Understanding Gradual Domain Adaptation: Improved Analysis, Optimal Path and Beyond

Haoxiang Wang [1] **Bo Li** [1] **Han Zhao** [1]

Abstract

The vast majority of existing algorithms for unsupervised domain adaptation (UDA) focus on adapting from a labeled source domain to an unlabeled target domain directly in a one-off way. Gradual domain adaptation (GDA), on the other hand, assumes a path of $(T-1)$ unlabeled intermediate domains bridging the source and target, and aims to provide better generalization in the target domain by leveraging the intermediate ones. Under certain assumptions, Kumar et al. (2020) proposed a simple algorithm, *gradual self-training*, along with a generalization bound in the order of $e^{\mathcal{O}(T)}(\varepsilon_0+\mathcal{O}(\sqrt{\frac{\log T}{n}}))$ for the target domain error, where ε_0 is the source domain error and n is the data size of each domain. Due to the exponential factor, this upper bound becomes vacuous when T is only moderately large. In this work, we analyze gradual self-training under more general and relaxed assumptions, and prove a significantly improved generalization bound as $\varepsilon_0+\widetilde{\mathcal{O}}(T\Delta+\frac{T}{\sqrt{n}}+\frac{1}{\sqrt{nT}})$, where Δ is the average distributional distance between consecutive domains. Compared with the existing bound with an *exponential* dependency on T as a *multiplicative* factor, our bound only depends on T *linearly and additively*. Perhaps more interestingly, our result implies the existence of an optimal choice of T that minimizes the generalization error, and it also naturally suggests an optimal way to construct the path of intermediate domains so as to minimize the accumulative path length $T\Delta$ between the source and target. To corroborate the implications of our theory, we examine gradual self-training on multiple semi-synthetic and real datasets, which confirms our findings. We believe our insights provide a path forward toward the design of future GDA algorithms.

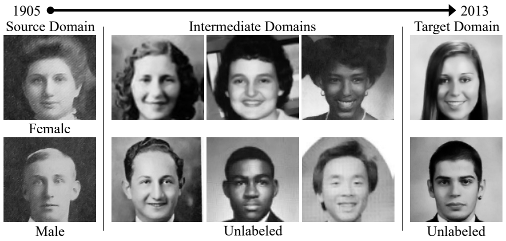


Figure 1. An example of gradual domain adaptation on Portraits (Ginosar et al., 2015), a historical image dataset of US high school yearbook. Given labeled data from a source domain, models are adapted to the target domain, with the help of unlabeled data from intermediate domains gradually shifting from the source to target.

1 Introduction

It is well known that machine learning models are generally not robust to distribution shifts (Sagawa et al., 2021; Koh et al., 2021; Hendrycks et al., 2021; Gulrajani & Lopez-Paz, 2021). As a result, models trained on one data distribution may have a severe performance drop on test data with a large distribution shift. Unsupervised domain adaptation (UDA) aims to tackle this challenge by adapting models to the test distribution with the help of unlabelled data from the target domain (Ganin et al., 2016; Long et al., 2015; Tzeng et al., 2017; Zhao et al., 2018).

In general, UDA considers a source domain and a target domain, in which the model is trained and tested, respectively. There is a discrepancy between the two domains, while they share certain similarities. Prior theoretical results show that the generalization error of UDA increases with the larger discrepancy between two domains (Ben-David et al., 2010; Zhao et al., 2019). However, empirically, due to the potentially large shift between these two domains, it may be hard to adapt to the target domain in a one-off fashion, and it is observed that existing UDA algorithms do not perform well (Kumar et al., 2020; Sagawa et al., 2021; Abnar et al., 2021) under large shifts.

Intuitively, one can expect that existing adaptation algorithms perform better under smaller shifts. Thus, one natural idea is to divide a large shift into multiple smaller shifts to mitigate the distribution shift issue. This "divide-and-conquer" idea has brought up to a setting known as *gradual domain adaptation* (GDA) (Hoffman et al., 2014; Gadermayr et al., 2018; Bobu et al., 2018; Wulfmeier et al., 2018;

[1] University of Illinois at Urbana-Champaign, Urbana, IL, USA. Correspondence to: Haoxiang Wang <hwang264@illinois.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Wang et al., 2020; Kumar et al., 2020; Abnar et al., 2021; Chen & Chao, 2021), where the learner has access to additional unlabeled data from some intermediate domains that *gradually* shift from the source to the target. GDA first fits a model to the source domain, then adapts it to a series of intermediate domains sequentially, and the ultimate goal is to generalize in the target domain. In this way, the large distribution shift is segmented into multiple smaller pieces between neighboring intermediate domains. Notably, in many real-world applications, the data distribution indeed changes gradually over time, distance, or environments (Hoffman et al., 2014; Farshchian et al., 2019; Kumar et al., 2020; Koh et al., 2021; Malinin et al., 2021; Zhao et al., 2021), in which the unlabelled intermediate domain data are available or can be easily acquired.

Kumar et al. (2020) proposes *Gradual Self-Training*, a simple yet effective GDA algorithm, which iteratively applies self-training (ST) on unlabelled data from intermediate domains. Briefly speaking, self-training (i.e., pseudo-labeling) is a popular semi-supervised learning technique that fine-tunes models with self-generated pseudo-labels on unlabelled data (Yarowsky, 1995; Lee et al., 2013; Xie et al., 2020). The power of gradual self-training is empirically validated on synthetic and real datasets by Kumar et al. (2020), and the authors also provide a theory to demonstrate the improvement of gradual self-training over baselines (i.e., source training only and vanilla self-training) in the presence of large shifts. However, the error bound of gradual self-training provided by Kumar et al. (2020) is quite pessimistic and unrealistic in a sense. Given source error ε_0 and T intermediate domains each with n unlabelled data, the bound of Kumar et al. (2020) scales as $e^{\mathcal{O}(T)}\big(\varepsilon_0+\mathcal{O}\big(\sqrt{\log(T)/n}\big)\big)$, which grows exponentially in T. This indicates that the more intermediate domains for adaptation, the worse performance that gradual self-training would obtain in the target domain. In contrast, people have empirically observed that a relatively large T is beneficial for gradual domain adaptation (Abnar et al., 2021; Chen & Chao, 2021).

Given the sharp gap between existing theory and empirical observations of gradual domain adaptation, we attempt to address the following important and fundamental questions:

> *For gradual domain adaptation, given the source domain and target domain, how does the number of intermediate domains impact the target generalization error? Is there an optimal choice for this number? If yes, then how to construct the optimal path of intermediate domains?*

To answer these questions, we focus on gradual self-training (Kumar et al., 2020), a representative gradual domain adaptation algorithm, and carry out a novel analysis drastically different from Kumar et al. (2020). Notably, our setting is more general than that of Kumar et al. (2020), in the sense that 1) we have a milder assumption on the distribution shift, 2) we put almost no restriction on the loss function, and 3) our technique applies to all the p-Wasserstein distance metrics. As a comparison, existing analysis is restricted to ramp loss[1] and only applies to the ∞-Wasserstein metric. At a high level, we first focus on analyzing a pair of consecutive domains, and upper bound the error difference of any classifier over domains bounded by their p-Wasserstein distance; then, we telescope this lemma to the entire path over a sequence of domains, and finally obtain an error bound for gradual self-training: $\varepsilon_0 + \widetilde{\mathcal{O}}(T\Delta + \frac{T}{\sqrt{n}} + \frac{1}{\sqrt{nT}})$, where Δ is the average p-Wasserstein distance between consecutive domains.

Interestingly, our bound indicates the existence of an optimal choice of T that minimizes the generalization error, which could explain the success of moderately large T used in practice. Notably, the $T\Delta$ in our bound could be interpreted as the length of the path of intermediate domains bridging the source and target, suggesting that one should also consider minimizing the path length $T\Delta$ in practices of gradual domain adaptation. For example, given fixed source and target domains, the path length $T\Delta$ is minimized as the intermediate domains are distributed along some Wasserstein geodesic between the source domain and target domain. We believe these insights on T and Δ obtained from our error bound are helpful to construct an optimal path of intermediate domains in bridging the source domain and target domain.

Empirically, we examine gradual self-training on two synthetic datasets (color-shift MNIST & rotated MNIST) and two real dataset (Portraits (Ginosar et al., 2015) & Cover-Type (Blackard & Dean, 1999)). Our experiments validate the insights of our theory: there indeed exists an optimal choice of T, and the intermediate domains should be chosen to minimize $T\Delta$. Our empirical observation sheds new light on the importance of constructing the optimal path connecting the source domain and target domain in gradual domain adaptation, and we hope our insight could inspire the design of future gradual domain adaptation algorithms.

2 Related Work

Self-Training Self-training, i.e., pseudo-labeling, is a popular semi-supervised learning approach (Yarowsky, 1995; Lee et al., 2013), which fine-tunes trained classifiers with pseudo-labels predicted on unlabelled data. There are some common techniques that can enhance self-training, e.g., adding noise to inputs or networks (Xie et al., 2020; Sohn et al., 2020), progressively assigning pseudo-labels on unlabelled data with high prediction confidence, or applying a strong regularization (Arazo et al., 2020; Pham et al., 2021).

Unsupervised Domain Adaptation Unsupervised domain

[1] Ramp loss can be seen as a truncated hinge loss so that it is bounded and more amenable for technical analysis.

adaptation (UDA) focuses on adapting models trained on labeled data of a source domain to a target domain with unlabelled data. A number of approaches have been proposed for UDA in recent years. Invariant representation learning is one of the most popular approaches (Sun & Saenko, 2016; Zhao et al., 2019), where adversarial training is usually adopted to learn feature representations invariant between source and target domains (Ajakan et al., 2014; Ganin et al., 2016). In addition, self-training (i.e., pseudo-labelling) is also adapted for UDA (Liang et al., 2019; 2020; Zou et al., 2018; 2019). In summary, pseudo-labels of unlabelled target domain data are generated by source-trained classifiers, and they are used to further fine-tune the trained classifiers. Notably, the performance of self-training degrades significantly when there is a large distribution shift between the source and target.

Gradual Domain Adaptation Gradual domain adaptation (GDA) introduces extra intermediate domains to the existing source and target domains of UDA. In general, the intermediate domains shift from the source to the target gradually, such as rotation or time evolution (Kumar et al., 2020), and GDA iteratively adapts models from the source to the target along the sequence of intermediate domains. The idea of GDA has been empirically explored in computer vision (Gadermayr et al., 2018; Hoffman et al., 2014; Wulfmeier et al., 2018) with various domain-specific algorithms. Recently, Kumar et al. (2020) proposes a general machine learning algorithm for GDA, *gradual self-training*, which outperforms vanilla self-training on several synthetic and real datasets. Moreover, Kumar et al. (2020) provides a generalization error bound for gradual self-training, which is the first theoretical guarantee for GDA. In parallel, Wang et al. (2020) proposes an adversarial adaptation algorithm for GDA. Later, Abnar et al. (2021) proposes a variant of gradual self-training that does not need intermediate domain data, since it could generate pseudo-data for intermediate domains. In GDA, the intermediate domains are given (i.e., ordering domains based on their distance to the source/target domain), Chen & Chao (2021) removes the requirement of the domain order by proposing a method called Intermediate Domain Labeler (IDOL). Chen & Chao (2021) shows gradual self-training with IDOL could obtain good performance even without the domain order. Recently, Zhou et al. (2022) proposed an algorithm under the teacher-student paradigm with active query strategy to tackle the gradual adaptation problem. Dong et al. (2022) studied a slightly different setting where labeled data is also available during the intermediate domains. In addition, one may notice that GDA has a setting similar to temporal domain generalization (Koh et al., 2021; Ye et al., 2022), while the latter has available labels in intermediate domains instead.

3 Preliminaries

Notation $\mathcal{X}, \mathcal{Y}$ denote the input and the output space, and X, Y denote random variables taking values in $\mathcal{X}, \mathcal{Y}$. In this work, each domain has a data distribution μ over $\mathcal{X} \times \mathcal{Y}$, thus it can be written as $\mu = \mu(X, Y)$. When we only consider samples and disregard labels, we use $\mu(X)$ to refer to the sample distribution of μ over the input space $\mathcal{X}$.

3.1 Problem Setup

Binary Classification In this work, we focus on binary classification with labels $\{-1, 1\}$. Also, we consider $\mathcal{Y}$ as a compact space in $\mathbb{R}$.

Gradually shifting distributions We have $T+1$ domains indexed by $\{0, 1, ..., T\}$, where domain 0 is the source domain, domain T is the target domain and domain $1, \dots, T-1$ are the intermediate domains. These domains have distributions over $\mathcal{X} \times \mathcal{Y}$, denoted as $\mu_0, \mu_1, \dots, \mu_T$.

Classifier and Loss Consider the hypothesis class as $\mathcal{H}$ and the loss function as ℓ. We define the population loss of classifier $h \in \mathcal{H}$ in domain t as

$$\varepsilon_t(h) \equiv \varepsilon_{\mu_t}(h) \triangleq \mathbb{E}_{\mu_t}[\ell(h(x),\ y)] = \mathbb{E}_{x,y\sim\mu_t}[\ell(h(x),\ y)]$$

Gradual Domain Adaptation In the standard setting of unsupervised domain adaptation (UDA) (Zhao et al., 2019), a model is trained with n_0 labeled data of the source domain and n unlabelled data of the target domain, and it is evaluated by labeled test data of the target domain. In gradual domain adaptation (GDA) (Kumar et al., 2020), the model is given additional $T-1$ sequentially indexed intermediate domains, each with n unlabelled data. GDA algorithms usually train the model in the source, then adapt it sequentially over the intermediate domains toward the target. Same as UDA, models trained by GDA are also evaluated by labeled test data of the target domain.

We make a mild assumption on the input data below, which can be easily achieved by data preprocessing. This assumption is common in machine learning theory works (Cao & Gu, 2019; Arora et al., 2019; Rakhlin & Sridharan, 2014).

Assumption 1 (Bounded Input Space)**.** *Consider the input space $\mathcal{X}$ is compact and bounded in the d-dimensional unit L_2 ball, i.e., $\mathcal{X} \subseteq \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$.*

To quantify distribution shifts between domains, we adopt the well-known Wasserstein distance metric in the Kantorovich formulation (Kantorovich, 1939), which is widely used in the optimal transport literature (Villani, 2009).

Definition 1 (p-Wasserstein Distance)**.** *Consider two measures μ and ν over $\mathbb{S} \subseteq \mathbb{R}^d$. For any $p \geq 1$, their p-Wasserstein distance is defined as*

$$W_p(\mu, \nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{S}\times\mathbb{S}} d(x, y)^p \, \mathrm{d}\gamma(x, y) \right)^{1/p} \quad (1)$$

where $\Gamma(\mu, \nu)$ is the set of all measures over $\mathbb{S} \times \mathbb{S}$ with marginals equal to μ and ν respectively.

In this paper, we consider p as a preset constant satisfying $p \geq 1$. Then, we can use the p-Wasserstein metric to measure the distribution shifts between consecutive domains.

Definition 2 (Distribution Shifts)**.** *For $t = 1, \dots, T$, denote*

$$\Delta_t = W_p(\mu_{t-1}, \mu_t) \tag{2}$$

Then, we define the average of distribution shifts between consecutive domains as

$$\Delta = \frac{1}{T} \sum_{t=1}^{T} \Delta_t \tag{3}$$

Remarks on Wasserstein Metrics The p-Wasserstein metric has been widely adopted in many sub-areas of machine learning, such as generative models (Arjovsky et al., 2017; Tolstikhin et al., 2018) and domain adaptation (Courty et al., 2014; 2016; 2017; Redko et al., 2019). Most of these works use $p = 1$ or 2, which is known to be good at quantifying many real-world data distributions (Peyré et al., 2019). However, the analysis in Kumar et al. (2020) only applies to $p = \infty$, which is uncommon in practice and can lead to a loose upper bound due to the monotonicity property of W_p. In fact, for some pairs of data distributions that look close to each other, the W_∞ distance may even become unbounded while W_1 and W_2 distances are small (e.g., with a few outlier data).

3.2 Gradual Self-Training

The vanilla self-training algorithm (denoted as ST) adapts classifier h with empirical risk minimization (ERM) over pseudo-labels generated on an unlabelled dataset S, i.e.,

$$h' = \mathrm{ST}(h, S) = \arg\min_{f \in \mathcal{H}} \sum_{x \in S} \ell(f(x), h(x)) \tag{4}$$

where $h(x)$ represents pseudo-labels provided by the trained classifier h, and h' is the new classifier fitted to the pseudo-labels. The technique of hard labelling (i.e., converting $h(x)$ to one-hot labels) is used in some practices of self-training (Xie et al., 2020; Van Engelen & Hoos, 2020), which can be viewed as adding a small modification to the loss function ℓ.

In gradual domain adaptation, we assume a set of n unlabelled data from each intermediate domain and the target domain. Namely, any domain $t \in \{1, \dots, T\}$ has a set of n unlabelled data i.i.d. sampled from μ_t, denoted as S_t. For simplicity, we assume the number of labeled data in the source domain is also equal to n.

Gradual self-training (Kumar et al., 2020), applies self-training to the intermediate domains and the target domain successively, i.e., for $t = 1, \dots, T$,

$$h_t = \mathrm{ST}(h_{t-1}, S_t) = \arg\min_{f \in \mathcal{H}} \sum_{x \in S_t} \ell(f(x), h_{t-1}(x)) \tag{5}$$

where h_0 is the model fitted on the source data. h_T is the final trained classifier that is expected to enjoy a low population error in the target domain, i.e., ε_T.

4 Theoretical Analyses

In this section, we theoretically analyze gradual self-training under assumptions more relaxed than Kumar et al. (2020), and obtain a significantly improved error bound. Our theoretical analysis is roughly split into two steps: (i) we focus on a pair of arbitrary consecutive domains with bounded distributional distance, and upper bound the prediction error difference of any classifier in the two domains by the distributional distance (Lemma 1); (ii) we view gradual self-training from an online learning perspective, and adopt tools in the online learning literature to analyze the algorithm together with results of step (i), leading to an upper bound (Theorem 1) of the target generalization error of gradual self-training. Notably, our bound provides several profound insights on the optimal path of intermediate domains used in gradual domain adaptation (GDA), and also sheds light on the design of GDA algorithms. The proofs of all theoretical statements are provided in Appendix A.

4.1 Error Difference over Distribution Shift

Intuitively, gradual domain adaptation (GDA) splits the large distribution shift between the source domain and target domain into smaller shifts that are segmented by intermediate domains. Thus, in the view of reductionism (Anderson, 1972), one should understand what happens in a pair of consecutive domains in order to comprehend the entire GDA mechanism.

To start, we adopt three assumptions from the prior work (Kumar et al., 2020)[2].

Assumption 2 (R-Lipschitz Classifier)**.** *We assume each classifier $h \in \mathcal{H}$ is R-Lipschitz in ℓ_2 norm, i.e., $\forall x, x' \in \mathcal{X}$,*

$$|h(x) - h(x')| \leq R \|x - x'\|_2$$

Assumption 3 (ρ-Lipschitz Loss)**.** *We assume the loss function ℓ is ρ-Lipschitz, i.e., $\forall y, y' \in \mathcal{Y}$,*

$$|\ell(y, \cdot) - \ell(y', \cdot)| \leq \rho \|y - y'\|_2 \tag{6}$$

$$|\ell(\cdot, y) - \ell(\cdot, y')| \leq \rho \|y - y'\|_2 \tag{7}$$

Assumption 4 (Bounded Model Complexity[3])**.** *We assume the Rademachor complexity (Bartlett & Mendelson, 2002), $\mathcal{R}$, of the hypothesis class, $\mathcal{H}$, is bounded for any distribution μ considered in this paper. That is, for some constant $B > 0$,*

$$\mathcal{R}_n(\mathcal{H}; \mu) = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i)\right] \leq \frac{B}{\sqrt{n}}$$

where the expectation is w.r.t. $x_i \sim \mu(X)$ and $\sigma_i \sim \mathrm{Uniform}(\{-1, 1\})$ for $i = 1, \dots, n$.

[2] Assumption 3 is not explicitly made by Kumar et al. (2020). Instead, they directly assume the loss function to be ramp loss, which is a more strict assumption than our Assumption 3.

[3] This assumption is actually reasonable and not strong. For example, under Assumption 1 and 2, linear models directly satisfy (6), as proved in (Kumar et al., 2020; Liang, 2016).

With these assumptions, we can bound the population error difference of a classifier between a pair of shifted domains in the following proposition. The proof is in Appendix A.1.

Lemma 1 (Error Difference over Shifted Domains)**.** *Consider two arbitrary measures μ, ν over $\mathcal{X} \times \mathcal{Y}$. Then, for arbitrary classifier h and loss function l satisfying Assumption 2, 3, the population loss of h on μ and ν satisfies*

$$|\varepsilon_\mu(h) - \varepsilon_\nu(h)| \leq \rho\sqrt{R^2+1}\, W_p(\mu, \nu) \tag{8}$$

where W_p is the Wasserstein-p distance metric and $p \geq 1$.

Eq. (5) depicts each iteration of gradual self-training with an past classifier h_t and a new one h_{t+1}, which are fitted to S_t and S_{t+1}, respectively. Naturally, one might be curious about how well the performance of h_{t+1} in domain $t+1$ is compared with h_t in domain t. We answer this question as follows, with proof in Appendix A.2.

Proposition 1 (The stability of the ST algorithm)**.** *Consider two arbitrary measures μ, ν, and denote S as a set of n unlabelled samples i.i.d. drawn from μ. Suppose $h \in \mathcal{H}$ is a pseudo-labeler that provides pseudo-labels for samples in S. Define $\hat{h} \in \mathcal{H}$ as an ERM solution fitted to the pseudo-labels,*

$$\hat{h} = \arg\min_{f \in \mathcal{H}} \sum_{x \in S} \ell(f(x), h(x)) \tag{9}$$

Then, for any $\delta \in (0, 1)$, the following bound holds true with probability at least $1 - \delta$,

$$|\varepsilon_\mu(\hat{h}) - \varepsilon_\nu(h)| \leq \mathcal{O}\left(W_p(\mu, \nu) + \frac{\rho B + \sqrt{\log \frac{1}{\delta}}}{\sqrt{n}}\right) \tag{10}$$

Comparison with Kumar et al. (2020) The setting of Kumar et al. (2020) is more restrictive than ours. For example, its analysis is specific to ramp loss (Huang et al., 2014), a rarely used loss function for binary classification. Kumar et al. (2020) also studies the error difference over consecutive domains, and prove a multiplicative bound (in Theorem 3.2 of Kumar et al. (2020)), which can be re-expressed in terms of our notations and assumptions as

$$\varepsilon_\mu(\hat{h}) \leq \frac{2}{1 - R\Delta_\infty} \varepsilon_\nu(h) + \varepsilon^*_\mu + \mathcal{O}\left(\frac{\rho B + \sqrt{\log \frac{1}{\delta}}}{\sqrt{n}}\right) \tag{11}$$

where $\varepsilon^*_\mu \triangleq \min_{f \in \mathcal{H}} \varepsilon_\mu(f)$ is the optimal error of $\mathcal{H}$ in μ, and $\Delta_\infty \triangleq \max_{y \in \{-1,1\}} (W_\infty(\mu(X|Y=y), \nu(X|Y=y)))$ can be seen as an analog to the $W_p(\mu, \nu)$ in (10). Kumar et al. (2020) assumes $1 - R\Delta_\infty > 0$, thus the error $\varepsilon_\nu(h)$ is increased by the factor $\frac{2}{1-R\Delta_\infty} > 1$ in the above error bound of $\varepsilon_\mu(\hat{h})$. This leads to a target domain error bound *exponential* in T (Corollary 3.3. of Kumar et al. (2020)) when one applies (11) to the sequence of domains iteratively in gradual self-training (i.e., Eq. (5)). In contrast, our (10) indicates $\varepsilon_\mu(\hat{h}) \leq \varepsilon_\nu(h) + \text{other terms}$, which increases the error $\varepsilon_\nu(h)$ in an *additive* way, leading to a target domain error bound *linear* in T.

Remarks on Generality Lemma 1 and Proposition 1 are not restricted to gradual domain adaptation. Of independent interest, they can be leveraged as useful theoretical tools to handle distribution shifts in other machine learning problems, including unsupervised domain adaptation, transfer learning, OOD robustness, and group fairness.

4.2 An Online Learning View of GDA

One can naively apply Proposition 1 to gradual self-training over the sequence of domains (i.e., Eq. (5)) iteratively and obtain an error bound of the target domain as

$$\varepsilon_T(h_T) \leq \varepsilon_0(h_0) + \mathcal{O}\left(T\Delta + T\frac{\rho B + \sqrt{\log \frac{1}{\delta}}}{\sqrt{n}}\right) \tag{12}$$

Obviously, the larger T, the higher the error bound becomes (this holds even if one assumes $T\Delta \leq \text{constant}$ for fixed source and target domains). However, this contradicts with empirical observations that a moderately large T is optimal (Kumar et al., 2020; Abnar et al., 2021; Chen & Chao, 2021).

To resolve this discrepancy, we take an online learning view of gradual domain adaptation, and expect to obtain a more optimistic error bound. Specifically, we consider the domains $t = 0, \ldots, T$ coming to the model in a sequential way. As the model observes data of domain t, it updates itself using ERM over pseudo-labels of the newly observed data, and then it will enter the next iteration of the domain $t+1$. Specifically, the first iteration is the source domain, and the model updates itself using ERM over the labeled data, where the labels can be seen as pseudo-labels generated by a ground-truth labeling function.

To proceed, certain structural assumptions and complexity measures are necessary. For example, VC dimension (Vapnik, 1999) and Rademacher complexity (Bartlett & Mendelson, 2002) are proposed for supervised learning. Similarly, in online learning, Littlestone dimension (Littlestone, 1988), sequential covering number (Rakhlin et al., 2010) and sequential Rademacher complexity (Rakhlin et al., 2010; 2015) are developed as useful complexity measures. To study gradual self-training in an online learning framework, we adopt the framework of Rakhlin et al. (2015), which views online binary classification as a process in the structure of a *complete binary tree* and defines the *sequential Rademacher complexity* upon that.

Definition 3 (Complete Binary Trees)**.** *We define two complete binary trees $\mathscr{X}, \mathscr{Y}$, and the path $\boldsymbol{\sigma}$ in the trees:*

$\mathscr{X} \triangleq (\mathscr{X}_0, ..., \mathscr{X}_T)$, a sequence of mappings with $\mathscr{X}_t : \{\pm 1\}^t \to \mathcal{X}$ for $t = 0, ..., T$.
$\mathscr{Y} \triangleq (\mathscr{Y}_0, ..., \mathscr{Y}_T)$, a sequence mappings with $\mathscr{Y}_t : \{\pm 1\}^t \to \mathcal{Y}$ for $t = 0, ..., T$.
$\boldsymbol{\sigma} = (\sigma_0, ..., \sigma_T) \in \{\pm 1\}^t$, a path in $\mathscr{X}$ or $\mathscr{Y}$.

Definition 4 (Sequential Rademacher Complexity)**.** *Con-*

sider $\boldsymbol{\sigma}$ as a sequence of Rademacher random variables and a t-dimensional probability vector $\mathbf{q}_t = (q_0, ..., q_{t-1})$, then the sequential Rademacher complexity of $\mathcal{H}$ is

$$\mathcal{R}_t^{\text{seq}}(\mathcal{H}) = \sup_{\mathscr{X}, \mathscr{Y}} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \sum_{\tau=0}^{t-1} \sigma_\tau q_\tau \ell\big(h(\mathscr{X}_\tau(\boldsymbol{\sigma})), \mathscr{Y}_\tau(\boldsymbol{\sigma})\big) \right]$$

To better understand this measure, we present examples of two common model classes, which are provided in Rakhlin & Sridharan (2014).

Example 1 (Linear Models)**.** *For the linear model class that is R-Liphschtiz, i.e., $\mathcal{H} = \{x \to w^\top x : \|w\|_2 \leq R\}$, we have $\mathcal{R}_t^{\text{seq}}(\mathcal{H}) \leq \frac{R}{\sqrt{t}}$ for $t \in \mathbb{Z}_+$.*

Example 2 (Neural Networks)**.** *Consider $\mathcal{H}$ as the hypothesis class of R-Lipschitz L-layer fully-connected neural nets with 1-Lipschitz activation function (e.g., ReLU, Sigmoid, TanH). Then, its sequential Rademacher complexity is bounded as $\mathcal{R}_t^{\text{seq}}(\mathcal{H}) \leq \mathcal{O}\left(R\sqrt{\frac{(\log t)^{3(L-1)}}{t}}\right)$ for $t \in \mathbb{Z}_+$.*

Besides the model complexity measure, we also adopt a measure of discrepancy among multiple data distributions, which is proposed in works of online learning for time-series data (Kuznetsov & Mohri, 2014; 2015; 2016; 2017; 2020).

Definition 5 (Discrepancy Measure)**.** *For any t-dimensional probability vector $\mathbf{q}_t = (q_0, ..., q_{t-1})$, the discrepancy measure $\operatorname{disc}(\mathbf{q}_t)$ is defined as*

$$\operatorname{disc}(\mathbf{q}_t) = \sup_{h \in \mathcal{H}} \left(\varepsilon_{t-1}(h) - \sum_{\tau=0}^{t-1} q_\tau \cdot \varepsilon_\tau(h) \right) \tag{13}$$

We can further bound this discrepancy in our setting (defined in Sec. 3) as follows. The proof is in Appendix A.3.

Lemma 2 (Discrepancy Bound)**.** *With Lemma 1, the discrepancy measure (13) can be upper bounded as*

$$\operatorname{disc}(\mathbf{q}_t) \leq \rho\sqrt{R^2+1} \sum_{\tau=0}^{t-1} q_\tau (t-\tau-1)\Delta \tag{14}$$

With $\mathbf{q}_t = \mathbf{q}_t^* = (\frac{1}{t}, ..., \frac{1}{t})$, this upper bound can be minimized as

$$\operatorname{disc}(\mathbf{q}_t^*) \leq \rho\sqrt{R^2+1}\, t\Delta/2 = \mathcal{O}(t\Delta) \tag{15}$$

4.3 Generalization Bound for Gradual Self-Training

With our results obtained in Section 4.1 and tools introduced in Section 4.2, we can prove a generalization bound for gradual self-training within online learning frameworks such as Kuznetsov & Mohri (2016; 2020). However, if we use these frameworks in an off-the-shelf way, the resulting generalization bound will have multiple terms with dependence on T and no dependence on n (the number of samples per domain), since these online learning works do not care about the data size of each domain. This will cause the resulting bound to be loose in terms of n. To resolve this, we come up with a novel reductive view of the learning process of gradual self-training, which is more fine-grained than the original view in Kumar et al. (2020). This reductive view enables us to make the generalization bound to depend on n in an intuitive way, which also tightens the final bound. We defer explanations of this view to Appendix A.4 along with the proof of Theorem 1.

Finally, we prove a generalization bound for gradual self-training that is much tighter than that of Kumar et al. (2020).

Theorem 1 (Generalization Bound for Gradual Self-Training)**.** *For any $\delta \in (0,1)$, the population loss of gradually self-trained classifier h_T in the target domain is upper bounded with probability at least $1-\delta$ as*

$$\varepsilon_T(h_T) \leq \sum_{t=0}^{T} q_t \varepsilon_t(h_t) + \|\mathbf{q}_{n(T+1)}\|_2 \left(1+\mathcal{O}\left(\sqrt{\log(1/\delta)}\right)\right) + \operatorname{disc}(\mathbf{q}_{T+1}) + \mathcal{O}\left(\sqrt{\log T}\,\mathcal{R}_{n(T+1)}^{\text{seq}}(\ell \circ \mathcal{H})\right)$$

For the class of neural nets considered in Example 2,

$$\varepsilon_T(h_T) \leq \varepsilon_0(h_0) + \mathcal{O}\Bigg(T\Delta + \frac{T}{\sqrt{n}} + T\sqrt{\frac{\log 1/\delta}{n}} + \frac{1}{\sqrt{nT}} + \sqrt{\frac{(\log nT)^{3L-2}}{nT}} + \sqrt{\frac{\log 1/\delta}{nT}}\Bigg) \tag{16}$$

Remark The bound in Eq. (16) is rather intuitive: the first term $\varepsilon_0(h_0)$ is the source error of the initial classifier, and $T\Delta$ corresponds to the total length of the path of intermediate domains connecting the source domain and the target domain. The asymptotic $\mathcal{O}(T/\sqrt{n})$ term is due to the accumulated estimation error of the pseudo-labeling algorithm incurred at each step. The $\mathcal{O}(1/\sqrt{nT})$ term characterizes the overall sample size used by the algorithm along the path, i.e., the algorithm has seen n samples in each domain, and there are T total domains that gradual self-training runs on.

Comparison with Kumar et al. (2020) Using our notation, the generalization bound of Kumar et al. (2020) can be re-expressed as

$$\varepsilon_T(h_T) \leq e^{\mathcal{O}(T)}\left(\varepsilon_0(h_0) + \mathcal{O}\Big(\frac{1}{\sqrt{n}} + \sqrt{\frac{\log T}{n}}\Big)\right), \tag{17}$$

which grows *exponentially* in T as a multiplicative factor. In contrast, our bound (16) grows only additively and linearly in T, achieving an *exponential improvement* compared with the bound of Kumar et al. (2020) shown in (17).

4.4 Optimal Path of Gradual Self-Training

It is worth pointing out that our generalization bound in Theorem 1 applies to any path connecting the source domain and target domain with T steps, as long as μ_0 is the source domain and μ_T is the target domain. In particular, if we define Δ_{max} to be an upper bound[4] on the average W_p

[4] Need to be large enough to ensure that $\mathcal{P}$ is non-empty.

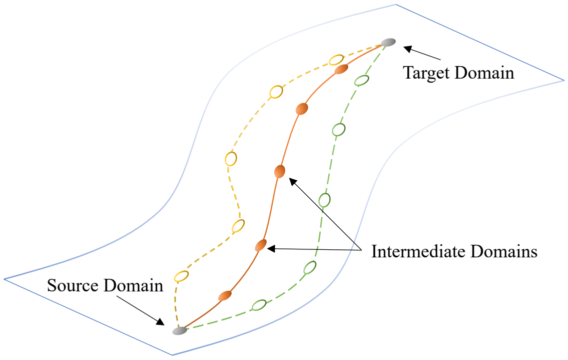


Figure 2. An illustration of the optimal path in gradual domain adaptation, with a detailed explanation in Sec. 4.4. The orange path is the geodesic connecting the source domain and target domain.

distance between any pair of consecutive domains along the path, i.e., $\Delta_{\max} \geq \frac{1}{T}\sum_{t=1}^{T} W_p(\mu_{t-1}, \mu_t)$, and let $\mathcal{P}$ to be the collection of paths with T steps connecting μ_0 and μ_T:

$$\mathcal{P} := \{(\mu_t)_{t=0}^{T} \mid \frac{1}{T}\sum_{t=1}^{T} W_p(\mu_{t-1}, \mu_t) \leq \Delta_{\max}\} ,$$

then we can extend the generalization bound in Theorem 1:

$$\varepsilon_T(h_T) \leq \varepsilon_0(h_0) + \inf_{\mathcal{P}} \widetilde{\mathcal{O}}\left(T\Delta_{\max} + \frac{T}{\sqrt{n}} + \sqrt{\frac{1}{nT}}\right). \quad (18)$$

Minimizing the RHS of the above upper bound w.r.t. T (the proof is provided in Appendix A.6), we obtain the optimal choice of T on the order of

$$\widetilde{\mathcal{O}}\left(\left(\frac{1}{1+\Delta_{\max}\sqrt{n}}\right)^{2/3}\right). \quad (19)$$

However, the above asymptotic optimal length may not be achievable, since we need to ensure that $T\Delta_{\max}$ is at least the length of the geodesic connecting the source domain and target domain. To this end, define L to be the W_p distance between the source domain and target domain, we thus have the optimal choice T^* as

$$T^* = \max\left\{\frac{L}{\Delta_{\max}}, \widetilde{\mathcal{O}}\left(\left(\frac{1}{1+\Delta_{\max}\sqrt{n}}\right)^{2/3}\right)\right\}. \quad (20)$$

Intuitively, the inverse scaling of T^* and $\Delta_{\max}$ suggests that, if the average distance between consecutive domains is large, it is better to take fewer intermediate domains.

Illustration of the Optimal Path To further illustrate the notion of the optimal path connecting the source domain and target domain implied by our theory, we provide an example in Fig. 2. Consider the metric space induced by W_p over all the joint distributions with finite p-th moment, where both the source and target could be understood as two distinct points. In this case, there are infinitely many paths of step size T connecting the source and target, such that the average pairwise distance is bounded by $\Delta_{\max}$. Hence, one insight we can draw from Eq. (18) is that: if the learner could construct the intermediate domains, then it is better to choose the path that is as close to the geodesic, i.e., the shortest path between the source and target (under W_p), as possible. This key observation opens a broad avenue forward toward algorithmic designs of gradual domain adaptation to *construct* intermediate domains for better generalization performance in the target domain. However, the design and discussion of algorithms along this direction are beyond the scope of this paper, and we leave it to future works.

5 Experiments

To empirically validate our theoretical findings, we examine gradual self-training on two synthetic and two real datasets.

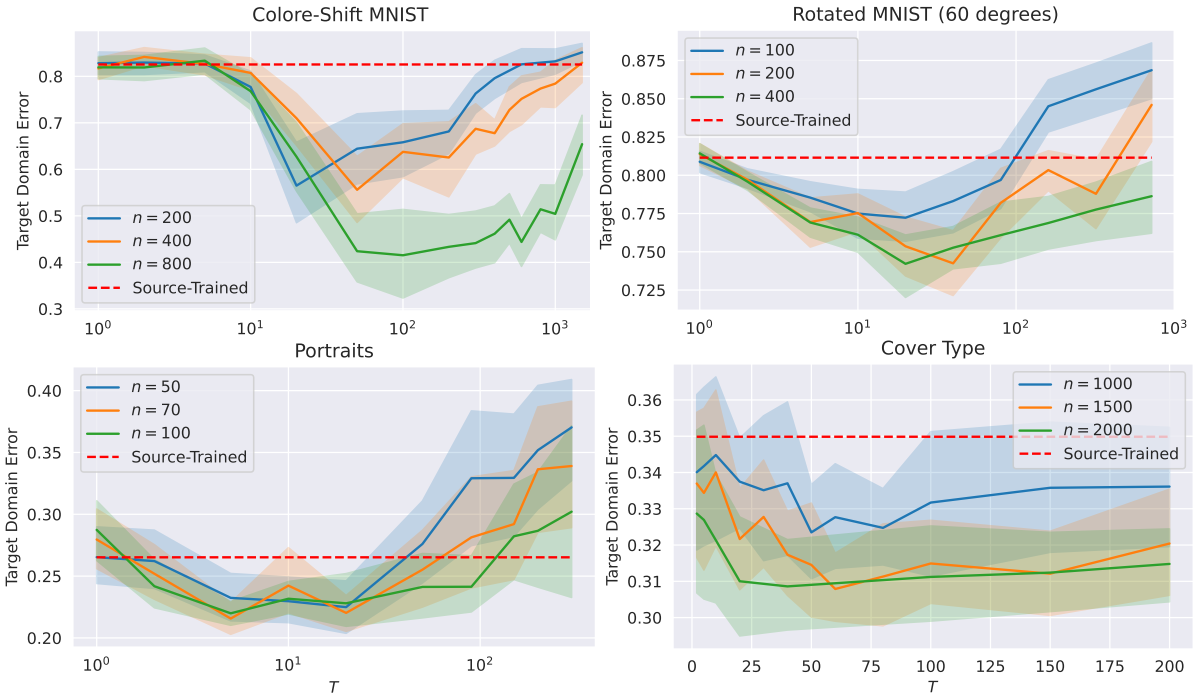
Implementation We adopt the official code of gradual self-training by Kumar et al. (2020) that is implemented in Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2016). The Color-Shift MNIST and CoverType datasets are not covered by the official code of Kumar et al. (2020), thus we include them by ourselves. We adopt the data splitting of CoverType from Kumar et al. (2020), and our data splitting of Rotated MNIST and Portraits are mostly the same as Kumar et al. (2020). Following Kumar et al. (2020), we use the architecture of 3-layer ReLU MLP with BatchNorm (Ioffe & Szegedy, 2015) and Dropout(0.5) (Srivastava et al., 2014), and apply it to all datasets. We use the cross-entropy loss and the Adam optimizer (Kingma & Ba, 2015) following the practices of Kumar et al. (2020).

Datasets We use two semi-synthetic datasets built upon MNIST (LeCun & Cortes, 1998): Color-Shift MNIST and Rotated MNIST. Color-Shift MNIST is our custom dataset that shifts the pixel color values of grayscale MNIST images, and Rotated MNIST is a common dataset for gradual domain adaptation studies (Kumar et al., 2020; Abnar et al., 2021; Chen & Chao, 2021). We also consider two real datasets, CoverType and Portraits, which are used by prior GDA works (Kumar et al., 2020; Chen & Chao, 2021).

Color-Shift MNIST: We normalize the pixel value of each MNIST image from [0,255] to [0,1]. The source domain is of the original MNIST data distribution, and the target domain contains images with pixels shifted by +1, i.e., the pixel value range is shifted from [0,1] to [1,2]. The 50K training set images of MNIST are split into a source domain of 5K images (no shift) and intermediate domains of the rest data (shifted by a value between 0 and +1 uniformly). 10K MNIST test images are all shifted by +1 to the target.

Rotated MNIST: A semi-synthetic dataset rotating MNIST images by an angle between 0 and 60 degrees. The 50K training set images of MNIST are divided into a source domain of 5K images (no rotation), intermediate domains of 42K images (0-60 degrees), and a set of validation data of the rest images. The 10K MNIST test set images are all rotated by 60 degrees to become data of the target domain.

CoverType (Blackard & Dean, 1999): A tabular dataset



Colore-Shift MNIST

Rotated MNIST (60 degrees)

Portraits

Cover Type

Figure 3. Empirical results of gradual self-training with increasing T on four datasets. The red dashed line is for models that are purely trained on the source domain and evaluated in the target domain. In each plot, results of three choices of n (number of samples per intermediate domain) are provided. The shading of each curve indicates the standard deviation of measured error over 20 runs.

hosted by the UCI repository (Dua & Graff, 2017) that aims to predict the forest cover type given 54 features. Following Kumar et al. (2020), we sort examples by their distances to the water body in ascending order, and split the data into a source domain (first 50K examples), intermediate domains (following 400K examples), and a target domain (the last 50K examples).

Portraits (Ginosar et al., 2015): An image dataset of grayscale photos of high school seniors from 1905 to 2013 (Fig. 1). We split the dataset into a source domain of 1905-1930 (1K images), intermediate domains of 1931-2009 (34K images), and a target domain of 2010-2013 (1K examples).

Empirical Results We run gradual self-training with various choices of n and T over these four datasets (i.e., there are $T-1$ intermediate domains and 1 target domain, each with n unlabelled data for adaptation). Each curve on the figure is the average accuracy measured over 20 repeated experiments. Empirical results shown in Fig. 3 are consistent with the theoretical prediction of our generalization bound that we discuss in Sec. 4.4: as the source and target are fixed, along a chosen path of intermediate domains (e.g., counter-clockwise rotation in Rotated MNIST from 0 to 60 degrees), the target domain test error decreases then increases, indicating the existence of an optimal choice of T for each n of consideration.

Remarks Our empirical results suggest that in practices of gradual domain adaptation, the hyper-parameters T and n are crucial and should be carefully treated. Also, if one can collect or generate intermediate domain data (Abnar et al., 2021), the choices of intermediate domains should be examined in advance, and our theoretical findings in this paper could serve as a guide.

6 Conclusion and Discussion

For unsupervised gradual domain adaptation, we provide a significantly improved analysis for the generalization error of the gradual self-training algorithm, under a more general setting with relaxed assumptions. In particular, compared with existing results, our bound provides an *exponential* improvement on the dependency of the step size T, as well as a better sample complexity of $O(1/\sqrt{nT})$, as opposed to $O(1/\sqrt{n})$ as in the existing work. Perhaps more interestingly, our generalization bound contains one term that admits a natural and intuitive interpretation: the length of the path produced by the intermediate domains that connect the source domain and target domain. Hence, our theory indicates that when constructing intermediate domains, an algorithm should aim to find those on the geodesic connecting the source domain and target domain. We believe this insight can open a broad avenue toward future algorithm designs for gradual domain adaptation when no intermediate domains are available. It also remains an open question on how to efficiently construct the optimal intermediate domains when only unlabeled data are available from the target domain.

Acknowledgements

This work is partially supported by NSF grant No.1910100, NSF CNS No.2046726, C3 AI, and the Alfred P. Sloan Foundation. BL and HZ would like to thank the support from a Facebook research award.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: A system for Large-Scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, Savannah, GA, November 2016. USENIX Association. ISBN 978-1-931971-33-1. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

Abnar, S., Berg, R. v. d., Ghiasi, G., Dehghani, M., Kalchbrenner, N., and Sedghi, H. Gradual domain adaptation in the wild: When intermediate distributions are absent. *arXiv preprint arXiv:2106.06080*, 2021.

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., and Marchand, M. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

Anderson, P. W. More is different. *Science*, 177(4047): 393–396, 1972.

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*, pp. 1–8. IEEE, 2020.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W. (eds.), *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 06–11 Aug 2017.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332, 2019.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

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

Blackard, J. A. and Dean, D. J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

Bobu, A., Tzeng, E., Hoffman, J., and Darrell, T. Adapting to continuously shifting domains, 2018. URL https://openreview.net/forum?id=BJsBjPJvf.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 10835–10845, 2019.

Chen, H.-Y. and Chao, W.-L. Gradual domain adaptation without indexed intermediate domains. *Advances in Neural Information Processing Systems*, 34, 2021.

Chollet, F. et al. Keras. https://keras.io, 2015.

Courty, N., Flamary, R., and Tuia, D. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 274–289. Springer, 2014.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9): 1853–1865, 2016.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. Joint distribution optimal transportation for domain adaptation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Dong, J., Zhou, S., Wang, B., and Zhao, H. Algorithms and theory for supervised gradual domain adaptation. *arXiv preprint arXiv:2204.11644*, 2022.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Farshchian, A., Gallego, J. A., Cohen, J. P., Bengio, Y., Miller, L. E., and Solla, S. A. Adversarial domain adaptation for stable brain-machine interfaces. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hyx6Bi0qYm.

Gadermayr, M., Eschweiler, D., Klinkhammer, B. M., Boor, P., and Merhof, D. Gradual domain adaptation for segmenting whole slide images showing pathological variability. In *International Conference on Image and Signal Processing*, pp. 461–469. Springer, 2018.

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

Ginosar, S., Rakelly, K., Sachs, S., Yin, B., and Efros, A. A. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1–7, 2015.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *CVPR*, pp. 15262–15271, 2021.

Hoffman, J., Darrell, T., and Saenko, K. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 867–874, 2014.

Huang, X., Shi, L., and Suykens, J. A. Ramp loss linear programming support vector machine. *The Journal of Machine Learning Research*, 15(1):2185–2211, 2014.

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

Kantorovich, L. V. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1939.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Kumar, A., Ma, T., and Liang, P. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pp. 5468–5479. PMLR, 2020.

Kuznetsov, V. and Mohri, M. Generalization bounds for time series prediction with non-stationary processes. In *International conference on algorithmic learning theory*, pp. 260–274. Springer, 2014.

Kuznetsov, V. and Mohri, M. Learning theory and algorithms for forecasting non-stationary time series. In *NIPS*, pp. 541–549. Citeseer, 2015.

Kuznetsov, V. and Mohri, M. Time series prediction and online learning. In *Conference on Learning Theory*, pp. 1190–1213. PMLR, 2016.

Kuznetsov, V. and Mohri, M. Generalization bounds for non-stationary mixing processes. *Machine Learning*, 106 (1):93–117, 2017.

Kuznetsov, V. and Mohri, M. Discrepancy-based theory and algorithms for forecasting non-stationary time series. *Annals of Mathematics and Artificial Intelligence*, 88(4): 367–399, 2020.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013.

Liang, J., He, R., Sun, Z., and Tan, T. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *CVPR*, pp. 2975–2984, 2019.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pp. 6028–6039, 2020.

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

Littlestone, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Malinin, A., Band, N., Gal, Y., Gales, M., Ganshin, A., Chesnokov, G., Noskov, A., Ploskonosov, A., Prokhorenkova, L., Provilkov, I., Raina, V., Raina, V., Roginskiy, D., Shmatova, M., Tigas, P., and Yangel, B. Shifts: A dataset of real distributional shift across multiple large-scale tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks*

Track (Round 2), 2021. URL https://openreview.net/forum?id=qM45LHaWM6E.

Peyré, G., Cuturi, M., et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. Meta pseudo labels. In *CVPR*, pp. 11557–11568, 2021.

Rakhlin, A. and Sridharan, K. Statistical learning and sequential prediction, 2014.

Rakhlin, A., Sridharan, K., and Tewari, A. Online learning: Random averages, combinatorial parameters, and learnability. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Rakhlin, A., Sridharan, K., and Tewari, A. Online learning via sequential complexities. *J. Mach. Learn. Res.*, 16(1): 155–186, 2015.

Redko, I., Courty, N., Flamary, R., and Tuia, D. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 849–858. PMLR, 2019.

Sagawa, S., Koh, P. W., Lee, T., Gao, I., Xie, S. M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., Beery, S., David, E., Stavness, I., Guo, W., Leskovec, J., Saenko, K., Hashimoto, T., Levine, S., Finn, C., and Liang, P. Extending the WILDS benchmark for unsupervised adaptation. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. URL https://openreview.net/forum?id=2EhHKKXMbG0.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 596–608. Curran Associates, Inc., 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation, 2016.

Talagrand, M. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Van Engelen, J. E. and Hoos, H. H. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 1999.

Villani, C. *Optimal transport: old and new*, volume 338. Springer, 2009.

Wang, H., He, H., and Katabi, D. Continuously indexed domain adaptation. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9898–9907. PMLR, 13–18 Jul 2020.

Wulfmeier, M., Bewley, A., and Posner, I. Incremental adversarial domain adaptation for continually changing environments. In *2018 IEEE International conference on robotics and automation (ICRA)*, pp. 4489–4495. IEEE, 2018.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995.

Ye, M., Jiang, R., Wang, H., Choudhary, D., Du, X., Bhushanam, B., Mokhtari, A., Kejariwal, A., and qiang liu. Future gradient descent for adapting the temporal shifting data distribution in online recommendation system. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

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

Zhao, Y., Qiao, Z., Xiao, C., Glass, L., and Sun, J. Pyhealth: A python library for health predictive models. *arXiv preprint arXiv:2101.04209*, 2021.

Zhou, S., Zhao, H., Zhang, S., Wang, L., Chang, H., Wang, Z., and Zhu, W. Online continual adaptation with active self-training. In *AISTATS*. PMLR, 2022.

Zou, Y., Yu, Z., Kumar, B. V., and Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pp. 289–305, 2018.

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

A Proof

A.1 Proof of Lemma 1 (Error Difference over Shifted Domains)

Restatement of Lemma 1. *Consider two arbitrary measures μ, ν over $\mathcal{X} \times \mathcal{Y}$. Then, for arbitrary classifier h and loss function l satisfying Assumption 2, 3, the population loss of h on μ and ν satisfies*

$$|\varepsilon_\mu(h) - \varepsilon_\nu(h)| \le \rho\sqrt{R^2+1}\, W_p(\mu, \nu) \tag{21}$$

where W_p is the Wasserstein-p distance metric and $p \ge 1$.

Proof. The population error difference of h over the two domains (i.e., μ and ν is

$$\begin{aligned} |\varepsilon_\mu(h) - \varepsilon_\nu(h)| &= |\mathbb{E}_{x,y\sim\mu}[\ell(h(x), y)] - \mathbb{E}_{x',y'\sim\nu}[\ell(h(x'), y')]| \\ &= \left|\int \ell(h(x), y) d\mu - \int \ell(h(x'), y') d\nu\right| \end{aligned} \tag{22}$$

Let γ be an arbitrary coupling of μ and ν, i.e., it is a joint distribution with marginals as μ and ν. Then, (22) can be re-written and bounded as

$$|\varepsilon_\mu(h) - \varepsilon_\nu(h)| = \left|\int \ell(h(x), y) - \int \ell(h(x'), y') d\gamma\right| \tag{23}$$

$$\text{(triangle inequality)} \le \int \left|\ell(h(x), y) - \int \ell(h(x'), y')\right| d\gamma \tag{24}$$

$$(\ell \text{ is } \rho\text{-Lipschitz}) \le \int \rho\left(\|h(x) - h(x')\| + \|y - y'\|\right) d\gamma \tag{25}$$

$$(h \text{ is } R\text{-Lipschitz}) \le \int \rho R\|x - x'\| + \rho\|y - y'\| d\gamma \tag{26}$$

$$(R > 0) \le \int \rho\sqrt{R^2+1}\left(\|x - x'\| + \|y - y'\|\right) d\gamma \tag{27}$$

Since γ is an arbitrary coupling, we know that

$$|\varepsilon_\mu(h) - \varepsilon_\nu(h)| \le \inf_\gamma \int \rho\sqrt{R^2+1}\left(\|x - x'\| + \|y - y'\|\right) d\gamma \tag{28}$$

$$= \rho\sqrt{R^2+1} W_1(\mu, \nu) \tag{29}$$

Since the Wasserstein distance W_p is monotonically increasing for $p \ge 1$, we have the following bound,

$$|\varepsilon_\mu(h) - \varepsilon_\nu(h)| \le \rho\sqrt{R^2+1} W_1(\mu, \nu) \le \rho\sqrt{R^2+1} W_p(\mu, \nu) \le \rho\sqrt{R^2+1} \tag{30}$$

□

A.2 Proof of Proposition 1 (Algorithm Stability)

Restatement of Proposition 1. *Consider two arbitrary measures μ, ν, and denote S as a set of n unlabelled samples i.i.d. drawn from μ. Suppose $h \in \mathcal{H}$ is a pseudo-labeler that provides pseudo-labels for samples in S. Define $\hat{h} \in \mathcal{H}$ as an ERM solution fitted to the pseudo-labels,*

$$\hat{h} = \arg\min_{f \in \mathcal{H}} \sum_{x \in S} \ell(f(x), h(x)) \tag{31}$$

Then, for any $\delta \in (0, 1)$, the following bound holds true with probability at least $1 - \delta$,

$$\left|\varepsilon_\mu(\hat{h}) - \varepsilon_\nu(h)\right| \le \mathcal{O}\left(W_p(\mu, \nu) + \frac{\rho B + \sqrt{\log\frac{1}{\delta}}}{\sqrt{n}}\right) \tag{32}$$

Proof. Define $\widehat{\varepsilon}_\mu(h) := \frac{1}{|S|}\sum_{x \in S} \ell(h(x), y)$ as the empirical loss over the dataset S, where S consists of samples i.i.d. drawn from $\mu(X)$ and y is the ground truth label of x.

Then, we have the following sequence of inequalities:

$$\begin{aligned}
\text{(Use Lemma A.1 of Kumar et al. (2020))} \quad \varepsilon_\mu(h) &\le \widehat{\varepsilon}_\mu(\hat{h}) + \mathcal{O}\left(R_n(\ell \circ \mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{n}}\right) \\
\left(\text{since } h(x) = \hat{h}(x)\ \forall x \in S\right) \quad &= \widehat{\varepsilon}_\mu(h) + \mathcal{O}\left(R_n(\ell \circ \mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{n}}\right) \\
\text{(Use Lemma A.1 of Kumar et al. (2020) again)} \quad &\le \varepsilon_\mu(h) + \mathcal{O}\left(2R_n(\ell \circ \mathcal{H}) + 2\sqrt{\frac{\log(1/\delta)}{n}}\right) \\
\text{(By Lemma 1)} &\le \varepsilon_\nu(h) + \rho\sqrt{R^2+1}W_p(\mu,\nu) \\
&\quad + \mathcal{O}\left(R_n(\ell \circ \mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{n}}\right) \\
\text{(By Talagrand's lemma with Assumption 3,4)} &\le \varepsilon_\nu(h) + \rho\sqrt{R^2+1}W_p(\mu,\nu) \\
&\quad + \mathcal{O}\left(\frac{\rho B}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right) \\
&\le \varepsilon_\nu(h) + \mathcal{O}\left(W_p(\mu,\nu) + \frac{\rho B}{\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right)
\end{aligned}$$

For the step using Talagrand's lemma (Talagrand, 1995), the proof of Lemma A.1 of Kumar et al. (2020) also involves an identical step, thus we do not replicate the specific details here. □

A.3 Proof of Lemma 2 (Discrepancy Bound)

Restatement of Lemma 2. *With Lemma 1, the discrepancy measure* (13) *can be upper bounded as*

$$\operatorname{disc}(\mathbf{q}_t) \le \rho\sqrt{R^2+1}\sum_{\tau=0}^{t-1} q_\tau \cdot (t-\tau-1)\Delta \tag{33}$$

Choosing $\mathbf{q}_t = \mathbf{q}_t^* = (\frac{1}{t}, ..., \frac{1}{t})$, *this upper bound can be minimized as*

$$\operatorname{disc}(\mathbf{q}_t^*) \le \rho\sqrt{R^2+1}\, t\Delta/2 = \mathcal{O}(t\Delta) \tag{34}$$

Proof. Within our setup of gradual self-training,

$$\begin{aligned}
\operatorname{disc}(\mathbf{q}_t) &= \sup_{h\in\mathcal{H}}\left(\varepsilon_{t-1}(h) - \sum_{\tau=0}^{t-1} q_\tau \cdot \varepsilon_\tau(h)\right) \\
&= \sup_{h\in\mathcal{H}}\left(\sum_{\tau=0}^{t-1} q_\tau \left(\varepsilon_{t-1}(h) - \varepsilon_\tau(h)\right)\right) \\
&\le \sup_{h\in\mathcal{H}}\left(\sum_{\tau=0}^{t-1} q_\tau |\varepsilon_{t-1}(h) - \varepsilon_\tau(h)|\right) \\
\text{(By Lemma 1)} &\le \rho\sqrt{R^2+1}\sum_{\tau=0}^{t-1} q_\tau \cdot (t-\tau-1)\Delta
\end{aligned}$$

With $\mathbf{q}_t = \mathbf{q}_t^* = (\frac{1}{t}, ..., \frac{1}{t})$, this bound becomes

$$\operatorname{disc}(\mathbf{q}_t^*) \le \rho\sqrt{R^2+1}\sum_{\tau=0}^{t-1} q_\tau \cdot (t-\tau-1)\Delta = \rho\sqrt{R^2+1}\,\frac{t}{2}\Delta = \mathcal{O}(t\Delta)$$

and it is trivial to show that this upper bound is smaller than any other $\mathbf{q}_t$ with $\mathbf{q}_t \neq \mathbf{q}_t^*$. □

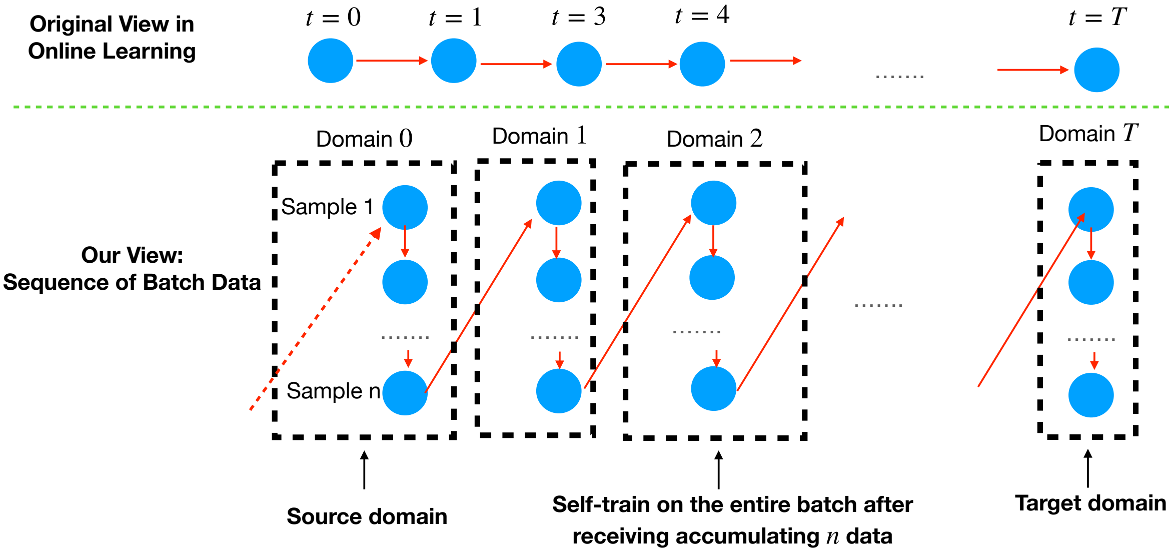

Original View in Online Learning
$t = 0$
$t = 1$
$t = 3$
$t = 4$
$t = T$
Our View: Sequence of Batch Data
Domain 0
Domain 1
Domain 2
Domain T
Sample 1
Sample n
Source domain
Self-train on the entire batch after receiving accumulating n data
Target domain

Figure 4. Our reductive view of gradual self-training that is helpful to Theorem 1.

A.4 Proof of Theorem 1 (Generalization Bound for Gradual Self-Training))

Restatement of Theorem 1. *For any $\delta \in (0, 1)$, the population loss of gradually self-trained classifier h_T in the target domain is upper bounded with probability at least $1 - \delta$ as*

$$\varepsilon_T(h_T) \leq \sum_{t=0}^{T} q_t \varepsilon_t(h_t) + \|\mathbf{q}_{n(T+1)}\|_2 \left(1+\mathcal{O}\left(\sqrt{\log(1/\delta)}\right)\right) + \text{disc}(\mathbf{q}_{T+1}) + \mathcal{O}\left(\sqrt{\log T} \mathcal{R}^{\text{seq}}_{n(T+1)}(\ell \circ \mathcal{H})\right)$$

For the class of neural nets considered in Example 2,

$$\varepsilon_T(h_T) \leq \varepsilon_0(h_0) + \mathcal{O}\left(T\Delta + \frac{T}{\sqrt{n}} + T\sqrt{\frac{\log 1/\delta}{n}} + \frac{1}{\sqrt{nT}} + \sqrt{\frac{(\log nT)^{3L-2}}{nT}} + \sqrt{\frac{\log 1/\delta}{nT}}\right) \tag{35}$$

A Reductive View of the Learning Process of Gradual Self-Training If we directly apply Corollary 2 of Kuznetsov & Mohri (2020), we can obtain a generalization bound as

$$\begin{aligned}\varepsilon_{\mu_T}(h) &\leq \sum_{t=0}^{T} q_t \varepsilon_{\mu_t}(h) + \text{disc}(\mathbf{q}_{T+1}) + \|\mathbf{q}_{T+1}\|_2 + 6M\sqrt{4\pi \log T} \mathcal{R}^{\text{seq}}_T(\ell \circ \mathcal{H}) \\ &\quad + M\|\mathbf{q}_{T+1}\|_2 \sqrt{8 \log \frac{1}{\delta}} \\ &\leq \sum_{t=0}^{T} q_t \varepsilon_{\mu_t}(h) + O(T\Delta) + \mathcal{O}(\frac{1}{\sqrt{T}}) + 6M\sqrt{4\pi \log T} \mathcal{R}^{\text{seq}}_T(\ell \circ \mathcal{H}) + \mathcal{O}(\sqrt{\frac{\log \frac{1}{\delta}}{T}})\end{aligned} \tag{36}$$

where M is an upper bound on the loss (Lemma 3 proves such a M exists), and the last inequality is obtained by setting $\mathbf{q}_{T+1} = \mathbf{q}^*_{T+1} = (\frac{1}{T+1}, \ldots, \frac{1}{T+1})$.

A typical generalization bound involves terms with dependence on N (the training set size), usually in the form $\mathcal{O}(\sqrt{\frac{1}{N}})$, and these terms vanish in the infinite-sample limit (i.e., $N \to \infty$). These terms also appear in standard generalization bounds of unsupervised domain adaptation (Ben-David et al., 2007; Zhao et al., 2019), where N becomes the number of available unlabelled data in the target domain.

In the case of gradual domain adaptation, the total number of available unlabelled is Tn, and we would expect Tn will appear in a form similar to $\mathcal{O}(\sqrt{\frac{1}{nT}})$, which vanishes in the infinite-sample limit (i.e., $nT \to \infty$). However, the generalization bound (1) has terms $\mathcal{O}(\sqrt{\frac{1}{T}})$ and $\mathcal{O}(\sqrt{\frac{\log \frac{1}{\delta}}{T}})$, which does not vanish even with infinite data per domain, i.e., $n \to \infty$ (certainly results in $Tn \to \infty$).

We attribute this issue to the coarse-grained nature of online learning analyses such as Kuznetsov & Mohri (2016; 2020), which do not take data size per domain into consideration.

To address this issue, we propose a novel reductive view of the entire learning process of gradual self-training, leading to a more fined-grained generalization bound than Eq. (36).

We draw a diagram to illustrate this reductive view in Fig. 4. Specifically, instead of viewing each domain as the smallest element, we zoom in to the sample-level and view each sample as the smallest element of the learning process. We view the gradual self-training algorithm as follows: it has a fixed data buffer of size n, and each newly observed sample is pushed to the buffer; the model updates itself by self-training once the buffer is full; after the update, the buffer is emptied. Notice that this view does not alter the learning process of gradual self-training.

With this reductive view, the learning process of gradual self-training consists of nT smallest elements (i.e., each sample is a smallest element), instead of T elements (i.e., each domain is a smallest element) in the view of online learning works (Kuznetsov & Mohri, 2016; 2020). As a result, terms of order $\mathcal{O}\left(\sqrt{\frac{1}{T}}\right)$ in (36) becomes $\mathcal{O}\left(\sqrt{\frac{1}{nT}}\right)$, and terms of order $\mathcal{O}\left(\frac{T}{n}\right)$ also vanish as $n \to \infty$. Notably, the upper bounds on the terms $\sum_{t=0}^{T} q_t \varepsilon_{\mu_t}(h)$ and $\mathrm{disc}(\mathbf{q}_{T+1})$ in (12) do not become larger with this view, since there is no distribution shift within each domain (e.g., the learning process over the first n samples in Fig. 4 does not involve any distribution shift, and the iteration $n-1 \mapsto n$ incurs a distribution shifts, since the $(n-1)$-th sample is in the first domain while the n-th sample is in the second domain).

With this reductive view, we can finally obtain a tighter generalization bound for gradual self-training without the issues mentioned previously.

Proof. With the inductive view introduced above, we can improve the naive bound (36) to

$$\begin{aligned}
\varepsilon_{\mu_T}(h_T) &\leq \sum_{t=0}^{T}\sum_{i=0}^{n-1} q_{nt+i}\varepsilon_{\mu_t}(h_T) + \mathrm{disc}(\mathbf{q}_{n(T+1)}) + \|\mathbf{q}_{n(T+1)}\|_2 + 6M\sqrt{4\pi\log nT}\mathcal{R}_{nT}^{\mathrm{seq}}(\ell\circ\mathcal{H}) \\
&\quad + M\|\mathbf{q}_{n(T+1)}\|_2\sqrt{8\log\frac{1}{\delta}} \\
&\leq \frac{1}{T+1}\sum_{t=0}^{T}\varepsilon_{\mu_t}(h_T) + \rho\sqrt{R^2+1}\,\frac{T+1}{2}\Delta + \frac{1}{\sqrt{nT}} + 6M\sqrt{4\pi\log nT}R_{nT}^{\mathrm{seq}}(\ell\circ\mathcal{H}) \\
&\quad + M\sqrt{\frac{8\log 1/\delta}{nT}} \\
&\leq \varepsilon_{\mu_0}(h_0) + \mathcal{O}\left(T\Delta + T\sqrt{\frac{\log 1/\delta}{n}} + \frac{1}{\sqrt{nT}} + \rho R\sqrt{\frac{(\log nT)^7}{nT}} + \sqrt{\frac{\log 1/\delta}{nT}}\right)
\end{aligned} \tag{37}$$

where $\mathbf{q}_{n(T+1)}$ is taken as $\mathbf{q}_{n(T+1)} = \mathbf{q}^*_{n(T+1)} = (\frac{1}{n(T+1)}), \ldots, \frac{1}{n(T+1)})$. We used the following facts when deriving the inequalities above:

- The first term of (37) has the following bound

$$\begin{aligned}
\sum_{t=0}^{T}\sum_{i=0}^{n-1} q_{nt+i}\varepsilon_{\mu_t}(h_T) &= \frac{1}{T+1}\sum_{t=0}^{T}\varepsilon_{\mu_t}(h_T) \\
&\leq \varepsilon_{\mu_0}(h_0) + \mathcal{O}(T\Delta) + \mathcal{O}\left(\frac{1}{\sqrt{n}} + T\sqrt{\frac{\log 1/\delta}{n}}\right)
\end{aligned} \tag{38}$$

which is obtained by recursively apply Lemma 1 and Proposition 1 to each term in the summation. For example, the last term in $\sum_{t=0}^{T} \varepsilon_{\mu_t}(h_T)$ can bounded by Proposition 1 as follows

$$\begin{aligned}\text{(By Proposition 1)} \quad \varepsilon_T(h_T) &\le \varepsilon_{\mu_{T-1}}(h_{T-1}) + \mathcal{O}\left(W_p(\mu_T, \mu_{T-1}) + \frac{1}{\sqrt{n}} + \sqrt{\frac{\log 1/\delta}{n}}\right)\\ \text{(Same as the above step)} &\le \dots\\ &\le \varepsilon_{\mu_0}(h_0) + \mathcal{O}(T\Delta + \mathcal{O}\left(T\sqrt{\frac{\log 1/\delta}{n}}\right)\end{aligned} \tag{39}$$

and the second last term can be bounded similarly with the additional help of Lemma 1

$$\begin{aligned}\text{(By Lemma 1)} \quad \varepsilon_{T-1}(h_T) &\le \varepsilon_{\mu_T}(h_T) + \mathcal{O}(W_p(\mu_T, \mu_{T-1}))\\ \text{(Apply Eq. (39))} &\le \varepsilon_{\mu_0}(h_0) + T\Delta + \mathcal{O}\left(T\sqrt{\frac{\log 1/\delta}{n}}\right)\end{aligned}$$

All the rest terms (i.e., $\varepsilon_{T-2}(h_T), \dots, \varepsilon_0(h_T)$) can be bounded in the same way.

- The second term of (37) can be bounded by applying Lemma 2.
- The value of $R^{\text{seq}}_{nT}(\ell \circ \mathcal{H})$ can be bounded by combining Lemma 4 and Example 2.

□

A.5 Helper Lemmas

Lemma 3 (Bounded Loss). *For any $x \in \mathcal{X}, y \in \mathcal{Y}, h \in \mathcal{H}$, the loss $\ell(x, y)$ is upper bounded by some constant M, i.e., $l(h(x), y) \le M$.*

Proof. Notice that i) the input x is bounded in a compact space, specifically, $\|x\|_2 \le 1$ (ensured by Assumption 1), ii) y lives in a compact space in $\mathbb{R}$ (defined in Sec. 3.1), iii) the hypothesis $h \in \mathcal{H}$ is R-Lipschitz, and iv) the loss function ℓ is ρ-Lipschitz.

Combining these conditions, one can easily find that there exists a constant M such that $l(h(x), y)$ for any $x \in \mathcal{X}, y \in \mathcal{Y}, h \in \mathcal{H}$. □

Lemma 4 (Lemma 14.8 of Rakhlin & Sridharan (2014)). *For ρ-Lipschitz loss function l, the sequential Rademacher complexity of the loss class $\ell \circ \mathcal{H}$ is bounded as*

$$\mathcal{R}^{\text{seq}}_T(\ell \circ \mathcal{H}) \le \mathcal{O}(\rho\sqrt{(\log T)^3})\mathcal{R}^{\text{seq}}_T(\mathcal{H}) \tag{40}$$

Proof. See Rakhlin & Sridharan (2014). □

A.6 Derivation of the Optimal T

In Sec. 4.4, we show a variant of the generalization bound in (18) as

$$\varepsilon_T(h_T) \le \varepsilon_0(h_0) + \inf_{\mathcal{P}} \widetilde{\mathcal{O}}\left(T\Delta_{\text{max}} + \frac{T}{\sqrt{n}} + \sqrt{\frac{1}{nT}}\right). \tag{41}$$

where Δ_{max} is an upper bound on the average W_p distance between any pair of consecutive domains along the path, i.e., $\Delta_{\max} \ge \frac{1}{T}\sum_{t=1}^{T} W_p(\mu_{t-1}, \mu_t)$.

Given that $T, \Delta_{\text{max}}, n$ are all positive, we know there exists an optimal $T = T^*$ that minimizes the function

$$f(T) := T\Delta_{\text{max}} + \frac{T}{\sqrt{n}} + \sqrt{\frac{1}{nT}}\ , \tag{42}$$

and one can straightforwardly derive that

$$T^* = \left(\frac{1}{2(1 + \Delta_{\text{max}}\sqrt{n}\,)}\right)^{\frac{2}{3}}\ . \tag{43}$$

Proof. The derivative of $f(T)$ is

$$f'(T) = \Delta_{\max} + \frac{1}{\sqrt{n}} - \frac{1}{2\sqrt{n}} T^{-\frac{3}{2}} \,, \tag{44}$$

and the second-order derivative of $f(T)$ is

$$f''(T) = \frac{3}{4\sqrt{n}} T^{-\frac{5}{2}} \,. \tag{45}$$

Eq. (45) indicates that $f(T)$ is strictly convex in $T \in (0, \infty)$. Then, we only need to solve for the equation

$$f'(T) = 0 \tag{46}$$

as $T \in (0, \infty)$, which gives our the solution

$$T^* = \left(\frac{1}{2(1 + \Delta_{\max} \sqrt{n}\,)} \right)^{\frac{2}{3}} \,. \tag{47}$$

□

B Experimental Details

Implementation. We adopt the code of (Kumar et al., 2020) from https://github.com/p-lambda/gradual_domain_adaptation. We make necessary modifications to include two new datasets (Color-Shift MNSIT and CoverType) to the codebase. Also, we directly use the model structure and hyper-parameters provided in this codebase.

Code. Our code is provided in

https://github.com/Haoxiang-Wang/gradual-domain-adaptation.