in\mathbb{R}^d} ||f_y(x) - x||_2 \leq \rho + \epsilon$ for $y \in \{-1, 1\}$[1], then we have:

$$\begin{aligned}
&\text{Err}(\theta, Q) \\
&= Q(Y \neq \text{sign}(M_\theta(X))) \\
&= Q(YM_\theta(X) \leq 0) \\
&= Q(Y = 1)Q(M_\theta(X) \leq 0 \mid Y = 1) + \\
&\quad Q(Y = -1)Q(M_\theta(X) \geq 0 \mid Y = -1) \\
&= P(Y = 1)P(M_\theta(f_1(X)) + b \leq 0 \mid Y = 1) + \\
&\quad P(Y = -1)P(M_\theta(f_{-1}(X)) + b \geq 0 \mid Y = -1) \\
&\leq P(Y = 1)P(M_\theta(X) \leq (\rho + \epsilon)R \mid Y = 1) + \\
&\quad P(Y = -1)P(M_\theta(X) \geq -(\rho + \epsilon)R \mid Y = -1) \\
&= P(YM_\theta(X) \leq (\rho + \epsilon)R)
\end{aligned}$$

Where the inequality follows since M_θ is R-Lipschitz:

$$\begin{aligned}
|M_\theta(X) - M_\theta(f_y(X))| &\leq R||X - f_y(X)||_2 \\
&\leq R(\rho + \epsilon)
\end{aligned}$$

Combining this with Equation (38), this gives us:

$$\text{Err}(\theta, Q) \leq \frac{1}{1 - (\rho + \epsilon)R} L_r(\theta, P) \tag{39}$$

Since $\epsilon > 0$ was arbitrary, by taking the infimum over all $\epsilon > 0$, we get:

$$\text{Err}(\theta, Q) \leq \frac{1}{1 - \rho R} L_r(\theta, P) \tag{40}$$

Which was what we wanted to show.

□

From the previous lemma, M_θ has low error on Q, or in other words only occasionally mislabels examples from Q. The next lemma says that if we minimize the ramp loss on a distribution where the points are only occasionally mislabeled, then we learn a classifier with low (good) ramp loss as well.

Lemma A.3. *Given random variables X, Y, Y' (defined on the same measure space) with joint distribution P, where X denotes the distribution over inputs, and Y, Y' denote*

[1] We need the ϵ here because a mapping with exactly the W_∞ distance may not exist, although if P and Q have densities then such a mapping does exist.

distinct distributions over labels. If $P(Y \neq Y') \leq \beta$ then for any θ, $L_r(\theta, P_X P_{Y'|X}) \leq L_r(\theta, P_X P_{Y|X}) + \beta$. Here $P_X P_{Y|X}$ denotes the distribution where the input X is sampled from P_X and then the label is sampled from $P_{Y|X}$.

Proof. The proof is by algebra, where we recall that $r : \mathbb{R} \to [0, 1]$ is the ramp loss which is bounded between 0 and 1:

$$\begin{aligned}
&L_r(\theta, P_X P_{Y'|X}) \\
&= \mathbb{E}\Big[r\big(Y' M_\theta(X)\big)\Big] \\
&= \mathbb{E}\Big[r\big(Y' M_\theta(X)\big)\mathbb{I}_{Y=Y'}\Big] + \\
&\quad \mathbb{E}\Big[r\big(Y' M_\theta(X)\big)\mathbb{I}_{Y\neq Y'}\Big] \\
&\leq \mathbb{E}\Big[r\big(Y' M_\theta(X)\big)\mathbb{I}_{Y=Y'}\Big] + \mathbb{E}\Big[\mathbb{I}_{Y\neq Y'}\Big] \\
&= \mathbb{E}\Big[r\big(Y' M_\theta(X)\big)\mathbb{I}_{Y=Y'}\Big] + \beta \\
&= \mathbb{E}\Big[r\big(Y M_\theta(X)\big)\mathbb{I}_{Y=Y'}\Big] + \beta \\
&\leq \mathbb{E}\Big[r\big(Y M_\theta(X)\big)\Big] + \beta \\
&= L_r(\theta, P_X P_{Y|X}) + \beta
\end{aligned}$$

□

Proof of Theorem 3.2. We begin by noting that there is some $\theta^* \in \Theta_R$ that gets low loss α^* on Q:

$$L_r(M_{\theta^*}, Q) = \alpha^* = \min_{\theta^* \in \Theta_R} L_r(M_{\theta^*}, Q) \tag{41}$$

In self-training, we do not have access to labels from Q so we use M_θ to pseudolabel examples X from Q, so let $w, b = \theta$ and let $Y' \mid X = \text{sign}(M_\theta(X))$ be the pseudolabel distribution $Q_{Y'|X}$.

However, our pseudolabels are mostly correct. That is, let $\beta = \frac{2}{1-\rho R} L_r(M_\theta, P)$. Since the conditions of Lemma A.2 are satisfied, $\text{Err}(M_\theta, Q) \leq \beta$. This means that the pseudolabels from M_θ and the true labels on Q mostly agree: $Q(Y \neq Y') \leq \beta$. So by Lemma A.3, θ^*, which attained low loss α^* on Q, also does fairly well on the pseudolabeled distribution $Q_X Q_{Y'|X}$, which denotes the distribution where the input X is sampled from Q_X and then the label is sampled from $Q_{Y'|X}$:

$$\begin{aligned}
L_r(M_{\theta^*}, Q_X Q_{Y'|X}) &\leq L_r(M_{\theta^*}, Q) + \beta \\
&\leq \alpha^* + \beta
\end{aligned} \tag{42}$$

Since we have n examples from $Q_X Q_{Y'|X}$, from Lemma A.1 the empirical risk minimizer θ' on the n examples satisfies:

$$\begin{aligned}
L_r(\theta', Q_X Q_{Y'|X}) \leq \min_{\theta \in \Theta_R} L_r(\theta, Q_X Q_{Y'|X}) \\
+ \frac{4B + \sqrt{2 \log 2/\delta}}{\sqrt{n}}
\end{aligned} \tag{43}$$

But minimizing the loss on $Q_X Q_{Y'|X}$ explicitly gives us a lower loss than θ^* gets on $Q_X Q_{Y'|X}$ (recall that θ^* is the minimizer of the loss on Q which is different):

$$\begin{aligned}
\min_{\theta \in \Theta_R} L_r(\theta, Q_X Q_{Y'|X}) &\leq L_r(M_{\theta^*}, Q_X Q_{Y'|X}) \\
&\leq \alpha^* + \beta
\end{aligned} \tag{44}$$

Combining Equations (43) and (44), we get:

$$L_r(\theta', Q_X Q_{Y'|X}) \leq \alpha^* + \beta + \frac{4B + \sqrt{2 \log 2/\delta}}{\sqrt{n}} \tag{45}$$

This bounds the ramp loss of θ' on the pseudolabeled distribution $Q_X Q_{Y'|X}$—to convert this back to Q we apply Lemma A.3 again which we can since $Q(Y \neq Y') \leq \beta$, which gives us:

$$L_r(\theta', Q) \leq \alpha^* + 2\beta + \frac{4B + \sqrt{2 \log 2/\delta}}{\sqrt{n}} \tag{46}$$

This completes the proof.

□

Restatement of Corollary 3.3. *Under the α^*-low-loss, no label shift, gradual shift, and bounded model complexity assumptions, if the source model θ_0 has low loss $\alpha_0 \geq \alpha^*$ on P_0 (i.e. $L_r(\theta_0, P_0) \leq \alpha_0$) and θ is the result of gradual self-training: $\theta = \text{ST}(\theta_0, (S_1, \ldots, S_n))$, letting $\beta = \frac{2}{1-\rho R}$:*

$$L_r(\theta, P_T) \leq \beta^{T+1}\Big(\alpha_0 + \frac{4B + \sqrt{2 \log 2T/\delta}}{\sqrt{n}}\Big). \tag{47}$$

Proof. We begin with a classifier with loss α_0. Applying Theorem 3.2 for each subsequent step of self-training, letting $\beta = \frac{2}{1-\rho R}$, we get:

$$\begin{aligned}
L_r(M_{\theta_{i+1}}, P_{i+1}) \leq &\beta L_r(M_{\theta_i}, P_i) + \alpha^* \\
&+ \frac{4B + \sqrt{2 \log 2T/\delta}}{\sqrt{n}}
\end{aligned} \tag{48}$$

Expanding, this becomes the sum of a geometric series. Noting that $\alpha^* \leq \alpha_0$, by using the formula for the sum of geometric series, we get:

$$L_r(M_{\theta_T}, P_T) \leq \beta^{T+1}\Big(\alpha_0 + \frac{4B + \sqrt{2 \log 2T/\delta}}{\sqrt{n}}\Big) \tag{49}$$

□

Restatement of Example 3.4. *Even under the α^*-low-loss, no label shift, gradual shift, and bounded model complexity assumptions, given $0 < \alpha_0 \le \frac{1}{4}$, for every T there exists distributions $P_0, \ldots, P_{2T}$, and $\theta_0 \in \Theta_R^L$ with $L_r(\theta_0, P_0) \le \alpha_0$, but if $\theta' = \mathrm{ST}(\theta_0, (P_1, \ldots, P_{2T}))$ then $L_r(\theta', P_{2T}) \ge \min(0.5, \frac{1}{2}2^T\alpha_0)$. Note that L_r is always in $[0, 1]$.*

Proof. The construction works even in 1-D. We will consider regularized linear models Θ_R^L with $R = 1$, so $\rho < \frac{1}{R}$. Such a model in 1D can be parametrized by 2 parameters, $w, b \in \mathbb{R}$ with $|w| \le 1$, where the output of the linear model for an input $x \in \mathbb{R}$ is $wx + b$, and the label is $\mathrm{sign}(wx + b)$.

First we give intuition, and then we dive into the formal details of the construction.

We start with a classifier $\theta_0 = (w_0, b_0) = (1, 0)$. We will construct the distributions so that the classifier $\theta_t = \theta_0$ for all t, that is, gradual self-training will not update the classifier. In the initial distribution P_0, all the negative examples will be located at $x = -10$, so the classifier gets them correct and incurs 0 loss on them. α_0 fraction of the positive examples will be at $x = -0.1$, these examples are misclassified so the classifier incurs loss α_0. The rest of the positive examples will be at $x = 1$, and the classifier incurs 0 loss on them.

In distribution P_1, $0.5\alpha_0$ fraction of the positive examples will move from $x = 1$ to $x = 0.5$, but everything else stays the same as in P_0. After pseudolabeling and self-training, the classifier still stays the same, that is $\theta_1 = \theta_0$. This is because the α_0 fraction of examples at $x = -0.1$ will be pseudolabeled negative, the $0.5\alpha_0$ fraction of examples at $x = 0.5$ pseudolabeled positive, and the remaining positive examples at $x = 1$ will be pseudolabeled positive. Training on this pseudolabeled distribution gives us $\theta_1 = (w_1, b_1) = (1, 0)$ as the optimal parameters.

In P_2, the $0.5\alpha_0$ fracton of points at $x = 0.5$ moves to $x = -0.1$. After pseudolabeling and self-training, we still get $\theta_2 = \theta_0$. At this point the classifier incurs loss $1.5\alpha_0$. We repeat this process, except for P_3, α_0 fraction of the positive examples move from $x = 1$ to $x = 0.5$, and then the next time in P_5, $2\alpha_0$ fraction of the positive examples move from $x = 1$ to $x = 0.5$, etc. So in this way the loss grows exponentially.

We now give the formal construction, which works even in just 1 dimension. First, we choose S to be the maximum integer such that $(2^{S-1} + \frac{1}{2})\alpha_0 < \frac{1}{2}$. We have $S \ge 1$, because $(2^{1-1} + \frac{1}{2})\alpha_0 = \frac{3}{2}\alpha_0 \le \frac{3}{2}\frac{1}{4} < \frac{1}{2}$.

We now define a sequence of weights, which represents the fraction of points we move in each step as in the sketch above. For $0 \le i \le S - 1$, let $w_i = \frac{1}{2}2^i\alpha_0$, and let $w_S = \frac{1}{2} - (2^{S-1} + \frac{1}{2})\alpha_0$. From the sum of geometric series, we can verify that each of these weights are positive, and the weights sum up to $\frac{1}{2}$.

We now define the distributions at each step, we case on whether the step is odd or even since as in the above high level sketch, it takes 2 steps to move a point from $x = 1$ across to the other side of the decision boundary. One subtlety is that unlike the sketch above, since we use the Monge form of the Wasserstein distance, we cannot have all the points exactly at $x = 1$ but keep them separated by a small distance $\delta = \frac{1}{10S}$. This is a technical detail, so on a first reading the reader may just pretend $\delta = 0$ to work through the structure of the proof.

(Odd case) For $0 \le t < \min(T, S + 1)$, P_{2t+1} is given by:

$$P_{2t+1}(x = -10 \wedge Y = -1) = 0.5 \tag{50}$$

$$P_{2t+1}(x = -0.1 \wedge Y = 1) = \alpha_0 + \sum_{i=0}^{t-1} w_i \tag{51}$$

$$P_{2t+1}(x = 0.5 \wedge Y = 1) = w_t \tag{52}$$

$$P_{2t+1}(x = 1 + i\delta \wedge Y = 1) = w_i \quad \forall t < i \le S \tag{53}$$

(Even case) For $0 \le t \le \min(T, S + 1)$, P_{2t} is given by:

$$P_{2t}(x = -10 \wedge Y = -1) = 0.5 \tag{54}$$

$$P_{2t}(x = -0.1 \wedge Y = 1) = \alpha_0 + \sum_{i=0}^{t-1} w_i \tag{55}$$

$$P_{2t}(x = 1 + i\delta \wedge Y = 1) = w_i \quad \forall t \le i \le S \tag{56}$$

If $T \ge S+1$, then for $2S+2 \le i \le 2T$, we set $P_i = P_{2S+2}$ (by this step the classifier will have reached ramp loss and error 0.5).

We can check that if $t < 2S$, then the classifier obtained from gradual self-training is $w_t = (1, 0)$ (the classifier does not change after self-training). When $t = 2S$, $w_t = (1, -0.9)$, and finally if $2S < t$ then $w_t = (1, -1.5)$. The edge case is because at the end all positive points are to the left of the classifier, so the classifier moves to the right.

Next we examine the loss values. If $t \le S$, the fraction of examples the classifier w_{2t} gets wrong on P_{2t} is:

$$\alpha_0 + \sum_{i=0}^{t-1} w_i = (2^{t-1} + \frac{1}{2})\alpha_0 \ge \frac{1}{2}2^t\alpha_0 \tag{57}$$

If $t > S$, the fraction of examples the classifier w_{2t} gets wrong on P_{2t} is 0.5. The ramp loss is bounded below by the error rate, which means:

$$L_r(\theta', P_{2T}) \ge \min(0.5, \frac{1}{2}2^T\alpha_0) \tag{58}$$

As desired.

We can verify that for every i, $W_\infty(P_i, P_{i+1}) \le 0.6 < 1 = \frac{1}{R}$, so these distributions satisfy the gradual shift assumption.

$P_i(Y=1) = P_i(Y=-1) = 0.5$ for all i, so the distributions satisfy the no label shift assumption. The classifier $(w, b) = (1, 5)$ gets 0 loss on all P_i, so the distributions satisfy the α^*-low-loss assumption with $\alpha^* = 0$. Finally, the data is all bounded in a constant region, between $x = -10$ and $x = 2$, so the distributions satisfy the bounded model complexity assumption. □

Restatement of Example 3.5. *Given a model $\theta \in \Theta^L_\infty$ (in other words $R = \infty$) and unlabeled examples S where for all $x \in S$, $M_\theta(x) \neq 0$, there exists $\theta' \in \text{ST}'(\theta, S)$ such that for all $x \in \mathbb{R}^d$, $M_\theta(x) = M_{\theta'}(x)$.*

Proof. The proof is straightforward: scaling up the parameters of the original model θ gives us a θ' that gets 0 loss (ramp or hinge) on the pseudolabeled distribution, but does not change the model predictions. For simplicity, we focus on the ramp loss but the proof applies to the hinge loss as well. Suppose $\theta = (w, b)$, where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

We choose our new parameters to be $\theta' = (\alpha w, \alpha b)$, where $\alpha \geq 1$ is a scaling factor we will choose. Then we can write $L(\theta')$, the loss of θ' on the pseudolabeled examples S as:

$$\begin{aligned} L(\theta') &= \frac{1}{|S|} \sum_{x \in S} \ell_r(M_{\theta'}(x), \text{sign}(M_\theta(x))) \\ &= \frac{1}{|S|} \sum_{x \in S} r(\text{sign}(w^T x + b)(\alpha w^T x + \alpha b)) \\ &= \frac{1}{|S|} \sum_{x \in S} r(\alpha |w^T x + b|) \end{aligned} \tag{59}$$

Now, we can choose large enough α so that the term inside the r in the last line above is always ≥ 1:

$$\alpha = \frac{1}{\min_{x \in S} |w^T x + b|} \tag{60}$$

So now, $|(w')^T x + b'| = \alpha |(w^T x + b)| \geq 1$ for all $x \in S$. This gives us that $L(\theta') = 0$, since $r(m) = 0$ for $m \geq 1$. Note that this is true for the hinge loss as well, $h(m) = 0$ for $m \geq 1$. Since L is bounded below by 0, θ' is a minimizer of the loss on the pseudolabeled distribution (which is what self-training minimizes, see Equation (3)).

Since θ' is just a scaled up version of θ, it does not change the predictions:

$$\text{sign}(\alpha w^T x + \alpha b) = \text{sign}(w^T x + b) \tag{61}$$

□

Restatement of Example 3.6. *For all Θ_R and $\theta \in \Theta_R$, θ is a minimizer of $L_{\sigma,\theta}$, that is, for all $\theta' \in \Theta_R$, $L_{\sigma,\theta}(\theta) \leq L_{\sigma,\theta}(\theta')$.*

Proof. The reason for this is that the logistic loss is a proper scoring loss—if we fix p, the loss of $ll(p, p')$ is minimized when $p' = p$. That is, if $0 \leq p, p' \leq 1$:

$$ll(p, p) \leq ll(p, p') \tag{62}$$

So we have:

$$\begin{aligned} L_{\sigma,\theta}(\theta') &= \mathbb{E}[ll(\sigma(M_\theta(X)), \sigma(M_{\theta'}(X)))] \\ &\geq \mathbb{E}[ll(\sigma(M_\theta(X)), \sigma(M_\theta(X)))] \\ &= L_{\sigma,\theta}(\theta) \end{aligned} \tag{63}$$

□

Restatement of Example 3.7. *Even under the α^*-low-loss, no label shift, and gradual shift assumptions, given $\alpha_0 > 0$, there exists distributions P_0, P_1, P_2 and $\theta_0 \in \Theta^L_R$ with $L_h(\theta_0, P_0) \leq \alpha$, but if $\theta' = \text{ST}(\theta_0, (P_1, P_2))$ then $L_h(\theta', P_2) \geq Err(\theta', P_2) = 1$ (θ' gets every example in P_2 wrong), where we use the hinge loss in self-training.*

Proof. We construct an example in 2D. We consider the set of regularized linear models Θ^L_R, where $R = 1$. Such a classifier is parametrized by (w, b) where $w \in \mathbb{R}^2$ with $||w||_2 \leq 1$, and $b \in \mathbb{R}$. The output of the model is $M_{w,b}(x) = w^T x + b$, and the predicted label is $\text{sign}(w^T x + b)$.

Set $\alpha_0 = \min(\frac{1}{2}, \frac{2\alpha}{3})$. We will construct an example where the initial hinge error is $\leq \alpha_0$, but it increases to over 1 and gets every example wrong, in 2 distribution shifts, even though there exists a single classifier with 0 hinge loss across all the distributions. Let $w_0 = (1, 0)$ and $b_0 = 0$. Consider a distribution Q_δ, for $\delta \in \mathbb{R}$, defined as follows:

$$Q_\delta(Y = 1 \wedge X = (\delta, 1)) = \frac{1 - \alpha_0}{2} \quad \text{[Point 1]}$$

$$Q_\delta(Y = 1 \wedge X = (-\frac{1}{2}, \frac{1 - \alpha_0}{\alpha_0})) = \frac{\alpha_0}{2} \quad \text{[Point 2]}$$

$$Q_\delta(Y = -1 \wedge X = (-\delta, -1)) = \frac{1 - \alpha_0}{2} \quad \text{[Point 3]}$$

$$Q_\delta(Y = -1 \wedge X = (\frac{1}{2}, -\frac{1 - \alpha_0}{\alpha_0})) = \frac{\alpha_0}{2} \quad \text{[Point 4]}$$

We will set $P_0 = Q_1$, $P_1 = Q_{1/3}$, and $P_2 = Q_{-1/3}$. First, we note that the Wasserstein-infinity distance between any consecutive one of these is at most $2/3 < 1$.

Next, we can verify that $L_h(w_0, P_0) = \frac{3}{2}\alpha_0 \leq \alpha$. In particular, w_0 gets points 2 and 4 incorrect, and points 1 and 3 correct with margin 1. Computing the expectation of the loss, we get $\frac{3}{2}\alpha_0$.

Now the algorithm self-trains on P_1: w_0 pseudolabels points 1 and 4 positive ($y = 1$), and pseudolabels points 2 and 3 negative ($y = -1$), again getting points 2 and 4 incorrect. From the KKT conditions, we can verify that the minimizer

of the hinge loss on these pseudolabeled points is $w_1 = w_0$, and $b_2 = 0$.

Finally, the algorithm self-trains on P_2: here w_0 pseudolabels points 3 and 4 positive, and 1 and 2 negative. That is, it gets all the examples wrong. Self-training on these pseudolabels, the model still gets every example wrong (one solution is $w_2 = (0, -1)$ and $b_2 = 0$). So $\text{Err}(w_2, P_2) = 1$, and the hinge loss is lower bounded by the error with $L_h(w_2, P_2) \geq \text{Err}(w_2, P_2)$.

On the other hand, the classifier $w^* = (0, 1)$ and $b^* = 0$, gets hinge loss 0 on P_1, P_2, P_3.

□

Restatement of Proposition 3.8. *Given $\alpha_0 > 0$, distributions $P_0 = \ldots = P_T$, and model $\theta_0 \in \Theta_R$ with $L_r(\theta_0, P_0) \leq \alpha_0$, $L_r(\theta', P_T) \leq \alpha_0(T + 1)$ where $\theta' = \text{ST}(\theta_0, (P_1, \ldots, P_T))$*

We give intuition for our argument, and then dive into the formal proof. Suppose we start out with a model that has ramp loss α_0 on $P = P_0 = \cdots = P_T$. After a single step of self-training, the loss can increase to $2\alpha_0$ on P. So a naive argument leads to an exponential bound (since the loss is now $2\alpha_0$, it can increase to $2 \cdot 2\alpha_0$ after another round of self-training, etc, so after T steps the loss on P is bounded by $2^T\alpha_0$). Showing a linear upper bound requires a more subtle argument that tracks some other invariants, and not just the loss value.

Roughly speaking, if the initial loss is below α_0, there cannot be more than α_0 fraction of points near the decision boundary. We show that this invariant is maintained by self-training: the 'number' of points near the decision boundary decreases, so it always stays below the initial value α_0. Finally, we show that if there are α_0 points near the decision boundary, then self-training cannot increase the loss by more than α_0 *no matter what the current loss is*. This shows that at each step the loss can only increase by α_0. Compare this with Example 3.4, where we do have distribution shift—in this case the 'number' of points near the decision boundary can keep increasing which can lead to an exponential growth in the loss.

We now dive into the formal proof—we begin by making some definitions and stating and proving lemmas that formalize the above intuition.

In self-training, we pseudolabel an example x with label $\text{sign}(M_\theta(x))$. We define the corresponding distribution on the pseudolabels $P_{Y|x,\theta}$ by $Y \mid x, \theta = \text{sign}(M_\theta(x))$.

Recall that the loss of θ on labeled data is (where r is the ramp loss):

$$\begin{aligned} L_r(\theta, P) &= \mathop{\mathbb{E}}_{X,Y\sim P}[\ell_r(M_\theta(X), Y)] \\ &= \mathop{\mathbb{E}}_{X,Y\sim P}[r(YM_\theta(X))] \end{aligned} \tag{64}$$

We define a loss on unlabeled data which corresponds to the loss of θ if every example was labeled by M_θ. This roughly corresponds to the 'number' of points near the decision boundary, since points far from the decision boundary incur 0 loss, but points near the decision boundary incur a loss between 0 and 1. Note that the unlabeled loss does not use the labels Y. Letting P_X denote the marginal distribution of P on X, and $P_X P_{Y|X}$ denote the distribution where X is sampled from P_X and Y is sampled from $P_{Y|X}$, the unlabeled loss U_r is:

$$\begin{aligned} U_r(\theta, P) &= L_r(\theta, P_X P_{Y|X,\theta}) \\ &= \mathop{\mathbb{E}}_{X\sim P}[\ell_r(M_\theta(X), \text{sign}(M_\theta(X)))] \\ &= \mathop{\mathbb{E}}_{X\sim P}[r(|M_\theta(X)|)] \end{aligned} \tag{65}$$

The unlabeled loss U_r and labeled loss L_r are always defined since the ramp loss is bounded below by 0. A straightforward lemma shows that the unlabeled loss lower bounds the labeled loss.

Lemma A.4 (Lower bounds labeled loss)**.** *The unlabeled loss lower bounds the labeled loss: $U_r(\theta, P) \leq L_r(\theta, P)$.*

Proof. Since $Y \in \{-1, 1\}$,

$$|M_\theta(X)| = |YM_\theta(X)| \geq YM_\theta(X) \tag{66}$$

Now, since r is a non-increasing function, we have:

$$r(|M_\theta(X)|) \leq r(YM_\theta(X)) \tag{67}$$

Taking expectations on both sides:

$$U_r(\theta, P) \leq L_r(\theta, P) \tag{68}$$

□

The next lemma shows that each step of self-training decreases the unlabeled loss.

Lemma A.5 (Unlabeled loss decreases)**.** *If $\theta, \theta' \in \Theta$ and $\theta' = \text{ST}(\theta, P)$, then $U_r(\theta', P) \leq U_r(\theta, P)$.*

Proof. Since the unlabeled loss does not depend on the labels, we have:

$$U_r(\theta', P) = U_r(\theta', P_X P_{Y|X,\theta}) \tag{69}$$

From Lemma A.4, the unlabeled loss lower bounds the labeled loss:

$$U_r(\theta', P_X P_{Y|X,\theta}) \le L_r(\theta', P_X P_{Y|X,\theta}) \tag{70}$$

But $P_X P_{Y|X,\theta}$ is the distribution of pseudolabels produced by model θ, which is exactly what self-training ($\theta' = \text{ST}(\theta, P)$) minimizes (recall the definition of self-training in Equation (4)), so θ' has lower loss than θ on the pseudolabeled distribution:

$$L_r(\theta', P_X P_{Y|X,\theta}) \le L_r(\theta, P_X P_{Y|X,\theta}) = U_r(\theta, P) \tag{71}$$

Which means that:

$$U_r(\theta', P) \le U_r(\theta, P) \tag{72}$$

□

We now show a type of triangle inequality for the loss, which says that the loss of θ' on P is upper bounded by the loss of θ' on pseudolabels from θ plus the loss of θ on P.

Lemma A.6 (Triangle Inequality)**.** $L_r(\theta', P) \le L_r(\theta', P_X P_{Y|X,\theta}) + L_r(\theta, P)$

Proof. We will first show that for any x, y, θ, θ':

$$\ell_r(M_{\theta'}(x), y) \le \max(\ell_r(M_{\theta'}(x), \text{sign}(M_\theta(x))), \ell_r(M_\theta(x), y)) \tag{73}$$

We can prove this by casing. If $M_{\theta'}(x)$ and $M_\theta(x)$ have different signs, or $M_\theta(x)$ and y have different signs, then the RHS is 1. But the ramp loss is bounded above by 1, so the LHS has loss at most 1, which makes this statement true. Otherwise, suppose $M_{\theta'}(x)$, $M_\theta(x)$, and y all have the same signs—but then $\text{sign}(M_\theta(x)) = y$, so $\ell_r(M_{\theta'}(x), y) = \ell_r(M_{\theta'}(x), \text{sign}(M_\theta(x)))$.

With this in hand, the result follows with some algebra:

$$\begin{aligned}
&L_r(\theta', P) \\
=&\mathbb{E}[\ell_r(M_{\theta'}(X), Y)] \\
\le&\mathbb{E}[\max(\ell_r(M_{\theta'}(X), \text{sign}(M_\theta(X))), \ell_r(M_\theta(X), Y))] \\
\le&\mathbb{E}[\ell_r(M_{\theta'}(X), \text{sign}(M_\theta(X))) + \ell_r(M_\theta(X), Y)] \\
=&L_r(\theta', P_X P_{Y|X,\theta}) + L_r(\theta, P)
\end{aligned} \tag{74}$$

□

Next, we show that if the unlabeled loss of θ is less than α, then self-training cannot increase the loss by more than α.

Lemma A.7 (Upper bounding loss growth)**.** *Suppose $U_r(\theta, P) \le \alpha$ and let $\theta' = \text{ST}(\theta, P)$. Then:*

$$L_r(\theta', P) \le L_r(\theta, P) + \alpha$$

Proof. From Lemma A.6, it suffices to show that $L_r(\theta', P_X P_{Y|X,\theta}) \le \alpha$. But as in Equation (71), this is simply because θ' minimizes the pseudolabeled loss so we have:

$$L_r(\theta', P_X P_{Y|X,\theta}) \le U_r(\theta, P) \le \alpha \tag{75}$$

□

The proof of Proposition 3.8 now simply inductively applies Lemma A.5 and Lemma A.7.

Proof of Proposition 3.8. Let $\theta_t = \text{ST}(\theta_{t-1}, P_t)$ for $1 \le t \le T$. The unlabeled loss lower bounds the labeled loss: that is, since $L_r(\theta_0, P_0) \le \alpha_0$, from Lemma A.4, $U_r(\theta_0, P_0) \le \alpha_0$. The unlabeled loss can only decrease with self-training: that is, inductively applying Lemma A.5, we get that for all t, $U_r(\theta_t, P_t) \le \alpha_0$. Then from Lemma A.7, the loss can only increase by α_0 at each step of self-training, so $L_r(\theta_T, P_T) \le L_r(\theta_0, P_0) + \alpha_0 T \le \alpha_0(T+1)$. □

The next Example shows that even without distribution shift, self-training can increase the loss of a model from α_0 to nearly $2\alpha_0$.

Example A.8. *Even under the α^*-low-loss and bounded model complexity assumptions, for every $0.25 > \alpha_0 > \epsilon > 0$, there exists a model θ_0 and distribution P with $L_r(\theta_0, P) \le \alpha_0$ but $L_r(\text{ST}(\theta_0, P), P) \ge 2\alpha_0 - \epsilon$.*

Proof. We give an example in 1D, where a linear model can be parametrized by 2 parameters, $w, b \in \mathbb{R}$ with $|w| \le 1$, where the output of the linear model for an input $x \in \mathbb{R}$ is $wx + b$, and the label is $\text{sign}(wx + b)$.

Let $\delta = \epsilon/3$ and $a = \alpha_0/(1+\delta)$. Let the data distribution P be given by:

$$P(X = -10 \wedge Y = -1) = 0.5 \tag{76}$$

$$P(X = 0 \wedge Y = 1) = a \tag{77}$$

$$P(X = 1 \wedge Y = 1) = a - \delta \tag{78}$$

$$P(X = 10 \wedge Y = 1) = 0.5 - 2a + \delta \tag{79}$$

Note that the probabilities are all non-negative and add up to 1 and the data is bounded between $x = -10$ and $x = 10$.

Let the initial model be $w_0 = 1$ and $b_0 = -\delta$. The initial loss is $L_r((w_0, b_0), P) = a + (a - \delta)\delta = \alpha_0 - \delta^2 \le \alpha_0$. We can check that after self-training, the updated parameters are $w_1 = 1$ and $b_1 = 1$. The final loss is $L_r((w_1, b_1), P) = 2a - \delta \ge 2\alpha_0(1 - \delta) - \delta \ge 2\alpha_0 - 3\delta = 2\alpha_0 - \epsilon$.

□

B. Proofs for Section 4

We prove Theorem 4.1 in Section 3, following the sketch described in the paper. Our first lemma shows that if μ does not change too much, then the optimal parameters $w^*(\mu)$ do not change too much either.

Lemma B.1. *w^* is $\frac{1}{B}$-Lipschitz, that is if $||\mu||_2, ||\mu'||_2 \geq B > 0$, then:*

$$||w^*(\mu') - w^*(\mu)||_2 \leq \frac{1}{B}||\mu' - \mu||_2 \tag{80}$$

Proof. Recall that $w^*(\mu) = \mu/||\mu||_2$, which is well defined since $||\mu||_2 > 0$. We will first prove that if $||v'||_2 \geq ||v||_2 = 1$, then the claim holds, that is:

$$||\frac{v'}{||v'||_2} - v||_2^2 \leq ||v' - v||_2^2 \tag{81}$$

Expanding both sides, this is equivalent to showing:

$$1 + ||v||_2^2 - \frac{2v^T v'}{||v'||_2} \leq ||v'||_2^2 + ||v||_2^2 - 2v^T v \tag{82}$$

Subtracting both sides by $||v||_2^2$, it suffices to show:

$$1 - \frac{2v^T v'}{||v'||_2} \leq ||v'||_2^2 - 2v^T v \tag{83}$$

But since $||v'|| \geq 1$, we can bound the LHS above if we multiply by $||v'||_2$:

$$\begin{aligned} 1 - \frac{2v^T v'}{||v'||_2} &\leq ||v'||_2 - 2v^T v' \\ &\leq ||v'||_2^2 - 2v^T v' \end{aligned} \tag{84}$$

So Equation (81) is true.

Now we prove the main claim. Without loss of generality, suppose $||\mu'||_2 \geq ||\mu||_2$, otherwise we can swap μ and μ'. Then we can scale μ' and reduce to the previous case:

$$\begin{aligned} ||w^*(\mu') - w^*(\mu)||_2 &= ||\frac{\mu'}{||\mu'||_2} - \frac{\mu}{||\mu||_2}||_2 \\ &= ||\frac{\mu'/||\mu||_2}{||\mu'||_2/||\mu||_2} - \frac{\mu}{||\mu||_2}||_2 \\ &= ||\frac{(\mu'/||\mu||_2)}{||(\mu'/||\mu||_2)||_2} - \frac{\mu}{||\mu||_2}||_2 \\ &\leq ||\frac{\mu'}{||\mu||_2} - \frac{\mu}{||\mu||_2}||_2 \\ &= \frac{1}{||\mu||_2}||\mu' - \mu||_2 \\ &\leq \frac{1}{B}||\mu' - \mu||_2 \end{aligned} \tag{85}$$

Where in the inequality on the 4th line we applied Equation (81). This completes the proof. □

We now state a standard lemma in measure theory, which says that if $f(x) \geq g(x)$ for all x, and the inequality is *strict* on a set of non-zero measure (volume), then the integral of f is strictly greater than the integral of g.

Lemma B.2. *Let μ be a measure on $\mathbb{R}^d$, and C be measurable with $\mu(C) > 0$. Suppose $f(x) > g(x)$ if $x \in C$, and $f(x) \geq g(x)$ for all $x \in \mathbb{R}^d$, where f and g are measurable functions with finite integrals. Then:*

$$\int_{\mathbb{R}^d} f(X)d\mu > \int_{\mathbb{R}^d} g(X)d\mu \tag{86}$$

Our next lemma is the key step of the proof. We show that $w^*(\mu)$ is a strict local minimizer of $U(w, P_{\mu,\sigma})$, that is it has lower loss than any other w nearby.

Lemma B.3. *For all $w \in \mathbb{R}^d$ with $||w||_2 \leq 1$ and $||w - w^*(\mu)||_2 < 1$, with $w \neq w^*(\mu)$, we have:*

$$U(w^*(\mu), P_{\mu,\sigma}) < U(w, P_{\mu,\sigma}) \tag{87}$$

Proof. Denote $w^*(\mu)$ as w^*. By Cauchy-Schwarz, since $\|w^*\|_2 = 1$ and $\|w - w^*\|_2 < 1$, we have $w \cdot w^* > 0$, and $\|w\|_2 > 0$. This is because $w \cdot w^* = \|w^*\|_2 + (w - w^*) \cdot w^* \geq 1 - \|w - w^*\|_2\|w^*\|_2 > 0$. Since the dot product is non-zero, neither vector can be 0.

We begin by noting that $U(w, P_{\mu,\sigma})$ is well-defined and finite: because $\phi(|X|)$ is between 0 and 1 so the expectation is well-defined with finite, non-negative value.

Step 1 (Scaling Parameters): First, we show that scaling up the parameters decreases the loss: for any w and $\lambda > 1$, $U(\lambda w, P_{\mu,\sigma}) < U(w, P_{\mu,\sigma})$.

Since ϕ is non-increasing, $\phi(|\lambda w^T x|) \geq \phi(|w^T x|)$. Now, let $C = \{x \in \mathbb{R}^d : 0 < \lambda w^T x < 1\}$. Since ϕ is strictly decreasing on $[0, 1]$, for $x \in C$, $\phi(|\lambda w^T x|) < \phi(|w^T x|)$. $P_{\mu,\sigma}(C) > 0$ (the Gaussian mixture distribution assigns positive probability to any set with non-zero volume / Lebesgue measure). Then from Lemma B.2:

$$\mathop{\mathbb{E}}_{X \sim P_{\mu,\sigma}}[\phi(|\lambda w^T X|)] < \mathop{\mathbb{E}}_{X \sim P_{\mu,\sigma}}[\phi(|w^T X|)] \tag{88}$$

Which is precisely saying $U(\lambda w, P_{\mu,\sigma}) < U(w, P_{\mu,\sigma})$.

This lets us assume, without loss of generality, that $||w||_2 = 1$ since scaling up w strictly decreases the loss, and the theorem statement assumes $||w||_2 \leq 1$.

Step 2 (Rotating parameters): Note that rotating the entire space does not change the loss values, formally if A is a rotation matrix then:

$$U(Aw, P_{A\mu,\sigma}) = U(w, P_{\mu,\sigma}) \tag{89}$$

So without loss of generality, we rotate the setup so that w and w^* lie on the XY plane (except for the first two

coordinates, all coordinates are 0). Let v be the unit bisector of w and w^*, given by $v = (w + w^*)/||w + w^*||_2$. Without loss of generality, rotate the setup so that v is along the positive Y axis (the second coordinate is 1, and all other coordinates are 0), and the first two coordinates of w^* are positive. Let $\mu = (r, s, 0)$ where $0 \in \mathbb{R}^{d-2}$, we then have that $r, s > 0$ since w^* and μ are in the same direction.

Step 3 (Symmetry argument): Now consider any point $u = (x, y, z)$ with $z \in \mathbb{R}^{d-2}$, with $x, y > 0$. Consider its reflection point around v, $u' = (-x, y, z)$. Let $\Delta(w, w', u) = \phi(|(w')^T u|) - \phi(|w^T u|)$ denote the increase in loss on x from using classifier w' instead of w. Now, from the way we constructed u', $w^T u = (w^*)^T u'$, and $(w^*)^T u = w^T u'$. So $\Delta(w, w^*, u) = -\Delta(w^*, w, u')$. That is, as per our sketch, the loss for u decreases when using w^* instead of w, but increases for u' when using w^* instead of w, but the magnitudes of these two quantities are equal.

Next, we will show that the probability density is higher for u than u'. Let p denote the density of $P_{\mu,\sigma}$. $P_{\mu,\sigma}$ is the mixture of two Gaussians, so for normalizing constant $k > 0$, we have:

$$p(u) = k\Big[\exp\Big(-\frac{1}{2\sigma}((r-x)^2 + (s-y)^2 + z^2)\Big) + \exp\Big(-\frac{1}{2\sigma}((r+x)^2 + (s+y)^2 + z^2)\Big)\Big] \tag{90}$$

$$p(u') = k\Big[\exp\Big(-\frac{1}{2\sigma}((r+x)^2 + (s-y)^2 + z^2)\Big) + \exp\Big(-\frac{1}{2\sigma}((r-x)^2 + (s+y)^2 + z^2)\Big)\Big] \tag{91}$$

We now use strict convexity of $\exp(-x)$ to show that $p(u) > p(u')$. Let $a = (r-x)^2 + (s-y)^2 + z^2$, $b = (r+x)^2 + (s+y)^2 + z^2$, $c = (r+x)^2 - (r-x)^2 = 4rx$. Since, $x, y, r, s > 0$, we have $0 < a < b$ and $0 < c < b - a$. Letting $f(x) = \exp(-x/(2\sigma))$ we can rewrite the above probabilities as:

$$p(u) = k\Big[f(a) + f(b)\Big] \tag{92}$$

$$p(u') = k\Big[f(a+c) + f(b-c)\Big] \tag{93}$$

Finally, we use strict convexity of $f(x) = \exp(-x)$ to show the desired result. Since $a < a + c < a + b$, for some $\alpha \in (0, 1)$, we can write:

$$a + c = \alpha a + (1 - \alpha)b \tag{94}$$

$$b - c = (1 - \alpha)a + \alpha b \tag{95}$$

Then, from strict convexity, we have:

$$f(a + c) < \alpha f(a) + (1 - \alpha)f(b) \tag{96}$$

$$f(b - c) < (1 - \alpha)f(a) + \alpha f(b) \tag{97}$$

Adding both of these, we get:

$$f(a + c) + f(b - c) < f(a) + f(b) \tag{98}$$

That is, we have shown $p(u) > p(u')$.

The case when $u = (-x, -y, z)$, where $z \in \mathbb{R}^{d-2}$ and $x, y > 0$ is symmetric. We ignore points $\{(x, y, z) : x = 0 \vee y = 0\}$ since this has measure 0.

Step 4 (Expectation): We give intuition and then dive into the math. For every pair of points in our pairing in Step 3, the contribution to the loss of w^* is at most as high as the contribution to the loss of w. So this trivially gives us $L(w^*, P_{\mu,\sigma}) \leq L(w, P_{\mu,\sigma})$, but we want a strict inequality. However, we can find a set of points with non-zero volume (Lebesgue measure) where the contribution to the loss for w^* is strictly less than for w, which completes the proof.

Formally, letting $S_+ = \{(x, y, z) : z \in \mathbb{R}^{d-2}, x > 0, y > 0\}$, we can write (where we defined Δ in Step 3):

$$\begin{aligned} &L(w, P_{\mu,\sigma}) - L(w^*, P_{\mu,\sigma}) \\ &= 2\int_{S_+} \big[p(u)\Delta(w^*, w, u) + p(u')\Delta(w^*, w, u')\big] \end{aligned} \tag{99}$$

Where the 2 comes from the fact that the case when $x, y < 0$ is symmetric and gives the same integral. Now, let $C = \{(x, y, z) : x > 0, y > 0, x^2 + y^2 \leq 1, z \in \mathbb{R}^{d-2}, w^*\}$ be a quarter cylinder. The volume of C is > 0, and $C \subseteq S_+$. Further, for all $x \in C$, we have:

$$p(u)\Delta(w^*, w, u) + p(u')\Delta(w^*, w, u') > 0 \tag{100}$$

So applying Lemma B.2 again, we get:

$$L(w, P_{\mu,\sigma}) - L(w^*, P_{\mu,\sigma}) > 0 \tag{101}$$

Which completes the proof. □

With these key lemmas, the proof of Theorem 4.1 is straightforward.

Restatement of Theorem 4.1. *Assuming the Gaussian setting, if $\|w_0 - w^*(\mu_0)\|_2 \leq \frac{1}{4}$, then we recover $w_T = w^*(\mu_T)$.*

Proof. The proof reduces to showing the one-step case: for $0 < t \leq T$, if $\|w_{t-1} - w^*(\mu_{t-1})\|_2 \leq \frac{1}{4}$ then $w_t = w^*(\mu_t)$, where the w_t is selected according to Equation (23). Applying this one-step result inductively gives us the desired result, that $w_T = w^*(\mu_T)$.

For the one-step case, from Lemma B.1, since $||\mu_{t-1}||_2, ||\mu_t||_2 \geq B > 0$, $||w^*(\mu_t) - w^*(\mu_{t-1})||_2 \leq \frac{1}{B}||\mu_t - \mu_{t-1}||_2 \leq \frac{1}{B}\frac{B}{4} = \frac{1}{4}$. Then by triangle inequality, since $||w_{t-1} - w^*(\mu_{t-1})||_2 \leq \frac{1}{4}$, we have

$||w_{t-1} - w^*(\mu_t)||_2 \leq \frac{1}{2}$. Further, $||w^*(\mu_t)||_2 \leq 1$, and by Lemma B.3, any other w satisfying $||w_{t-1} - w||_2 \leq \frac{1}{2} < 1$, $||w||_2 \leq 1$, and $w \neq w^*(\mu_t)$ satisfies $U(w^*(\mu_t), P_{\mu_t,\sigma_t}) < U(w, P_{\mu_t,\sigma_t})$. So $w^*(\mu_t)$ is the unique minimizer in the constrained set, which means $w_t = w^*(\mu_t)$. □

C. Experimental details for Section 5

In this section, we provide additional experimental details, show results on a synthetic Gaussian dataset, and give results for ablations for the experiments in Section 5.1. An advantage of gradual self-training is that it has a very small number of hyperparameters and we show that our findings are robust to different choices of these parameters—even if we do not do confidence thresholding, train every method for more iterations, and use a smaller window size, gradual self-training does better than self-training directly to the target and the other baselines. For reproducibility, we provide all code but we also describe our datasets and models here.

C.1. Datasets

We ran experiments on 4 datasets:

1. *Gaussian in $d = 100$ dimensions*: We randomly select an initial mean and covariance for each of the two classes, and a final mean and covariance for each class, all in d dimensions. Note that unlike in the theory in Section 4, each class can have a different (non diagonal) covariance matrix. The initial and final covariance matrices can also be different. The marginal probability of each class is the same, 0.5. We get labeled source data sampled from the gaussian with the initial mean and covariance for each class. For the intermediate domains, we linearly interpolate the means and covariances for each class between the initial and final, and sample points from a gaussian with the corresponding mean and covariance matrices. The number of labeled and unlabeled examples is on the order of d (as opposed to exponential in d, which importance weighting approaches would need). We provide more details next.

   Details: We sample $\mu_0^{(-1)}, \mu_0^{(+1)}, \mu_T^{(-1)}, \mu_T^{(+1)}$ independently from $\mathcal{N}(0, I)$ in d dimensions. Since d is high, these are all nearly orthogonal to each other. We then sample covariance matrices $\Sigma_0^{(-1)}, \Sigma_0^{(+1)}, \Sigma_T^{(-1)}, \Sigma_T^{(+1)}$ independently by sampling a diagonal matrix and rotation matrix (since the covariance matrices are PSD they decompose into $UDU^\top$ for rotation matrix U and diagonal matrix D). We first sample a diagonal matrix D in d dimensions where each entry is uniformly random and independently sampled between min_var$= 0.05$ and max_var$= 0.1$. Then, we sample a rotation matrix U from the Haar distribution (which is a standard way to sample random orthogonal matrices), and then compose these to get $UDU^\top$.

   At all times, we keep $\mathbb{P}(Y = +1) = \mathbb{P}(Y = -1) = 0.5$. We now sample $N = 500$ labeled examples from the source domain, where $\mathbb{P}(X|Y =$

$1) = \mathcal{N}(\mu_0^{(+1)}, \Sigma_0^{(+1)})$ and $\mathbb{P}(X|Y = -1) = \mathcal{N}(\mu_0^{(-1)}, \Sigma_0^{(-1)})$. We sample $T = 5000$ unlabeled intermediate examples. For $y \in \{-1, 1\}$, let $\mu_t^{(y)} = (t/T)\mu_0^{(y)} + ((T-t)/T)\mu_T^{(y)}$ and $\Sigma_t^{(y)} = (t/T)\Sigma_0^{(y)} + ((T-t)/T)\Sigma_T^{(y)}$. We then sample $y_t \sim \text{Bern}(0.5)$, and $x_t \sim \mathcal{N}(\mu_t^{(y)}, \Sigma_t^{(y)})$—the model only gets to see x_t but not y_t. The unseen target images are sampled from the final means and covariances for each class, and we measure accuracy on these held out examples.

We use $N = 500$ (500 labeled examples from the source), $T = 5000$ (so 5000 unlabeled examples in total), and use min_var=0.05, max_var=0.1 (the standard-deviation is the square root of these). We sample 1000 target examples to check accuracy. To ensure that the benefits of gradual self-training are not merely because it has access to more data, self-training directly to the target gets $T = 5000$ unlabeled examples from the target, which is the same total number of unlabeled examples gradual self-training uses.

2. *Rotating MNIST*: We shuffle the MNIST training data and normalize the values to the range $[0, 1]$ by dividing by 255. We then split the dataset, using the first $N_{\text{src}} = 5000$ images as the source training set, next $N_{\text{val}} = 1000$ images as source validation set, next $N_{\text{inter}} = 42000$ images as unlabeled intermediate examples, and the next $N_{\text{trg}} = 2000$ images as unseen target examples. We use the first 1000 images in the target to evaluate target accuracies, holding out the next 1000 for future work. We rotate each source image by an angle uniformly selected between 0 and 5 degrees. The i-th intermediate example is rotated by angle $5 + 55i/N_{\text{inter}}$ degrees. Each target image is rotated by an angle uniformly selected between 55 degrees and 60 degrees. To ensure that the benefits of gradual self-training are not merely because it has access to more data, self-training directly to the target gets $T = 42000$ unlabeled examples from the target (rotated by an angle uniformly selected between 55 degrees and 60 degrees), which is the same total number of unlabeled examples gradual self-training uses. The oracle classifier is trained on the last 2000 intermediate examples, but with labels.

3. *Cover Type*: A more realistic dataset from the UCI repository where the goal is to predict the forest cover type at a particular location given 54 features (Blackard and Dean, 1999). We normalize each feature across the dataset to have mean 0 and standard deviation 1. We convert the problem into a binary classification task by keeping the first two out of six cover types in the dataset: spruce/fir and lodgepole pine. These classes comprise the majority of the dataset: 495,141 out of 581,012 examples. We sort the examples by increasing distance to water body, splitting the data into a source domain (first 50K examples), intermediate domain (next 400K examples), and a target domain (next 45K examples), leaving out the final 141 examples to get a nice round number. We shuffle the source data, using 10K examples as training, and 40K as a source validation set. We shuffle the target dataset and use 25K examples to evaluate all methods, holding out the other 20K examples for future experiments. The target self-training method sees the last 50K unlabeled examples in the intermediate domain, while self-training on all and gradual self-training use all 400K unlabeled examples in the intermediate domain. The oracle classifier is trained on the last 50K intermediate examples, but with labels.

4. *Portraits*: A more realistic dataset where we do not control the structure of the shift, consisting of photos of high school seniors taken across many years. Additionally, there is label shift, that is the proportions of males and females, $\mathbb{P}(Y)$, changes over time (see Figure 3), unlike our theory which assumes that the probability of each label stays constant. We use the first 2000 images as source images. We shuffle these, and use 1000 for training, and 1000 for validation. We use the next 14000 images as unlabeled intermediate examples. Finally, we use the next 2000 images as unseen target examples, shuffling them and using the first 1000 to evaluate our methods, and holding out the other 1000 for future experiments. We downsample the images to 32x32 and normalize the values to the range $[0, 1]$ by dividing by 255 but do no other preprocessing. We reserve images at the end of the dataset as held-out examples for future work, and so that we can test how the method extrapolates past the point we validate on. The target self-training method sees the last 2K unlabeled examples in the intermediate domain (which are closest in time to the target domain), while self-training on all and gradual self-training use all 14K unlabeled examples in the intermediate domain. The oracle classifier is trained on the last 2K intermediate examples, but with labels.

C.2. Algorithm and baselines

Next, we describe the gradual self-training algorithm and parameters in more detail. Algorithm 1 shows pseudocode for gradual self-training. filter_low_confidence filters out the α fraction of examples where the model is least confident, where confidence is measured as the maximum of the softmax output of the classifier. This filtering is standard in many instances of self-training (Xie et al., 2020).

For the baselines—for target self-train, we self-train mul-

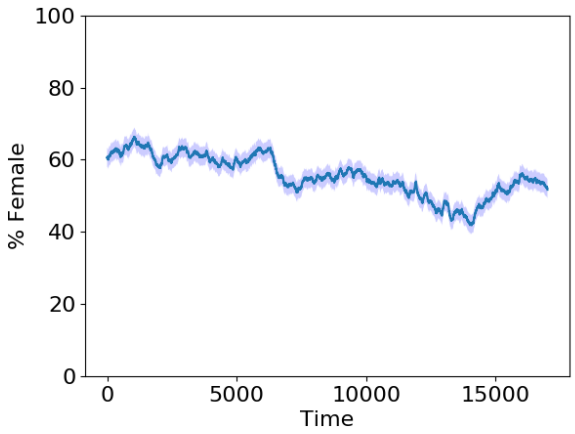


Figure 3. The plot shows a rolling average of the fraction of images that are female, over a window size of 1000, with 90% confidence intervals. The plot suggests that the proportion of males and females changes over time, and is not constant—this label shift might make the task more challenging for self-training methods.

Algorithm 1 Gradual Self-Training

Input: Labeled source examples S, Intermediate unlabeled examples I, Window size W, Confidence threshold $\alpha \in (0, 1)$, Number of Epochs n, Regularized model M
Assume: W divides $|I|$
Train M on S for n epochs
for $t = 1$ **to** $|I|/W$ **do**
  cur_xs= $I[(t-1)W : tW]$
  pseudolabeled_ys= M.predict_labels(cur_xs)
  confident_idxs=filter_low_confidence(M, cur_xs, α)
  filtered_xs = cur_xs[confident_idxs]
  filtered_ys = pseudolabeled_ys[confident_idxs]
  Train M on filtered_xs, filtered_ys for n epochs
end for

tiple times (iteratively) on the target. Each round of self-training uses the current model M to pseudolabel examples in the target, and then trains on these pseudolabeled examples. Specifically, to make comparisons fair we self-train $|I|/W$ times on the target, so that the total number of self-training steps performed by the target self-train baseline and gradual self-training are the same. Similarly, when we self-train to all examples, we self-train multiple times on all unlabeled data, self-training $|I|/W$ times. Here W is the window size in Algorithm 1 which is the number of examples in each intermediate domain.

Note that in the synthetic datasets (rotating MNIST and Gaussian) we ensure that target self-train gets access to the same number of unlabeled examples as gradual self-training does in total, to ensure that the improvements are not simply because gradual self-training consumes more unlabeled data (accumulated over all of the intermediate domains). For the real datasets (Cover Type and Portraits), we cannot generate additional examples for target self-train. However, this is why we also compare against self-training directly to all the unlabeled data, which gets access to exactly the same data that gradual self-training does but does not leverage the gradual structure.

C.3. Models and parameter settings

Next, we describe the models and parameter settings we used:

1. *Models*: For the Gaussian dataset we use a logistic regression classifier, with l2 regularization 0.02. For the MNIST and Portraits dataset, we use a 3 layer convolutional network with ReLU activations. For each conv layer we use a filter size of 5x5, stride of 2x2, 32 output channels, and relu activation. We added dropout(0.5) after the final conv layer, and batchnorm after dropout. We flatten the final layer, and then apply a single linear layer to output logits (the number of logits is the number of classes in the dataset which is 10 for rotating MNIST and 2 for Portraits). We then take the softmax of the logits, and optimize the cross-entropy loss. We did not tune the model architecture for our experiments, however we checked that adding an extra layer, changing the number of output channels, and using a different architecture with an extra fully connected layer on top, have little impact on the results. For the Cover Type dataset, we used a 2 hidden layer fully connected feedforward neural network, with ReLU activations, with 32 nodes in each of the two hidden layers. We then apply a single linear layer to output logits. We added dropout(0.5) after the final hidden layer, and batchnorm after dropout. We take the softmax of the logits, and optimize the cross-entropy loss. This model performed better than logistic regression on held out examples from the source.

2. *Parameters*: For the window size, we use $W = 500$ for the Gaussian dataset, and $W = 2000$ for the rotating MNIST and Portraits dataset, and $W = 50000$ for the Cover Type dataset. We use a smaller window for the Gaussian dataset because the data is lower dimensional and we have less unlabeled data, and a larger window for Cover Type because we have substantially more data (400K intermediate examples). We train the model for 5 epochs, 10 epochs, 20 epochs, and 100 epochs in each round for the Cover Type, rotating MNIST, Portraits, and Gaussian dataset respectively. The larger datasets need to be trained for fewer epochs because each epoch passes through many more examples—these numbers were chosen on validation data on the source without examining the intermediate or target data, and we show an ablation which suggests

Table 4. Percentage classification accuracies for gradual self-train (ST) and baselines on the Gaussian dataset, with 90% standard errors for the mean over 5 runs in parantheses. Gradual ST does better than self-training directly on the target or self-training on all the unlabeled data pooled together and gets an average accuracy of over 99% over 5 runs.

	GAUSSIAN
SOURCE MODEL	39.4 (±4.2)
TARGET ST	-9.1 (±14.7)
ALL ST	+46.7 (±12.4)
GRADUAL ST	**+59.9 (±4.0)**

that the results are not sensitive to these choices.

3. *Confidence thresholding*: We chose $\alpha = 0.1$ to filter out the 10% least confident examples, since these are examples the model is not confident on, so the predicted label is less likely to be correct. We run an ablation without this filtering and see that all methods perform slightly worse, but the relative ordering is similar—gradual self-training is still significantly better than all the other methods.

C.4. Results for the Gaussian Dataset

In Table 4 we show results for the three self-training methods on the synthetic Gaussian dataset, like in Table 1 for the other three datasets. Gradual self-training outperforms the baselines and gets an average accuracy of over 99%. Self-training on all the examples also does fairly well, getting an average accuracy of over 85%. Self-training on the target actually worsens performance because the domain shift is large—a model trained on the source (with no adaptation) gets under 50% accuracy on this binary classification task, so the source model does worse than random.

C.5. Ablations

We run ablations which suggest that the results in Section 5.1 are robust to the choice of algorithm hyperparameters.

Confidence thresholding: Table 5 shows the results for rotating MNIST and Portraits without confidence thresholding. All methods do worse without confidence thresholding but gradual self-training does significantly better than the other methods.

Window sizes: Table 6 shows the results for rotating MNIST and Portraits if we use smaller window sizes (from 2000 to 1000). Gradual self-training still does significantly better than the other methods.

Additional ablations for Portraits: We ran two additional ablations, focusing on Portraits. In the first ablation, we trained every method of self-training for 50% more epochs.

Table 5. Classification accuracies for gradual self-train (ST) and baselines without confidence thresholding/filtering, with 90% confidence intervals for the mean over 5 runs. All methods do worse without confidence thresholding but gradual self-training does significantly better than the other methods.

	ROT MNIST	PORTRAITS
SOURCE	30.5±1.0	76.2±0.5
TARGET ST	31.1±1.4	76.9±1.3
ALL ST	32.6±1.3	77.1±0.5
GRADUAL ST	**80.3±1.4**	**81.7±1.3**

Table 6. Classification accuracies for gradual self-train (ST) and baselines with smaller window sizes, with 90% confidence intervals for the mean over 5 runs. Gradual self-training still does significantly better than the other methods.

	ROT MNIST	PORTRAITS
SOURCE	35.6±1.7	74.1±1.4
TARGET ST	36.0±1.5	77.9±1.4
ALL ST	38.5±2.6	76.3±2.2
GRADUAL ST	**90.4±2.0**	**83.8±0.5**

Over 5 trials, gradual self-training got an accuracy of $83.9 \pm 0.4\%$, target self-train got an accuracy of $80.7 \pm 1.1\%$, and self-training to all unlabeled examples got an accuracy of $79.6 \pm 2.2\%$. The non-adaptive baseline got an accuracy of $77.3 \pm 1.0\%$.

We also ran an experiment on Portraits where we extrapolate further in time. Here we use the first 2000 images as source, next 20,000 images as unlabeled intermediate examples, and next 2000 images as the target. Here the accuracy of gradual self-training is $60.6 \pm 1.4\%$, self-training on the target directly is $56.5 \pm 1.4\%$, and self-training on all unlabeled data is $57.4 \pm 0.3\%$. Gradual self-training still does better, but all methods do quite poorly—developing and analyzing new techniques for gradual domain adaptation is an exciting avenue for future work.