A Theory of Label Propagation for Subpopulation Shift

Tianle Cai [*1 2] **Ruiqi Gao** [*1 2] **Jason D. Lee** [*1] **Qi Lei** [*1]

Abstract

One of the central problems in machine learning is domain adaptation. Unlike past theoretical work, we consider a new model for subpopulation shift in the input or representation space. In this work, we propose a provably effective framework for domain adaptation based on label propagation. In our analysis, we use a simple but realistic expansion assumption, proposed in Wei et al. (2021). Using a teacher classifier trained on the source domain, our algorithm not only propagates to the target domain but also improves upon the teacher. By leveraging existing generalization bounds, we also obtain end-to-end finite-sample guarantees on the entire algorithm. In addition, we extend our theoretical framework to a more general setting of source-to-target transfer based on a third unlabeled dataset, which can be easily applied in various learning scenarios. Inspired by our theory, we adapt consistency-based semi-supervised learning methods to domain adaptation settings and gain significant improvements.

1. Introduction

The recent success of supervised deep learning is built upon two crucial cornerstones: That the training and test data are drawn from an *identical distribution*, and that representative *labeled* data are available for training. However, in real-world applications, labeled data drawn from the same distribution as test data are usually unavailable. Domain adaptation (Quionero-Candela et al., 2009; Saenko et al., 2010) suggests a way to overcome this challenge by transferring the knowledge of labeled data from a source domain to the target domain.

Without further assumptions, the transferability of information is not possible. Existing theoretical works have investigated suitable assumptions that can provide learning guarantees. Many of the works are based on the *covariate shift* assumption (Heckman, 1979; Shimodaira, 2000), which states that the conditional distribution of the labels (given the input x) is invariant across domains, i.e., $p_S(y|x) = p_T(y|x)$. Traditional approaches usually utilize this assumption by further assuming that the source domain covers the support of the target domain. In this setting, importance weighting (Shimodaira, 2000; Cortes et al., 2010; 2015; Zadrozny, 2004) can be used to transfer information from source to target with theoretical guarantees. However, the assumption of covered support rarely holds in practice.

In the seminal works of Ben-David et al. (2010); Ganin et al. (2016), the authors introduced a theory that enables generalization to out-of-support samples via distribution matching. They showed that the risk on the target domain can be bounded by the sum of two terms a) the risk on the source domain plus a discrepancy between source and target domains, and b) the optimal joint risk that a function in the hypothesis class can achieve. Inspired by this bound, numerous domain-adversarial algorithms aimed at matching the distribution of source and target domains in the feature space have been proposed (Ajakan et al., 2014; Long et al., 2015; Ganin et al., 2016). These methods show encouraging empirical performance on transferring information from domains with different styles, e.g., from colorized photos to gray-scale photos. However, the theory of distribution matching can be violated since only two terms in the bound are optimized in the algorithms while the other term can be arbitrary large (Zhao et al., 2019a; Wu et al., 2019; Li et al., 2020). In practice, forcing the representation distribution of two domains to match may also fail in some settings. As an example, Li et al. (2020) gives empirical evidence of this failure on datasets with subpopulation shift. Li et al. (2020) describes a classification task between vehicle and person; subpopulation shift happens when the source vehicle class contains 50% car and 50% motorcycle, while the target vehicle class contains 10% car and 90% motorcycle.

In real-world applications, subpopulation shift is pervasive, and often in a fine-grained manner. The source domain will inevitably fail to capture the diversity of the target domain, and models will encounter unseen subpopulations in the target domain, e.g., unexpected weather conditions for self-driving or different diagnostic setups in medical applications (Santurkar et al., 2021). The lack of theoretical

[*] Alphabetical order. [1] Princeton University [2] Zhongguancun Haihua Institute for Frontier Information Technology. Correspondence to: Jason D. Lee <jasonlee@princeton.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

understanding of subpopulation shift motivates us to study the following question:

How to provably transfer from source to target domain under subpopulation shift using unlabeled data?

To address this question, we develop a general framework of domain adaptation where we have a supervision signal on the source domain (through a teacher classifier which has non-trivial performance on the source domain but is allowed to be entirely wrong on the target domain (See Assumption 1(a) and Figure 1)) and unlabeled data on both source and target domains. The key of the analysis is to show that the supervision signal can be propagated to the unlabeled data. To do so, we partition data from both domains into some subpopulations and leverage a simple but realistic expansion assumption (Definition 2.1) proposed in Wei et al. (2021) on the subpopulations. We then prove that by minimizing a consistency regularization term (Miyato et al., 2018; Shu et al., 2018; Xie et al., 2020) on unlabeled data from both domains plus a 0-1 consistency loss with the supervision signal (i.e., the teacher classifier) on the source domain, the supervision signal from the subpopulations of the source domain can not only be propagated to the subpopulations of target domain but also refine the prediction on the source domain. In Theorem 2.1 and 2.2, we give bounds on the test performance on the target domain. Using off-the-shelf generalization bounds, we also obtain end-to-end finite-sample guarantees for neural networks in Section 2.3.

In Section 3, we extend our theoretical framework to a more general setting with source-to-target transfer based on an additional unlabeled dataset. As long as the subpopulation components of the unlabeled dataset satisfy the expansion property and cover both the source and target subpopulation components, then one can provably propagate label information from source to target *through* the unlabeled data distribution (Theorem 3.1 and 3.2). As corollaries, we immediately obtain learning guarantees for both semi-supervised learning and unsupervised domain adaptation. The results can also be applied to various settings like domain generalization etc., see Figure 2.

We implement the popular consistency-based semi-supervised learning algorithm FixMatch (Sohn et al., 2020) on the subpopulation shift task from BREEDS (Santurkar et al., 2021), and compare it with popular distributional matching methods (Ganin et al., 2016; Zhang et al., 2019). Results show that the consistency-based method outperforms distributional matching methods by over 8%, partially verifying our theory on the subpopulation shift problem. We also show that combining distributional matching methods and consistency-based algorithm can improve the performance upon distributional matching methods on classic unsupervised domain adaptation datasets such as Office-31 (Saenko et al., 2010) and Office-Home (Venkateswara et al., 2017).

In summary, our contributions are: 1) We introduce a theoretical framework of learning under subpopulation shift through label propagation; 2) We provide accuracy guarantees on the target domain for a consistency-based algorithm using a fine-grained analysis under the expansion assumption (Wei et al., 2021); 3) We provide a generalized label propagation framework that easily includes several settings, e.g., semi-supervised learning, domain generalization, etc.

1.1. Related work

We review some more literature on domain adaptation, its variants, and consistency regularization, followed by discussions on the distinction of our contributions compared to Wei et al. (2021).

For the less challenging setting of covariate shift where the source domain covers the target domain's support, prior work regarding importance weighting focuses on estimations of the density ratio (Lin et al., 2002; Zadrozny, 2004) through kernel mean matching (Huang et al., 2006; Gretton et al., 2007; Zhang et al., 2013; Shimodaira, 2000), and some standard divergence minimization paradigms (Sugiyama et al., 2008; 2012; Uehara et al., 2016; Menon and Ong, 2016; Kanamori et al., 2011). For out-of-support domain adaptation, recent work investigate approaches to match the source and target distribution in representation space (Glorot et al., 2011; Ajakan et al., 2014; Long et al., 2015; Ganin et al., 2016). Practical methods involve designing domain adversarial objectives (Tzeng et al., 2017; Long et al., 2017a; Hong et al., 2018; He and Zhang, 2019; Xie et al., 2019; Zhu et al., 2019) or different types of discrepancy minimization (Long et al., 2015; Lee et al., 2019; Roy et al., 2019; Chen et al., 2020a). Another line of work explore self-training or gradual domain adaptation (Gopalan et al., 2011; Gong et al., 2012; Glorot et al., 2011; Kumar et al., 2020). For instance, Chen et al. (2020c) demonstrates that self-training tends to learn robust features in some specific probabilistic setting.

Variants of domain adaptation have been extensively studied. For instance, weakly-supervised domain adaptation considers the case where the labels in the source domain can be noisy (Shu et al., 2019; Liu et al., 2019); multi-source domain adaptations adapts from multiple source domains (Xu et al., 2018; Zhao et al., 2018); domain generalization also allows access to multiple training environments, but seeks out-of-distribution generalization *without* prior knowledge on the target domain (Ghifary et al., 2015; Li et al., 2018; Arjovsky et al., 2019; Ye et al., 2021).

The idea of consistency regularization has been used in many settings. Miyato et al. (2018); Qiao et al. (2018); Xie et al. (2020) enforce consistency with respect to adversarial

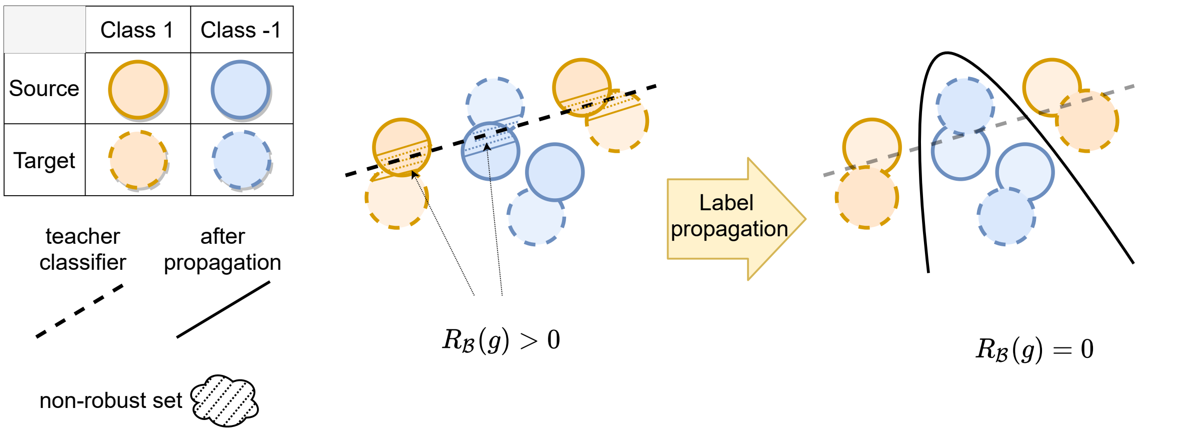


Figure 1. A toy illustration of our framework on label propogation on subpopulations, formalized in Section 2. Although the formal definition (Assumption 1) involves a neighborhood function $\mathcal{B}(\cdot)$ and possibly a representation space, one can understand it by the above toy model: a set of S_i and T_i where each $S_i \cup T_i$ forms a regular connected component. The consistency loss $R_{\mathcal{B}}(g)$ measures the amount of non-robust set of g, which contains points whose predictions by g is inconsistent in a small neighborhood. Our main theorems (Theorem 2.1 and 2.2) state that, starting from a teacher with information on the source data, consistency regularization (regularizing $R_{\mathcal{B}}(g)$ on unlabeled data) can result in the propogation of label information, thereby obtaining a good classifier on the target domain, which may also improve upon the accuracy of the teacher on the source domain.

examples or data augmentations for semi-supervised learning. Shu et al. (2019) combines domain adversarial training with consistency regularization for unsupervised domain adaptation. Recent work on self-supervised learning also leverages the consistency between two aggressive data augmentations to learn meaningful features (Chen et al., 2020b; Grill et al., 2020; Caron et al., 2020).

Most closely related to our work is Wei et al. (2021), which introduces a simple but realistic "expansion" assumption to analyze label propagation, which states that a low-probability subset of the data must expand to a neighborhood with larger probability relative to the subset. Under this assumption, the authors show learning guarantees for unsupervised learning and semi-supervised learning.

The focus of Wei et al. (2021) is not on domain adaptation, though the theorems directly apply. This leads to several drawbacks that we now discuss. Notably, in the analysis of Wei et al. (2021) for unsupervised domain adaptation, the population test risk is bounded using the population risk of a pseudo-labeler on the *target domain*.[1] The pseudo-labeler is obtained via training with labeled data on the *source domain*. For domain adaptation, we do not expect such a pseudo-labeler to be directly informative when applied to the target domain, especially when the distribution shift is severe. In contrast, our theorem does not rely on a good pseudo-labeler on the target domain. Instead, we prove that with only supervision on the source domain, the population risk on the target domain can converge to zero as the value of the consistency regularizer of the ground truth classifier decreases (Theorem 2.1, 2.2). In addition, Wei et al. (2021) assumes that the probability mass of *each class* together satisfies the expansion assumption. However, each class may consist of several disjoint subpopulations. For instance, the dog class may have different breeds as its subpopulations. This setting differs from the concrete example of the Gaussian mixture model shown in Wei et al. (2021) where the data of each class concentrate following a Gaussian distribution. In this paper, we instead take a more realistic usage of the expansion assumption by assuming expansion property on the *subpopulations* of each class (Assumption 1). Behind this relaxation is a fine-grained analysis of the probability mass's expansion property, which may be of independent interest.

2. Label Propogation in Domain Adaptation

In this section, we consider label propagation for unsupervised domain adaptation. We assume the distributions' structure can be characterized by a specific subpopulation shift with the expansion property. In Section 2.1, we introduce the setting, including the algorithm and assumptions. In Section 2.2, we present the main theorem on bounding the target error. In Section 2.3, we provide an end-to-end guarantee on the generalization error of adapting a deep neural network to the target distribution with finite data. In Section 2.4 we provide a proof sketch for the theorems.

[1] In the new version of Wei et al. (2021), the authors proposed a refined result based on iterative training, which alleviates the error bound's dependency on the error of the target domain pseudo-labeler. Still, a mostly correct pretrained pseudo-labeler is required.

2.1. Setting

We consider a multi-class classification problem $\mathcal{X} \to \mathcal{Y} = \{1, \cdots, K\}$. Let S and T be the source and target distribution on $\mathcal{X}$ respectively, and we wish to find a classifier $g : \mathcal{X} \to \mathcal{Y}$ that performs well on T. Suppose we have a teacher classifier g_{tc} on S. The teacher g_{tc} can be obtained by training on the labeled data on S (standard unsupervised domain adaptation), or by training on a small subset of labeled data on S, or by direct transferring from some other trained classifier, etc. In all, the teacher classifier represents all label information we know (and is allowed to have errors). Our goal is to transfer the information in g_{tc} onto T using only unlabeled data.

Our setting for subpopulation shift is formulated in the following assumption.

Assumption 1. *Assume the source and target distributions have the following structure:* $\mathrm{supp}(S) = \cup_{i=1}^m S_i$, $\mathrm{supp}(T) = \cup_{i=1}^m T_i$, *where* $S_i \cap S_j = T_i \cap T_j = S_i \cap T_j = \emptyset$ *for all* $i \neq j$. *We assume the ground truth class* $g^*(x)$ *for* $x \in S_i \cup T_i$ *is consistent (constant), which is denoted as* $y_i \in \{1, \cdots, K\}$. *We abuse the notation to let* S_i, T_i *also denote the conditional distribution (probability measure) of* S, T *on the set* S_i, T_i *respectively. In addition, we make the following canonical assumptions:*

1. *The teacher classifier on* S_i *is informative of the ground truth class* y_i *by a margin* $\gamma > 0$*, that is,*

$$\mathbb{P}_{x \sim S_i}[g_{tc}(x) = y_i] \geq \mathbb{P}_{x \sim S_i}[g_{tc}(x) = k] + \gamma, \\ \forall k \in \{1, \cdots, K\} \backslash \{y_i\}.$$

2. *On each component, the ratio of the population under domain shift is upper-bounded by a constant* r*, i.e.*

$$\frac{\mathbb{P}_T[T_i]}{\mathbb{P}_S[S_i]} \leq r, \forall i \in \{1, \cdots, m\}.$$

Following Wei et al. (2021), we make use of a consistency regularization method, i.e. we expect the predictions to be stable under a suitable set of input transformations $\mathcal{B}(x) \subset \mathcal{X}$. The regularizer of g on the mixed probability measure $\frac{1}{2}(S+T)$ is defined as

$$R_{\mathcal{B}}(g) := \mathbb{P}_{x \sim \frac{1}{2}(S+T)}[\exists x' \in \mathcal{B}(x), \text{s.t. } g(x) \neq g(x')],$$

and a low regularizer value implies the labels are with high probability constant within $\mathcal{B}(x)$. Prior work on using consistency regularization for unlabeled self-training includes Miyato et al. (2018) where $\mathcal{B}(\cdot)$ can be understood as a distance-based neighborhood set and Adel et al. (2017); Xie et al. (2020) where $\mathcal{B}(\cdot)$ can be understood as the set of data augmentations. In general, $\mathcal{B}(x)$ takes the form $\mathcal{B}(x) = \{x' : \exists A \in \mathcal{A} \text{ such that } d(x', \mathcal{A}(x)) \leq r\}$[2] for a small number $r > 0$, some distance function d, and a class of data augmentation functions $\mathcal{A}$.

The set $\mathcal{B}(x)$ is used in the following expansion property. First, for $x \in S_i \cup T_i$ ($i \in \{1, \cdots, m\}$), we define the neighborhood function $\mathcal{N}$ as

$$\mathcal{N}(x) := (S_i \cup T_i) \cap \{x' | \mathcal{B}(x) \cap \mathcal{B}(x') \neq \emptyset\}$$

and the neighborhood of a set $A \in \mathcal{X}$ as

$$\mathcal{N}(A) := \cup_{x \in A \cap (\cup_{i=1}^m S_i \cup T_i)} \mathcal{N}(x).$$

The expansion property on the mixed distribution $\frac{1}{2}(S+T)$ is defined as follows:

Definition 2.1 (Expansion (Wei et al., 2021))**.**

1. *(Multiplicative Expansion) We say* $\frac{1}{2}(S+T)$ *satisfies* (a, c)*-multiplicative expansion for some constant* $a \in (0, 1)$, $c > 1$*, if for any* i *and any subset* $A \subset S_i \cup T_i$ *with* $\mathbb{P}_{\frac{1}{2}(S_i+T_i)}[A] \leq a$*, we have* $\mathbb{P}_{\frac{1}{2}(S_i+T_i)}[\mathcal{N}(A)] \geq \min\left(c\mathbb{P}_{\frac{1}{2}(S_i+T_i)}[A], 1\right)$.

2. *(Constant Expansion) We say* $\frac{1}{2}(S+T)$ *satisfies* (q, ξ)*-constant expansion for some constant* $q, \xi \in (0, 1)$*, if for any set* $A \subset \mathcal{X}$ *with* $\mathbb{P}_{\frac{1}{2}(S+T)}[A] \geq q$ *and* $\mathbb{P}_{\frac{1}{2}(S_i+T_i)}[A] \leq \frac{1}{2}, \forall i$*, we have* $\mathbb{P}_{\frac{1}{2}(S+T)}[\mathcal{N}(A)] \geq \min\left(\xi, \mathbb{P}_{\frac{1}{2}(S+T)}[A]\right) + \mathbb{P}_{\frac{1}{2}(S+T)}[A]$.

The expansion property implicitly states that S_i and T_i are close to each other and regularly shaped. Through the regularizer $R_{\mathcal{B}}(g)$ the label can "propagate" from S_i to T_i[3]. One can keep in mind the specific example of Figure 1 where $B(x) = \{x' : \|x - x'\|_2\} \leq r\}$ and $S_i \cup T_i$ forms a single connected component.

Finally, let G be a function class of the learning model. We consider the *realizable* case when the ground truth function $g^* \in G$. We assume that the consistency error of the ground truth function is small, and use a constant $\mu > 0$ to represent an upper bound: $R_{\mathcal{B}}(g^*) < \mu$. We find the classifier g with

[2] In this paper, consistency regularization, expansion property, and label propagation can also be understood as happening in a representation space, as long as $d(x, x') = \|h(x) - h(x')\|$ for some feature map h.

[3] Note that our model for subpopulation shift allows any fine-grained form ($m \gg K$), which makes the expansion property more realistic. In image classification, one can take for example S_i as "Poodles eating dog food" v.s. T_i as "Labradors eating meat" (they're all under the dog class), which is a rather typical form of shift in a real dataset. The representations of such subpopulations can turn out quite close after certain data augmentation and perturbations as in $\mathcal{B}(\cdot)$.

the following algorithm:

$$g = \underset{g:\mathcal{X}\to\mathcal{Y}, g\in G}{\operatorname{argmin}} L_{01}^S(g, g_{tc})$$
$$\text{s.t. } R_{\mathcal{B}}(g) \le \mu, \tag{1}$$

where $L_{01}^S(g, g_{tc}) := \mathbb{P}_{x\sim S}[g(x) \ne g_{tc}(x)]$ is the 0-1 loss on the *source domain* which encourages g to be aligned with g_{tc} on the source domain. In this paper, we are only concerned with the results of label propagation and not with the optimization process, so we simply take the solution g of (1) as found and perform analysis on g.

Our main theorem will be formulated using $(\frac{1}{2}, c)$-multiplicative expansion or (q, μ)-constant expansion[4].

2.2. Main Theorem

With the above preparations, we are ready to establish bounds on the target error $\epsilon_T(g) := \mathbb{P}_{x\sim T}(g(x) \ne g^*(x))$.

Theorem 2.1 (Bound on Target Error with Multiplicative Expansion)**.** *Suppose Assumption 1 holds and $\frac{1}{2}(S+T)$ satisfies $(\frac{1}{2}, c)$-multiplicative expansion. Then the classifier obtained by (1) satisfies*

$$\epsilon_T(g) \le \max\left(\frac{c+1}{c-1}, 3\right)\frac{8r\mu}{\gamma}.$$

Theorem 2.2 (Bound on Target Error with Constant Expansion)**.** *Suppose Assumption 1 holds and $\frac{1}{2}(S+T)$ satisfies (q, μ)-constant expansion. Then the classifier obtained by (1) satisfies*

$$\epsilon_T(g) \le (2\max(q, \mu) + \mu)\frac{8r\mu}{\gamma}.$$

We make the following remarks on the main results, and also highlight the differences from directly applying Wei et al. (2021) to domain adaptation.

Remark 1. *The theorems state that as long as the ground truth consistency error (equivalently, μ) is small enough, the classifier can achieve near-zero error. This result does not rely on the teacher being close to zero error, as long as the teacher has a positive margin γ. As a result, the classifier g can improve upon g_{tc} (including on S, as the proof of the theorems can show), in a way that the error of g converge to zero as $\mu \to 0$, regardless of the error of g_{tc}. This improvement is due the algorithmic change in Equation (1) which strongly enforces label propagation. Under multiplicative expansion, Wei et al. (2021) attain a bound of the form $O(\frac{1}{c}\text{error}(g_{tc}) + \mu)$, which explicitly depends on the accuracy of the teacher g_{tc} on the target domain.[5] The improvement is due to that we strongly enforce consistency rather than balancing consistency with teacher classifier fit.*

Remark 2. *We do not impose any lower bound on the measure of the components S_i, T_i, which is much more general and realistic. From the proofs, one may see that we allow some components to be entirely mislabeled, but in the end, the total measure of such components will be bounded. Directly applying Wei et al. (2021) would require a stringent lower bound on the measure of each S_i, T_i.*

Remark 3. *We only require expansion with respect to the individual components $S_i \cup T_i$, instead of the entire class (Wei et al., 2021), which is a weaker requirement.*

The proofs are essentially because the expansion property turns local consistency into a form of global consistency. The proof sketch is in Section 2.4, and the full proof is in Appendix A.

2.3. Finite Sample Guarantee for Deep Neural Networks

In this section, we leverage existing generalization bounds to prove an end-to-end guarantee on training a deep neural network with finite samples on S and T. The results indicate that if the ground-truth class is realizable by a neural network f^* by a large robust margin, then the total error can be small.

For simplicity let there be n i.i.d. data each from S and T (a total of $2n$ data), and the empirical distribution is denoted $\hat{S}$ and $\hat{T}$. In order to upper-bound the loss L_{01} and $R_{\mathcal{B}}(g)$, we apply a notion of all-layer margin (Wei and Ma, 2019) [6], which measures the stability of the neural net to simultaneous perturbations to each hidden layer. We first cite the useful results from Wei et al. (2021). Suppose $g(x) = \text{argmax}_{i\in\{1,\cdots,K\}} f(x)_i$ where $f : \mathcal{X} \to \mathbb{R}^K, x \mapsto W_p\phi(\cdots\phi(W_1x)\cdots)$ is the neural network with weight matrices $\{W_i\}_{i=1}^p$, [7] and q is the maximum width of any layer. Let $m(f, x, y) \ge 0$ denote the all-layer margin

[4] Wei et al. (2021) contains several examples and illustrations of the expansion property, e.g., the Gaussian mixture example satisfies $(a, c) = (0.5, 1.5)$ multiplicative expansion. The radius r in $\mathcal{B}$ is much smaller than the norm of a typical example, so our model, which requires a separation of $2r$ between components to make $R_{\mathcal{B}}(g^*)$ small, is much weaker than a typical notion of "clustering".

[5] In the new version of Wei et al. (2021), the authors proposed a refined result based on iterative training, which alleviates the error bound's dependency on the error of the target domain pseudolabeler. However, their results still require a mostly correct pseudolabler on the target domain and require the expansion constant c to be much larger than 1.

[6] Though other notions of margin can also work, this helps us to leverage the results from Wei et al. (2021).

[7] Similarly, f^* and g^* is the ground truth network and its induced classifier.

at input x for label y. [8] We also define the robust margin $m_{\mathcal{B}}(f, x) = \min_{x' \in \mathcal{B}(x)} m(f, x', \operatorname{argmax}_i f(x)_i)$. We state the following results.

Proposition 2.1 (Theorem C.3 from Wei et al. (2021))**.** *For any $t > 0$, with probability $1 - \delta$,*

$$L_{01}^{S}(g, g_{tc}) \leq \mathbb{P}_{x \sim \hat{S}}[m(f, x, g_{tc}(x)) \leq t] + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right),$$

where $\widetilde{O}(\cdot)$ hides poly-logarithmic factors in n and d.

Proposition 2.2 (Theorem 3.7 from Wei et al. (2021))**.** *For any $t > 0$, With probability $1 - \delta$,*

$$R_{\mathcal{B}}(g) \leq \mathbb{P}_{x \sim \frac{1}{2}(\hat{S}+\hat{T})}[m_{\mathcal{B}}(f, x) \leq t] + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right).$$

To ensure generalization we replace the loss functions with the margin loss in the algorithm and solve

$$g = \underset{g:\mathcal{X}\to\mathcal{Y}, g \in G}{\operatorname{argmin}} \ \mathbb{P}_{x \sim \hat{S}}[m(f, x, g_{tc}(x)) \leq t]$$
$$\text{s.t. } \mathbb{P}_{x \sim \frac{1}{2}(\hat{S}+\hat{T})}[m_{\mathcal{B}}(f, x) \leq t] \leq \mu \qquad (2)$$

where $\mu \geq \mathbb{P}_{x \sim \frac{1}{2}(\hat{S}+\hat{T})}[m_{\mathcal{B}}(f^*, x) \leq t]$. Based on these preparations, we are ready to state the final bound.

Theorem 2.3. *Suppose Assumption 1 holds, and g is returned by (2). With probability $1 - \delta$, we have:*

(a) Under $(1/2, c)$-multiplicative expansion on $\frac{1}{2}(S + T)$ we have

$$\epsilon_T(g) \leq \frac{8r}{\gamma}\left(\max\left(\frac{c+1}{c-1}, 3\right)\hat{\mu} + \Delta\right).$$

(b) Under $(q, \hat{\mu})$-constant expansion on $\frac{1}{2}(S + T)$ we have

$$\epsilon_T(g) \leq \frac{8r}{\gamma}\left(2\max\left(q, \hat{\mu}\right) + \hat{\mu} + \Delta\right).$$

where

$$\Delta = \widetilde{O}\left(\left(\mathbb{P}_{x \sim \hat{S}}[m(f^*, x, g_{tc}(x)) \leq t] - L_{01}^{\hat{S}}(g^*, g_{tc})\right) + \frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right),$$

$$\hat{\mu} = \mu + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right).$$

Remark 4. *Note that the first term in Δ is small if t is small, and as $n \to \infty$, the bounds Δ can be close to 0 and $\hat{\mu}$ can be close to μ, which gives us the bounds in Section 2.2.*

Similar to the argument in Wei et al. (2021), it is worth noting that our required sample complexity does not depend exponentially on the dimension. This is in stark contrast to classic non-parametric methods for unknown "clusters" of samples, where the sample complexity suffers the curse of dimensionality of the input space.

The proof of Theorem 2.3 is in Appendix B.

2.4. Proof Sketch for Theorem 2.1 and 2.2

To prove the theorems, we first introduce some concepts and notations.

A point $x \in \mathcal{X}$ is called *robust* w.r.t. $\mathcal{B}$ and g if for any x' in $\mathcal{B}(x)$, $g(x) = g(x')$. Denote

$$RS(g) := \{x | g(x) = g(x'), \forall x' \in \mathcal{B}(x)\},$$

which is called the *robust set* of g. Let

$$A_{ik} := RS(g) \cap (S_i \cup T_i) \cap \{x | g(x) = k\}$$

for $i \in \{1, \cdots, m\}$, $k \in \{1, \cdots, K\}$, and they form a partition of the set $RS(g)$. Denote

$$y_i^{\text{Maj}} := \underset{k \in \{1,\cdots,K\}}{\operatorname{argmax}} \ \mathbb{P}_{\frac{1}{2}(S+T)}[A_{ik}],$$

which is the majority class label of g in the robust set on $(S_i \cup T_i)$. We also call

$$M_i := \bigcup_{k \in \{1,\cdots,K\} \backslash \{y_i^{\text{Maj}}\}} A_{ik}$$

and $M := \bigcup_{i=1}^{m} M_i$ the *minority robust set* of g. In addition, let

$$\widetilde{M_i} := (S_i \cup T_i) \cap \{x | g(x) \neq y_i^{\text{Maj}}\}$$

and $\widetilde{M} := \bigcup_{i=1}^{m} \widetilde{M_i}$ be the *minority set* of g, which is superset to the minority robust set.

The expansion property can be used to control the total population of the minority set.

Lemma 2.1 (Upper Bound of Minority Set)**.** *For the classifier g obtained by (1), $\mathbb{P}_{\frac{1}{2}(S+T)}[\widetilde{M}]$ can be bounded as follows:*

(a) Under $(\frac{1}{2}, c)$-multiplicative expansion, we have $\mathbb{P}_{\frac{1}{2}(S+T)}[\widetilde{M}] \leq \max\left(\frac{c+1}{c-1}, 3\right)\mu$.

(b) Under (q, μ)-constant expansion, we have $\mathbb{P}_{\frac{1}{2}(S+T)}[\widetilde{M}] \leq 2\max\left(q, \mu\right) + \mu$.

[8] For now, we only use $m(f, x, y) = 0$ if $f(x) \neq y$, so that we can upper bound $\mathbf{1}(g(x) \neq g^*(x))$ with $\mathbf{1}\left(m(f, x, y) \geq t\right)$ for any $t > 0$. One can refer the datailed definition to Appendix B or in Wei and Ma (2019).

Based on the bound on the minority set, our next lemma says that on most subpopulation components, the inconsistency between g and g_{tc} is no greater than the error of g_{tc} plus a margin $\frac{\gamma}{2}$. Specifically, define

$$\begin{aligned} I =&\{i \in \{1, \cdots, m\} | \\ &\mathbb{P}_{x\sim S_i}[g(x) \neq g_{tc}(x)] > \mathbb{P}_{x\sim S_i}[g_{tc}(x) \neq y_i] + \frac{\gamma}{2}\} \end{aligned}$$

and we have the following result

Lemma 2.2 (Upper Bound on the Inconsistent Components I)**.** *Suppose $\mathbb{P}_{\frac{1}{2}(S+T)}[\widetilde{M}] \leq C$, then*

$$\mathbb{P}_S[\cup_{i\in I} S_i] \leq \frac{4C}{\gamma}.$$

Based on the above results, we are ready to bound the target error $\epsilon_T(g)$.

Lemma 2.3 (Bounding the Target Error)**.** *Suppose $\mathbb{P}_{\frac{1}{2}(S+T)}[\widetilde{M}] \leq C$. Let*

$$\epsilon_T^i(g) = \mathbb{P}_T[T_i]\mathbb{P}_{x\sim T}[g(x) \neq y_i]$$

for i in $\{1, \cdots, m\}$, so that $\epsilon_T(g) = \sum_{i=1}^m \epsilon_T^i(g)$. Then we can separately bound

(a) $\sum_{i\in I} \epsilon_T^i(g) \leq \frac{4rC}{\gamma}$

(b) $\sum_{i\in\{1,\cdots,m\}\backslash I} \epsilon_T^i(g) \leq \frac{4rC}{\gamma}$

so that the combination gives

$$\epsilon_T(g) \leq \frac{8rC}{\gamma}.$$

Specically, Lemma 2.3(a) is obtained by directly using Lemma 2.2, and Lemma 2.3(b) is proved by a fine-grained analysis on the minority set.

Finally, we can plug in C from Lemma 2.1 and the desired main results are obtained.

3. Label Propogation in Generalized Subpopulation Shift

In this section, we show that the previous label propagation algorithm can be applied to a much more general setting than standard unsupervised domain adaptation. In a word, as long as we perform consistency regularization on an unlabeled dataset that covers both the teacher classifier's domain and the target domain, we can perform label propagation through the subpopulation of the unlabeled data.

Specifically, we still let S be the source distribution where we have a teacher g_{tc} on, and T is the target distribution. The difference is that we have a "covering" distribution U (Assumption 2(c)) where we only make use of unlabeled data, and the expansion property is assumed to hold on U.

Assumption 2. *Assume the distributions are of the following structure: $\text{supp}(S) = \cup_{i=1}^m S_i$, $\text{supp}(T) = \cup_{i=1}^m T_i$, $\text{supp}(U) = \cup_{i=1}^m U_i$, where $U_i \cap U_j = \emptyset$ for $i \neq j$, and $S_i \cup T_i \subset U_i$. Again, assume the ground truth class $g^*(x)$ for $x \in U_i$ is consistent (constant), denoted y_i. We abuse the notation to let S_i, T_i, U_i also denote the conditional distribution of S, T, U on the set S_i, T_i, U_i respectively. We also make the following assumptions, with an additional (c) that says U "covers" S, T.*

(a)(b): Same as Assumption 1(a)(b).

(c) There exists a constant $\kappa \geq 1$ such that the measure S_i, T_i are bounded by κU_i. That is, for any $A \subset \mathcal{X}$,

$$\mathbb{P}_{S_i}(A) \leq \kappa\mathbb{P}_{U_i}(A) \textbf{ and } \mathbb{P}_{T_i}(A) \leq \kappa\mathbb{P}_{U_i}(A).$$

The regularizer now becomes

$$R_{\mathcal{B}}(g) := \mathbb{P}_{x\sim U}[\exists x' \in \mathcal{B}(x), \text{s.t. } g(x) \neq g(x')].$$

On can see that the main difference is that we replaced $\frac{1}{2}(S+T)$ from the previous domain adaptation with a general distribution U. Indeed, we assume expansion on U and can establish bounds on $\epsilon_T(g)$.

Definition 3.1 (Expansion on U)**.** *(1) We say U satisfies (a, c)-multiplicative expansion for some constant $a \in (0, 1)$, $c > 1$, if for any i and any subset $A \subset U$ with $\mathbb{P}_{U_i}[A] \leq a$, we have $\mathbb{P}_{U_i}[\mathcal{N}(A)] \geq \min(c\mathbb{P}_{U_i}[A], 1)$.*

(2) We say U satisfies (q, ξ)-constant expansion for some constant $q, \xi \in (0, 1)$, if for any set $A \subset \mathcal{X}$ with $\mathbb{P}_{U_i}[A] \geq q$ and $\mathbb{P}_{U_i}[A] \leq \frac{1}{2}, \forall i$, we have $\mathbb{P}_U[\mathcal{N}(A)] \geq \min(\xi, \mathbb{P}_U[A]) + \mathbb{P}_U[A]$.

Theorem 3.1 (Bound on Target Error with Multiplicative Expansion, Generalized)**.** *Suppose Assumption 2 holds and U satisfies $(\frac{1}{2}, c)$-multiplicative expansion. Then the classifier obtained by (1) satisfies*

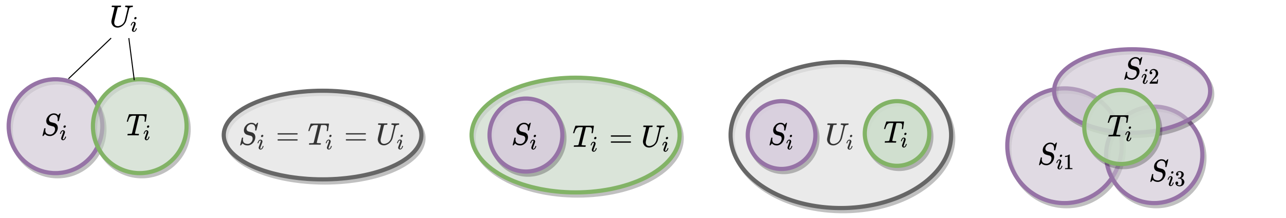
$$\epsilon_T(g) \leq \max\left(\frac{c+1}{c-1}, 3\right)\frac{4\kappa r\mu}{\gamma}.$$

Theorem 3.2 (Bound on Target Error with Constant Expansion, Generalized)**.** *Suppose Assumption 1 holds and U satisfies (q, μ)-constant expansion. Then the classifier obtained by (1) satisfies*

$$\epsilon_T(g) \leq (2\max(q, \mu) + \mu)\frac{4\kappa r\mu}{\gamma}.$$

Choosing special cases of the structure U, we can naturally obtain the following special cases that correspond to the models shown in Figure 2.

1. **Unsupervised domain adaptation** (Figure 2(a)). When $U_i = \frac{1}{2}(S_i + T_i)$, we immediately obtain the



(a) Unsupervised domain adaptation (b) Semi-supervised learning or self-supervised denoising (c) Domain expansion (d) Domain extrapolation (e) Multi-source domain adaptation or domain generalization

Figure 2. Settings of generalized subpopulation shift in Section 3. The figures only draw one subpopulation i for each model.

results in Section 2.2 by plugging in $\kappa = 2$. Therefore, Theorem 2.1 and 2.2 is just a special case of Theorem 3.1 and 3.2.

2. **Semi-supervised learning or self-supervised denoising** (Figure 2(b)). When $S_i = T_i = U_i$, the framework becomes the degenerate version of learning a g from a g_{tc} in a single domain. g_{tc} can be a pseudo-labeler in the semi-supervised learning or some other pre-trained classifier self-supervised denoising. Our results improve upon Wei et al. (2021) under this case as discussed in Remark 1, 2.

3. **Domain expansion** (Figure 2(c)). When $T_i = U_i$, this becomes a problem between semi-supervised learning and domain adaptation, and we call it domain expansion. That is, the source S is a sub-distribution of T where we need to perform well. Frequently, we have a big unlabeled dataset and the labeled data is only a specifc part.

4. **Domain extrapolation** (Figure 2(d)). When $S_i \cup T_i$ does not satisfy expansion by itself, e.g. they are not connected by $\mathcal{B}(\cdot)$, but they are connected through U_i, we can still obtain small error on T. We term this kind of task domain extrapolation, where we have a small source and small target distribution that is not easy to directly correlate, but is possible through a third and bigger unlabeled dataset U where label information can propagate.

5. **Multi-Source domain adaptation or domain generalization** (Figure 2(e)). We have multiple source domains and take U as the union (average measure) of all source domains. Learning is guaranteed if in the input space or some representation space, U can successfully "cover" T, the target distribution in multi-source domain adaptation or the test distribution in domain generalization. Also, as the framework suggests, we do not require all the source domains to be labeled, depending on the specific structure.

The general label propogation framework proposed in this section is widely applicable in many practical scenarios, and would also be an interesting future work. The full proof of the theorems in this section is in Appendix A.

4. Experiments

In this section, we first conduct experiments on a dataset that is constructed to simulate natural subpopulation shift. Then we generalize the aspects of subpopulation shift to classic unsupervised domain adaptation datasets by combining distributional matching methods and consistency-based label propagation method.

4.1. Subpopulation Shift Dataset

We empirically verify that label propagation via consistency regularization works well for subpopulation shift tasks. Towards this goal, we constructed an Unsupervised Domain Adaptation (UDA) task using the challenging ENTITY-30 task from BREEDS tasks (Santurkar et al., 2021), and directly adapt FixMatch (Sohn et al., 2020), an existing consistency regularization method for *semi-supervised learning* to the subpopulation shift tasks. The main idea of FixMatch is to optimize the supervised loss on weak augmentations of source samples, plus consistency regularization, which encourages the prediction of the classifier on strong augmentations of a sample to be the same to the prediction on weak augmentations of the sample[9]. In contrast to semi-supervised learning where the supports of unlabeled data and labeled data are inherently the same, in subpopulation shift problems, the support sets of different domains are disjoint. To enable label propagation, we need a good feature map to enable label propagation on the *feature space*. We thus make use of the feature map learned by a self-supervised learning algorithm SwAV (Caron et al., 2020), which simultaneously *clusters* the data while *enforcing consistency* between cluster assignments produced for different augmentations of the same image. This representation has two merits; first, it encourages subpopulations with similar

[9] Empirically, FixMatch also combines self-training techniques that take the hard label of the prediction on weak augmentations. We also use Distribution Alignment (Berthelot et al., 2019) mentioned in Section 2.5 of the FixMatch paper.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Average
MDD	94.97±0.70	98.78±0.07	100±0	92.77±0.72	75.64±1.53	72.82±0.52	89.16
MDD+FixMatch	95.47±0.95	98.32±0.19	100±0	93.71±0.23	76.64±1.91	74.93±1.15	**89.84**

Table 2. Performance of MDD and MDD+FixMatch on Office-31 dataset.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Average
MDD	54.9±0.7	74.0±0.3	77.7±0.3	60.6±0.4	70.9±0.7	72.1±0.6	60.7±0.8	53.0±1.0	78.0±0.2	71.8±0.4	59.6±0.4	82.9±0.3	68.0
MDD+FixMatch	55.1±0.9	74.7±0.8	78.7±0.5	63.2±1.3	74.1±1.8	75.3±0.1	63.0±0.6	53.0±0.6	80.8±0.4	73.4±0.1	59.4±0.7	84.0±0.5	**69.6**

Table 3. Performance of MDD and MDD+FixMatch on Office-Home dataset.

representations to cluster in the feature space. Second, it enforces the augmented samples to be close in the feature space. We expect that subclasses from the same superclass will be assigned to the same cluster and thus enjoy the expansion property to a certain extent in the feature space. We defer the detailed experimental settings to Appendix C and report the results here.

Method	Source Acc	Target Acc
Train on Source	91.91±0.23	56.73±0.32
DANN (Ganin et al., 2016)	92.81±0.50	61.03±4.63
MDD (Zhang et al., 2019)	92.67±0.54	63.95±0.28
FixMatch (Sohn et al., 2020)	90.87±0.15	72.60±0.51

Table 1. Comparison of performance on ENTITY-30 (Acc refers to accuracy which is measured by percentage).

We compare the performance of the adaptation of FixMatch with popular distributional matching methods, i.e., DANN (Ganin et al., 2016) and MDD (Zhang et al., 2019)[10]. For a fair comparison, all models are finetuned from SwAV representation. As shown in Table 4.1, the adaptation with FixMatch obtains significant improvement upon the baseline method that only trains on the source domain by more than 15% points on the target domain. FixMatch also outperforms distributional matching methods by more than 8%. The results suggest that unlike previous distributional matching-based methods, consistency regularization-based methods are preferable on domain adaptation tasks when encountering subpopulation shift. This is also aligned with our theoretical findings.

4.2. Classic Unsupervised Domain Adaptation Datasets

In this section we conduct experiments on classic unsupervised domain adaptation datasets, i.e., Office-31 (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), where source and target domains mainly differ in style, e.g., artistic images to real-world images. Distributional matching methods seek to learn an invariant representation which removes confounding information such as the style. Since the feature distributions of different domains are encouraged to be matched, the supports of different domains in the feature space are overlapped which enables label propagation. In addition, subpopulation shift from source to target domain may remain even if the styles are unified in the feature space. This inspires us to combine distributional matching methods and label propagation.

As a preliminary attempt, we directly combine MDD (Zhang et al., 2019) and FixMatch (Sohn et al., 2020) to see if there is gain upon MDD. Specifically, we first learn models using MDD on two classic unsupervised domain adaptation datasets, Office-31 and Office-Home. Then we finetune the learned model using FixMatch (with Distribution Alignment extension as described in previous subsection). The results in Table 2, 3 confirm that finetuning with FixMatch can improve the performance of MDD models. The detailed experimental settings can be found in Appendix C.

5. Conclusion

In this work, we introduced a new theoretical framework of learning under subpopulation shift through label propagation, providing new insights on solving domain adaptation tasks. We provided accuracy guarantees on the target domain for a consistency regularization-based algorithm using a fine-grained analysis under the expansion assumption. Our generalized label propagation framework in Section 3 subsumes the previous domain adaptation setting and also provides an interesting direction for future work.

Acknowledgements

JDL acknowledges support of the ARO under MURI Award W911NF-11-1-0303, the Sloan Research Fellowship, NSF CCF 2002272, and an ONR Young Investigator Award. QL is supported by NSF #2030859 and the Computing Research Association for the CIFellows Project. We thank Prof. Yang Yuan for providing computational resources. We also thank Difan Zou for pointing out a mistake in the original proof of Lemma A.1 which is now corrected in the revision.

[10] We use the implementation from Junguang Jiang (2020), which shows that MDD has the best performance among the evaluated methods.

References

Adel, T., Zhao, H., and Wong, A. (2017). Unsupervised domain adaptation with a relaxed covariate shift assumption. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. (2020). Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR.

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

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Becker, C. J., Christoudias, C. M., and Fua, P. (2013). Non-linear domain adaptation with boosting. In *Neural Information Processing Systems (NIPS)*, number CONF.

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

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. (2019). Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments.

Chen, C., Fu, Z., Chen, Z., Jin, S., Cheng, Z., Jin, X., and Hua, X.-S. (2020a). Homm: Higher-order moment matching for unsupervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3422–3429.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chen, Y., Wei, C., Kumar, A., and Ma, T. (2020c). Self-training avoids using spurious features under domain shift. *arXiv preprint arXiv:2006.10032*.

Cortes, C., Mansour, Y., and Mohri, M. (2010). Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450.

Cortes, C., Mohri, M., and Muñoz Medina, A. (2015). Adaptation algorithm and theory based on generalized discrepancy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169–178.

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

Ghifary, M., Kleijn, W. B., Zhang, M., and Balduzzi, D. (2015). Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE.

Gopalan, R., Li, R., and Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*.

He, Z. and Zhang, L. (2019). Multi-adversarial faster-rcnn for unrestricted object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6668–6677.

Heckman, J. J. (1979). Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR.

Hong, W., Wang, Z., Yang, M., and Yuan, J. (2018). Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608.

Javed, K., White, M., and Bengio, Y. (2020). Learning causal models online. *arXiv preprint arXiv:2006.07461*.

Jhuo, I.-H., Liu, D., Lee, D., and Chang, S.-F. (2012). Robust visual domain adaptation with low-rank reconstruction. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2168–2175. IEEE.

Junguang Jiang, Bo Fu, M. L. (2020). Transfer-learning-library. `https://github.com/thuml/Transfer-Learning-Library`.

Kanamori, T., Suzuki, T., and Sugiyama, M. (2011). f-divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE transactions on information theory*, 58(2):708–720.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Priol, R. L., and Courville, A. (2020). Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*.

Kumar, A., Ma, T., and Liang, P. (2020). Understanding self-training for gradual domain adaptation. *arXiv preprint arXiv:2002.11361*.

Lee, C.-Y., Batra, T., Baig, M. H., and Ulbricht, D. (2019). Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295.

Li, B., Wang, Y., Che, T., Zhang, S., Zhao, S., Xu, P., Zhou, W., Bengio, Y., and Keutzer, K. (2020). Rethinking distributional matching based domain adaptation. *arXiv preprint arXiv:2006.13352*.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. (2018). Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lin, Y., Lee, Y., and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1):191–202.

Liu, F., Lu, J., Han, B., Niu, G., Zhang, G., and Sugiyama, M. (2019). Butterfly: A panacea for all difficulties in wildly unsupervised domain adaptation. *arXiv preprint arXiv:1905.07720*.

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

Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2017a). Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017b). Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR.

Menon, A. and Ong, C. S. (2016). Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pages 304–313. PMLR.

Mitrovic, J., McWilliams, B., Walker, J., Buesing, L., and Blundell, C. (2020). Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. (2020). Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*.

Pei, Z., Cao, Z., Long, M., and Wang, J. (2018). Multi-adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Qiao, S., Shen, W., Zhang, Z., Wang, B., and Yuille, A. (2018). Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset shift in machine learning*.

Roy, S., Siarohin, A., Sangineto, E., Bulo, S. R., Sebe, N., and Ricci, E. (2019). Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9471–9480.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Santurkar, S., Tsipras, D., and Madry, A. (2021). {BREEDS}: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Shu, R., Bui, H. H., Narui, H., and Ermon, S. (2018). A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*.

Shu, Y., Cao, Z., Long, M., and Wang, J. (2019). Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4951–4958.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. (2008). Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746.

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

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. (2016). Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*.

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

Wei, C. and Ma, T. (2019). Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv preprint arXiv:1910.04284*.

Wei, C., Shen, K., Chen, Y., and Ma, T. (2021). Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*.

Wu, Y., Winston, E., Kaushik, D., and Lipton, Z. (2019). Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pages 6872–6881. PMLR.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. (2020). Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33.

Xie, R., Yu, F., Wang, J., Wang, Y., and Zhang, L. (2019). Multi-level domain adaptive learning for cross-domain detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-I., and Jegelka, S. (2021). How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*.

Xu, R., Chen, Z., Zuo, W., Yan, J., and Lin, L. (2018). Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973.

Ye, H., Xie, C., Cai, T., Li, R., Li, Z., and Wang, L. (2021). Towards a theoretical framework of out-of-distribution generalization.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2013). Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827.

Zhang, L. (2019). Transfer adaptation learning: A decade survey. *arXiv preprint arXiv:1903.04687*.

Zhang, Y., Liu, T., Long, M., and Jordan, M. (2019). Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. (2019a). On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*.

Zhao, H., Dan, C., Aragam, B., Jaakkola, T. S., Gordon, G. J., and Ravikumar, P. (2020a). Fundamental limits and tradeoffs in invariant representation learning. *arXiv preprint arXiv:2012.10713*.

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

Zhao, S., Li, B., Yue, X., Gu, Y., Xu, P., Hu, R., Chai, H., and Keutzer, K. (2019b). Multi-source domain adaptation for semantic segmentation. *arXiv preprint arXiv:1910.12181*.

Zhao, S., Yue, X., Zhang, S., Li, B., Zhao, H., Wu, B., Krishna, R., Gonzalez, J. E., Sangiovanni-Vincentelli, A. L., Seshia, S. A., et al. (2020b). A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhu, X., Pang, J., Yang, C., Shi, J., and Lin, D. (2019). Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 687–696.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

A. Proof of Theorem 2.1, 2.2, 3.1, and 3.2

Note that in Section 3, by taking $U = \frac{1}{2}(S+T)$, in Assumption 2(c) we have $\kappa = 2$. By plugging in κ, Theorem 2.1 and 2.2 immediately becomes the corollary of Theorem 3.1 and 3.2. Therefore, we only provide a full proof for Theorem 3.1 and 3.2 here.

First, similar to Section 2.4, we give a proof sketch for Theorem 3.1 and 3.2, which includes the corresponding definitions and lemmas for this generalized setting.

A.1. Proof Sketch for Theorem 3.1 and 3.2

To prove the theorems, we first introduce some concepts and notations.

A point $x \in \mathcal{X}$ is called *robust* w.r.t. $\mathcal{B}$ and g if for any x' in $\mathcal{B}(x)$, $g(x) = g(x')$. Denote

$$RS(g) := \{x | g(x) = g(x'), \forall x' \in \mathcal{B}(x)\},$$

which is called the *robust set* of g. Let

$$A_{ik} := RS(g) \cap U_i \cap \{x | g(x) = k\}$$

for $i \in \{1, \cdots, m\}$, $k \in \{1, \cdots, K\}$, and they form a partition of the set $RS(g)$. Denote

$$y_i^{\text{Maj}} := \underset{k \in \{1,\cdots,K\}}{\text{argmax}} \; \mathbb{P}_U[A_{ik}],$$

which is the majority class label of g in the robust set on U_i. We also call

$$M_i := \bigcup_{k \in \{1,\cdots,K\} \backslash \{y_i^{\text{Maj}}\}} A_{ik}$$

and $M := \bigcup_{i=1}^m M_i$ the *minority robust set* of g. In addition, let

$$\widetilde{M}_i := U_i \cap \{x | g(x) \neq y_i^{\text{Maj}}\}$$

and $\widetilde{M} := \bigcup_{i=1}^m \widetilde{M}_i$ be the *minority set* of g, which is superset to the minority robust set.

The expansion property can be used to control the total population of the minority set.

Lemma A.1 (Upper Bound of Minority Set)**.** *For the classifier g obtained by (1), $\mathbb{P}_U[\widetilde{M}]$ can be bounded as follows:*

(a) Under $(\frac{1}{2}, c)$-multiplicative expansion, we have $\mathbb{P}_U[\widetilde{M}] \leq \max\left(\frac{c+1}{c-1}, 3\right)\mu$.

(b) Under (q, μ)-constant expansion, we have $\mathbb{P}_U[\widetilde{M}] \leq 2\max(q, \mu) + \mu$.

Based on the bound on the minority set, our next lemma says that on most subpopulation components, the inconsistency between g and g_{tc} is no greater than the error of g_{tc} plus a margin $\frac{\gamma}{2}$. Specifically, define

$$\begin{aligned} I = &\{i \in \{1, \cdots, m\} | \\ &\mathbb{P}_{x \sim S_i}[g(x) \neq g_{tc}(x)] > \mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i] + \frac{\gamma}{2}\} \end{aligned}$$

and we have the following result

Lemma A.2 (Upper Bound on the Inconsistent Components I)**.** *Suppose $\mathbb{P}_U[\widetilde{M}] \leq C$, then*

$$\mathbb{P}_S[\cup_{i \in I} S_i] \leq \frac{2\kappa C}{\gamma}.$$

Based on the above results, we are ready to bound the target error $\epsilon_T(g)$.

Lemma A.3 (Bounding the Target Error)**.** *Suppose $\mathbb{P}_U[\widetilde{M}] \leq C$. Let*

$$\epsilon_T^i(g) = \mathbb{P}_T[T_i]\mathbb{P}_{x \sim T}[g(x) \neq y_i]$$

for i in $\{1, \cdots, m\}$, so that $\epsilon_T(g) = \sum_{i=1}^m \epsilon_T^i(g)$. Then we can separately bound

(a) $\sum_{i \in I} \epsilon_T^i(g) \leq \frac{2\kappa r C}{\gamma}$

(b) $\sum_{i \in \{1,\cdots,m\} \backslash I} \epsilon_T^i(g) \leq \frac{2\kappa r C}{\gamma}$

so that the combination gives

$$\epsilon_T(g) \leq \frac{4\kappa r C}{\gamma}.$$

Specically, Lemma A.3(a) is obtained by directly using Lemma A.2, and Lemma A.3(b) is proved by a fine-grained analysis on the minority set.

Finally, we can plug in C from Lemma A.1 and the desired results in Theorem 3.1 and 3.2 are obtained.

To make the proof complete, we provide a detailed proof of Lemma A.1, A.2, A.3 in the following subsections.

A.2. Proof of Lemma A.1.

Proof. We first prove the (q, μ)-expansion case (a). The probability function $\mathbb{P}$ are all w.r.t. the distribution U in this lemma, so we omit this subscript. We also use $\mathbb{P}_i$ for $\mathbb{P}_{U_i}$ in this lemma.

The robust minority set is $M_i = \cup_{k \in \{1,\cdots,K\} \backslash \{y_i^{\text{Maj}}\}} A_{ik}$. In order to do expansion, we partition M_i into two halves:

Lemma A.4 (Partition of M_i)**.** *For each $i \in \{1, \cdots, m\}$, there exists a partition of the set $\{1, \cdots, K\} \backslash \{y_i^{Maj}\}$ into J_{i1} and J_{i2} such that the corresponding partition $M_i = M_i^1 \cup M_i^2$ ($M_i^1 = \cup_{k \in J_{i1}} A_{ik}$, $M_i^2 = \cup_{k \in J_{i2}} A_{ik}$) satisfies $\mathbb{P}_i[M_i^1] \leq \frac{1}{2}$ and $\mathbb{P}_i[M_i^2] \leq \frac{1}{2}$.*

Proof. Starting from $J_{i1} = J_{i2} = \emptyset$, and each time we add an element $k_0 \in \{1, \cdots, K\} \backslash \{y_i^{\text{Maj}}\}$ into J_{i1} or J_{i2} while keeping the properties $\mathbb{P}_i[M_i^1] \leq \frac{1}{2}$ and $\mathbb{P}_i[M_i^2] \leq \frac{1}{2}$ hold. We prove that for any $k_0 \in \{1, \cdots, K\} \backslash (\{y_i^{\text{Maj}}\} \cup J_{i1} \cup J_{i2})$, either $\mathbb{P}_i[\cup_{k \in J_{i1} \cup \{k_0\}} A_{ik}] \leq \frac{1}{2}$ or $\mathbb{P}_i[\cup_{k \in J_{i2} \cup \{k_0\}} A_{ik}] \leq \frac{1}{2}$ holds, so we can repeat the process until J_{i1} and J_{i2} is a partition of $\{1, \cdots, K\} \backslash \{y_i^{\text{Maj}}\}$. In fact, since

$$\begin{aligned} &\mathbb{P}_i[\cup_{k \in J_{i1} \cup \{k_0\}} A_{ik}] + \mathbb{P}_i[\cup_{k \in J_{i2} \cup \{k_0\}} A_{ik}] \\ \leq &\mathbb{P}_i[\cup_{k \in J_{i1} \cup \{k_0\}} A_{ik}] + \mathbb{P}_i[\cup_{k \in J_{i2} \cup \{y_i^{\text{Maj}}\}} A_{ik}] \\ &\text{(By the definition of } y_i^{\text{Maj}}) \\ \leq &\mathbb{P}_i[\cup_{k \in \{1, \cdots, K\}} A_{ik}] \\ \leq &1, \end{aligned}$$

we know that either $\mathbb{P}_i[\cup_{k \in J_{i1} \cup \{k_0\}} A_{ik}]$ or $\mathbb{P}_i[\cup_{k \in J_{i2} \cup \{k_0\}} A_{ik}]$ is no more than $\frac{1}{2}$, and Lemma A.4 is proved. □

Let $M^1 = \cup_{i=1}^m M_i^1$, $M^2 = \cup_{i=1}^m M_i^2$, so that M^1 and M^2 form a partition of M. Based on Lemma A.4, we know that either $\mathbb{P}[M^1] < q$, or M^1 satisfies the requirement for (q, μ)-constant expansion. Hence,

$$\mathbb{P}[\mathcal{N}(M^1)] \geq \mathbb{P}[M^1] + \min\left(\mu, \mathbb{P}[M^1]\right) \text{ or } \mathbb{P}[M^1] < q \tag{3}$$

On the other hand, we claim that $\mathcal{N}(M^1) \backslash M^1$ contains only non-robust points. Otherwise, suppose there exists a robust point $x \in \mathcal{N}(M^1) \backslash M^1$, say $x \in \mathcal{N}(A_{ik})$ for some $i \in \{1, \cdots, m\}$ and $k \in J_{i1}$. By the definition of neighborhood, there exists $x' \in A_{ik}$ such that there exists $x'' \in \mathcal{B}(x) \cap \mathcal{B}(x')$. Therefore, by the definition of robustness, $g(x) = g(x'') = g(x') = k$. Also by the definition of neighborhood, we know that $x \in U_i$, so it must be that $x \in A_{ik}$ since x is robust. This contradicts with $x \notin M^1$! Therefore, $\mathcal{N}(M^1) \backslash M^1$ is a subset of the set of all non-robust points. Since the total measure of non-robust points is $R_{\mathcal{B}}(g)$ by definition, we know that

$$\mathbb{P}[\mathcal{N}(M^1)] - \mathbb{P}[M^1] \leq \mathbb{P}[\mathcal{N}(M^1) \backslash M^1] \leq R_{\mathcal{B}}(G) < \mu. \tag{4}$$

Combining (3) and (4), we know that under $\mathbb{P}[M^1] \geq q$, it must hold that $\mathbb{P}[M^1] < \mu$, or else (3) and (4) would be a contradiction. In all, this means that $\mathbb{P}[M^1] \leq \max(q, \mu)$ in any case.

Similarly, we know $\mathbb{P}[M^2] \leq \max(q, \mu)$ also hold. Therefore, $\mathbb{P}[M] \leq 2\max(q, \mu)$.

Since $\widetilde{M} \backslash M$ only consists of non-robust points, we know that

$$\mathbb{P}[\widetilde{M}] \leq \mathbb{P}[M] + R_B(g) \leq 2\max(q, \mu) + \mu,$$

which is the desired result (a).

For the $(\frac{1}{2}, c)$-multiplicative expansion case (b), it is easy to verify that $(\frac{1}{2}, c)$-multiplicative expansion must imply $(\frac{\mu}{c-1}, \mu)$-constant expansion (See Lemma B.6 in Wei et al. (2021)). Therefore, the result is obtained by plugging in $q = \frac{\mu}{c-1}$. □

A.3. Proof of Lemma A.2

Proof. We first prove the following lemma.

Lemma A.5. *For any $i \in \{1, \cdots, m\}$, we have*

$$\mathbb{P}_{x \sim S_i}[g(x) \neq g_{tc}(x)] + \mathbb{P}_{S_i}[\widetilde{M}_i] \geq \mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i]. \tag{5}$$

Proof. Based on the margin assumption (Assumption 2(a)), we know that $\mathbb{P}_{x \sim S_i}[g_{tc}(x) = y_i] \geq \mathbb{P}_{x \sim S_i}[g_{tc}(x) = y_i^{\text{Maj}}]$. Therefore, $\mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i] \leq \mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i^{\text{Maj}}]$. Along with triangle inequality we know that

$$\begin{aligned} &\mathbb{P}_{x \sim S_i}[g(x) \neq g_{tc}(x)] + \mathbb{P}_{S_i}[\widetilde{M}_i] \\ = &\mathbb{P}_{x \sim S_i}[g(x) \neq g_{tc}(x)] + \mathbb{P}_{x \sim S_i}[g(x) \neq y_i^{\text{Maj}}] \\ \geq &\mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i^{\text{Maj}}] \\ \geq &\mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i], \end{aligned}$$

which proves the result. □

Based on Lemma A.5, we can write:

$$\begin{aligned} &L_{01}^S(g, g_{tc}) \\ = &\sum_{i \in I} \mathbb{P}_S[S_i] \mathbb{P}_{x \sim S_i}[g(x) \neq g_{tc}(x)] \\ &+ \sum_{i \in \{1, \cdots, m\} \backslash I} \mathbb{P}_S[S_i] \mathbb{P}_{x \sim S_i}[g(x) \neq g_{tc}(x)] \\ \geq &\sum_{i \in I} \mathbb{P}_S[S_i] \left(\mathbb{P}_{x \sim S_i}[g(x) \neq y_i] + \frac{\gamma}{2}\right) \\ &+ \sum_{i \in \{1, \cdots, m\} \backslash I} \mathbb{P}_S[S_i] \left[\mathbb{P}_{x \sim S_i}[g_{tc}(x) \neq y_i] - \mathbb{P}_{S_i}[\widetilde{M}_i]\right] \\ &\text{(by definition of } I \text{ for the first term,} \\ &\quad \text{by Lemma A.5 for the second term)} \\ = &L_{01}^S(g^*, g_{tc}) + \frac{\gamma}{2} \sum_{i \in I} \mathbb{P}_S[S_i] \\ &- \sum_{i \in \{1, \cdots, m\} \backslash I} \mathbb{P}_S[\widetilde{M}_i]. \end{aligned} \tag{6}$$

Since by definition of our algorithm, $L_{01}^S(g, g_{tc}) \leq$

$L_{01}^{S}(g^*, g_{tc})$, we finally know that

$$\begin{aligned}\sum_{i\in I}\mathbb{P}_S[S_i] &\le \frac{2}{\gamma}\sum_{i\in\{1,\cdots,m\}\backslash I}\mathbb{P}_S[\widetilde{M}_i]\\ &\le \frac{2}{\gamma}\mathbb{P}_S[\widetilde{M}]\\ &\le \frac{2\kappa}{\gamma}\mathbb{P}_U[\widetilde{M}] \quad \text{(by Assumption 2(c))}\\ &\le \frac{2\kappa C}{\gamma},\end{aligned}$$

which is the desired result. □

A.4. Proof of Lemma A.3.

Proof. (a) This is a direct result from Lemma A.2 since

$$\begin{aligned}\sum_{i\in I}\epsilon_T^i(g) &\le \sum_{i\in I}\mathbb{P}_T[T_i]\\ &\le \sum_{i\in I} r\mathbb{P}_S[S_i]\\ &\le \frac{2\kappa rC}{\gamma}. \end{aligned} \tag{7}$$

(b) For $i\in\{1,\cdots,m\}\backslash I$, we proceed by considering the following two cases: $y_i = y_i^{\text{Maj}}$ or $y_i \ne y_i^{\text{Maj}}$.

If $y_i = y_i^{\text{Maj}}$, we have

$$\epsilon_T^i(g) = \mathbb{P}_T[\widetilde{M}_i] \le \kappa\mathbb{P}_U[\widetilde{M}_i].$$

If $y_i \ne y_i^{\text{Maj}}$, we have

$$\begin{aligned}\frac{\mathbb{P}_S[\widetilde{M}_i]}{\mathbb{P}_S[S_i]} &= \mathbb{P}_{S_i}[\widetilde{M}_i]\\ &= \mathbb{P}_{x\sim S_i}[g(x)\ne y_i^{\text{Maj}}]\\ &\ge \mathbb{P}_{x\sim S_i}[g_{tc}(x)\ne y_i^{\text{Maj}}] - \mathbb{P}_{x\sim S_i}[g_{tc}(x)\ne g(x)]\\ &\text{(triangle inequality)}\\ &= 1-\mathbb{P}_{x\sim S_i}[g_{tc}(x) = y_i^{\text{Maj}}]\\ &\quad -\mathbb{P}_{x\sim S_i}[g_{tc}(x)\ne g(x)]\\ &\ge 1-(\mathbb{P}_{x\sim S_i}[g_{tc}(x)=y_i]-\gamma)\\ &\quad -(\mathbb{P}_{x\sim S_i}[g_{tc}(x)\ne y_i]+\frac{\gamma}{2})\\ &\text{(Assumption 2(a) and Definition of } I)\\ &= \frac{\gamma}{2}.\end{aligned}$$

Then we have

$$\begin{aligned}\epsilon_T^i(g) &\le \mathbb{P}_T[T_i]\\ &\le r\mathbb{P}_S[S_i]\\ &\le \frac{2r}{\gamma}\mathbb{P}_S[\widetilde{M}_i]\\ &\le \frac{2\kappa r}{\gamma}\mathbb{P}_U[\widetilde{M}_i]\end{aligned}$$

Summarizing the two cases above, since $\frac{2\kappa r}{\gamma}\ge 2$ must hold, we always have

$$\epsilon_T^i(g) \le \frac{2\kappa r}{\gamma}\mathbb{P}_U[\widetilde{M}_i],$$

and as a result,

$$\begin{aligned}\sum_{i\in\{1,\cdots,m\}\backslash I}\epsilon_T^i(g) &\le \sum_{i\in\{1,\cdots,m\}\backslash I}\frac{2\kappa r}{\gamma}\mathbb{P}_U[\widetilde{M}_i]\\ &\le \frac{2\kappa r}{\gamma}\mathbb{P}_U[\widetilde{M}]\\ &= \frac{2\kappa rC}{\gamma}.\end{aligned}$$

□

B. Proof of Results in Section 2.3

As a side note, we first state the definition of the all-layer margin from Wei and Ma (2019). For the neural network $f(x) = W_p\phi(\cdots\phi(W_1x)\cdots)$, we write f as $f(x) = f_{2p-1}\circ\cdots\circ f_1(x)$, where the f_i's alternate between matrix multiplications and applications of the activation function ϕ. Let $\delta_1,\cdots,\delta_{2p-1}$ denote perturbations intended to be applied at each layer $i = 1,\cdots,2p-1$, and the perturbed network output $f(x,\delta_1,\cdots,\delta_{2p-1})$ is recursively defined as

$$\begin{aligned}h_1(x,\delta) &= f_1(x)+\delta_1\|x\|_2,\\ h_i(x,\delta) &= f_i(h_{i-1}(x,\delta))+\delta_i\|h_{i-1}(x,\delta)\|_2,\\ f(x,\delta) &= h_{2p-1}(x,\delta).\end{aligned}$$

And the all-layer margin is defined as the minimum norm of δ required to make the classifier misclassify the input, i.e.

$$m(f,x,y) := \min_{\delta_1,\cdots,\delta_{2p-1}}\sqrt{\sum_{i=1}^{2p-1}\|\delta_i\|_2^2}$$
$$\text{subject to } \operatorname*{argmax}_{y'} f(x,\delta_1,\cdots,\delta_{2p-1})_{y'}\ne y.$$

The related results about all-layer margin (Proposition 2.1 and 2.2), though, come directly from Wei et al. (2021).

B.1. Proof of Theorem 2.3

We first state a stronger version of Lemma 2.2 and 2.3 in the following lemma (a) and (b).

Lemma B.1 (Stronger Version of Lemma 2.2 and 2.3)**.** *We assume that $L_{01}^S(g, g_{tc}) \le L_{01}^S(g^*, g_{tc}) + 2\Delta$. (In the previous proofs of Section 2.2, $\Delta = 0$.) Similarly, suppose $\mathbb{P}_{\frac{1}{2}(S+T)}[\widetilde{M}] \le C$, then we have the following results:*

(a) The "inconsistency set" I is upper-bounded by

$$\mathbb{P}_S[\cup_{i\in I} S_i] \le \frac{4(C+\Delta)}{\gamma}.$$

(b) The final target error is upper-bounded by

$$\epsilon_T(g) \le \frac{8r(C+\Delta)}{\gamma}.$$

So we only need to find C and Δ. C can be found by Lemma 2.1 by using a suitable $\hat{\mu}$ where $R_{\mathcal{B}}(g) \le \hat{\mu}$. These results are given by the following lemma.

Lemma B.2 (Finite Sample Bound)**.** *We have*

$$L_{01}^S(g, g_{tc}) \le L_{01}^S(g^*, g_{tc}) + 2\Delta$$

and

$$R_{\mathcal{B}}(g) \le \hat{\mu}$$

for

$$\Delta = \widetilde{O}\left(\left(\mathbb{P}_{x\sim\hat{S}}[m(f^*, x, g_{tc}(x)) \le t] - L_{01}^{\hat{S}}(g^*, g_{tc})\right)\right.$$
$$\left. + \frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right),$$
$$\hat{\mu} = \mu + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right).$$

And by plugging in the results from Lemma B.1 and B.2, along with the constant C in Lemma 2.1, we immediately get the result in Theorem 2.3. The proof of Lemma 2.1 can be founded in the proof of Lemma A.1 in Appendix A, so we only need to prove Lemma B.1 and B.2 below.

Proof of Lemma B.1.

Proof. (a). We only need to modify the proof of Lemma A.2 (Appendix A.3), equation (6), in the following way (where $\kappa = 2$):

$$\begin{aligned}
& L_{01}^S(g, g_{tc}) \\
&= \sum_{i\in I} \mathbb{P}_S[S_i]\mathbb{P}_{x\sim S_i}[g(x) \neq g_{tc}(x)] \\
&\quad + \sum_{i\in\{1,\cdots,m\}\backslash I} \mathbb{P}_S[S_i]\mathbb{P}_{x\sim S_i}[g(x) \neq g_{tc}(x)] \\
&\ge \sum_{i\in I} \mathbb{P}_S[S_i]\left(\mathbb{P}_{x\sim S_i}[g(x) \neq y_i] + \frac{\gamma}{2}\right) \\
&\quad + \sum_{i\in\{1,\cdots,m\}\backslash I} \mathbb{P}_S[S_i]\left[\mathbb{P}_{x\sim S_i}[g_{tc}(x) \neq y_i] - \mathbb{P}_{S_i}[\widetilde{M}_i]\right] \\
&\quad \text{(by definition of } I \text{ for the first term,} \\
&\quad\quad \text{by Lemma A.5 for the second term)} \\
&= L_{01}(g^*, g_{tc}) + \frac{\gamma}{2}\sum_{i\in I}\mathbb{P}_S[S_i] \\
&\quad - \sum_{i\in\{1,\cdots,m\}\backslash I} \mathbb{P}_S[\widetilde{M}_i] \\
&\ge L_{01}^S(g, g_{tc}) - 2\Delta + \frac{\gamma}{2}\sum_{i\in I}\mathbb{P}_S[S_i] - 2C,
\end{aligned}$$

and we immediately obtain

$$\sum_{i\in I}\mathbb{P}_S[S_i] \le \frac{4(C+\Delta)}{\gamma}.$$

(b). Similarly, we only need to modify the proof of Lemma A.3 (a) (Appendix A.4), equation (7) based on the previous Lemma B.1 in the following way (where $\kappa = 2$):

$$\begin{aligned}
\sum_{i\in I}\epsilon_T^i(g) &\le \sum_{i\in I}\mathbb{P}_T[T_i] \\
&\le \sum_{i\in I} r\mathbb{P}_S[S_i] \\
&\le \frac{4r(C+\Delta)}{\gamma}.
\end{aligned}$$

And since Lemma A.3 (b)

$$\sum_{i\in\{1,\cdots,K\}\backslash I}\epsilon_T^i(g) \le \frac{4rC}{\gamma}$$

holds without change, together we easily have

$$\epsilon_T(g) \le \frac{8r(C+\Delta)}{\gamma}.$$

□

Proof of Lemma B.2.

Proof. By Proposition 2.1, we know that:

$$
\begin{aligned}
& L_{01}^S(g, g_{tc}) - L_{01}^S(g^*, g_{tc}) \\
\le\ & \mathbb{P}_{x\sim\hat{S}}[m(f,x,g_{tc}(x)) \le t] - L_{01}^S(g^*, g_{tc}) \\
& + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta)+p\log n}{n}}\right) \\
\le\ & \mathbb{P}_{x\sim\hat{S}}[m(f,x,g_{tc}(x)) \le t] - L_{01}^S(g^*, g_{tc}) \\
& + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta)+p\log n}{n}}\right) \\
& \text{(by algorithm (2))} \\
\le\ & \mathbb{P}_{x\sim\hat{S}}[m(f,x,g_{tc}(x)) \le t] - L_{01}^{\hat{S}}(g^*, g_{tc}) \\
& + O\left(\sqrt{\frac{\log(1/\delta)}{n}}\right) \\
& + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta)+p\log n}{n}}\right) \\
& \text{(by standard concentration bound)} \\
=\ & 2\Delta.
\end{aligned}
$$

By Proposition 2.2, we have

$$
\begin{aligned}
R_{\mathcal{B}}(g) \le\ & \mathbb{P}_{x\sim\frac{1}{2}(\hat{S}+\hat{T})}[m_{\mathcal{B}}(f,x) \le t] \\
& + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta)+p\log n}{n}}\right) \\
\le\ & \hat{\mu}. \quad \text{(by algorithm (2))}
\end{aligned}
$$

And the lemma is proved.

□

C. Detailed Experimental Settings

In this section, we describe the detailed setting of our experiments.

C.1. Dataset

ENTITY-30 (Santurkar et al., 2021). We use the ENTITY-30 dataset from BREEDS (Santurkar et al., 2021) to simulate natural subpopulation shift. ENTITY-30 is constructed by data from ImageNet. It consists of 30 superclasses of entities, e.g., insect, carnivore, and passerine, which are the labels of classification task. Each superclass has eight subclasses; for example, the superclass insect has fly, leafhopper, etc., as its subclasses. The dataset is constructed by splitting each superclass's subclasses into two random and disjoint sets and assigning one of them to the source and the other to the target domain. Each subclass has the same probability of being chosen into source and target and has the same number of samples. This ensures the source and target datasets are approximately balanced w.r.t. superclass. To simulate subpopulation shift scenarios, we construct an unsupervised domain adaptation task. We provide labels of superclasses on the source domain and only unlabeled data on the target domain. The goal is to achieve good population accuracy in the target domain. In the randomly generated ENTITY-30 dataset we used for experiments, there are 157487 labeled samples in the source domain and 150341 unlabeled data in the target domain.

Office-31 (Saenko et al., 2010). Office-31 is a standard domain adaptation dataset of three diverse domains, Amazon from Amazon website, Webcam by web camera and DSLR by digital SLR camera with 4,652 images in 31 unbalanced classes.

Office-Home (Venkateswara et al., 2017). Office-Home is a more complex dataset containing 15,500 images from four visually very different domains: Artistic images, Clip Art, Product images, and Real-world images.

C.2. Adaptation of FixMatch for Subpopulation Shift

We adapt the state-of-the-art semi-supervised learning method FixMatch (Sohn et al., 2020) to the subpopulation shift. Unlike semi-supervised learning, where the support sets of unlabeled data and labeled data are inherently the same, the support sets of different domains may disjoint a lot in subpopulation shift problems. To enable label propagation, we need a good feature map to enable label propagation on the *feature space*. Such a feature map should be obtained without the need for labels on the target domain. Under these constraints, we hypothesize that the feature map learned by modern self-supervised learning algorithms helps. Concretely, we use the feature map learned by SwAV (Caron et al., 2020) which simultaneously *clusters* the data while *enforcing consistency* between cluster assignments produced for different augmentations of the same image. This representation has two merits; first, it encourages subpopulations with similar representations to cluster in the feature space; second, it enforces the augmented samples to be close in the feature space. We expect that subclasses from the same superclass will be assigned to the same cluster and thus overlap in the feature space.

Our adaptation of FixMatch has the following pipeline:

- Step 1: We first finetune a ResNet50 model with pretrained SwAV representation[11] on the source domain;

[11] Since pretraining from scratch requires much computation resource, we simply take the officially released checkpoint from `https://github.com/facebookresearch/swav`. Note this representation is learned on *unlabeled* ImageNet

- Step 2: Then we use this model as the base classifier and further finetune it with the objective function of FixMatch, i.e., supervised loss on weak augmentations of source samples plus consistency regularization, which encourages the prediction of the classifier on strong augmentations of a sample to be same to the prediction on weak augmentations of the sample[12].

C.3. Hyperparameter Settings and Training Details

C.3.1. Subpopulation Shift Datasets

We evaluate four methods, i.e., Training only on the Source Domain (we use TSD for acronym), FixMatch, DANN (Ganin et al., 2016), MDD (Zhang et al., 2019). For Step 1 of FixMatch mentioned in Section C.2, we simply take the model training only on the source domain.

We hardly tune the hyperparameter from their default values from the released repos: `https://github.com/facebookresearch/swav` for the hyperparameters for finetuning from SwAV, `https://github.com/kekmodel/FixMatch-pytorch` for Fixmatch training, `https://github.com/thuml/Transfer-Learning-Library` for DANN and MDD.

We train all models for 30 epochs using SGD (FixMatch is finetuned from TSD for 30 epochs). Each configuration is evaluated with 3 different random seeds, and the mean and standard deviation are reported. We follow the configuration of `eval_semisup.py` in `https://github.com/facebookresearch/swav/blob/master/eval_semisup.py` for TSD and FixMatch with some scaling of learning rate together with batch size and further take the hyperparameters of FixMatch from `https://github.com/kekmodel/FixMatch-pytorch`. For TSD: we use an initial learning rate of 0.4 for the last linear layer and 0.02 for other layers; we decay the learning rate by 10 at epoch 15 and 22; we train on 4 NVIDIA RTX 2080 Ti GPUs with 64 samples on each GPU. For FixMatch, we use an initial learning rate of 0.1 for the last linear layer and 0.005 for other layers; we use a cosine learning rate decay; we train on 4 NVIDIA RTX 2080 Ti GPUs with 16 labeled data from the source domain and 3×16 (set the hyperparameter μ in FixMatch to 3, whose default value is 7, due to the limitation of GPU memory of RTX 2080 Ti) unlabeled data from target domain; we use Distribution Alignment (Berthelot et al., 2019) extension from the FixMatch paper (Sohn et al., 2020) (Section 2.5)[13]; we select the parameter λ_u of FixMatch (the coefficient of the consistency loss) from $\{1, 10, 100\}$ and use 10 for our experiments; the threshold hyperparameter τ is set to the default value 0.95. Since DANN and MDD are already algorithms for unsupervised domain adaptation, we directly use the default hyperparameters from `https://github.com/thuml/Transfer-Learning-Library` for DANN and MDD. We train each DANN and MDD model on a single NVIDIA RTX 2080 Ti GPU as the original code does not support multi-GPU training, and we just keep it.

In all experiments, we use Pytorch 1.7.1 with CUDA version 10.1. For all optimizers, we use SGD with the Nesterov momentum 0.9 and weight decay 5e-4. For TSD and FixMatch, we further use NVIDIA's apex library to enable mixed-precision training (with optimization level O1).

C.3.2. Classic Unsupervised Domain Adaptation Datasets

We train MDD models on Office-31 and Office-Home datasets following the configuration in `https://github.com/thuml/Transfer-Learning-Library`. Then we finetune the learned model using FixMatch (with Distribution Alignment extension) for 20 epochs. We do not tune any hyperparameters but directly use the same learning rate scale and learning rate scheduler as MDD, i.e., the batch size is 64, the initial learning rate is 0.008 for the last layers while is 0.0008 for the backbone feature extractor (ResNet50)[14], the learning rate at step i follows the schedule $\text{lr} = \text{initial lr} \times (1 + 0.0002i)^{-0.75}$. The hyperparameters of FixMatch is set as $\mu = 3$ and $\lambda_u = 1$.

In all experiments, we use Pytorch 1.7.1 with CUDA version 10.1. For all optimizers, we use SGD with the Nesterov momentum 0.9 and weight decay 5e-4. For FixMatch finetuning, we further use NVIDIA's apex library to enable mixed-precision training (with optimization level O1).

training set, a superset of ENTITY-30 training set. There is no leakage of label information on the target domain.

[12] Empirically, FixMatch also combines self-training techniques that take the hard label of the prediction on weak augmentations and soft label for strong augmentations. We also use Distribution Alignment extension mentioned in Section 2.5 in FixMatch paper (Sohn et al., 2020).

[13] As mentioned in the FixMatch paper, this extension encourages the model predictions to have the same class distribution as the labeled set, and their results show that this extension is effective when the number of labeled data is small. We find this extension is also helpful in our subpopulation shift setting (improves the accuracy from 68.5% to 72.6%). We hypothesize that this is because distribution alignment helps to learn a suitable representation on which the separation and expansion of subpopulation are well-satisfied so that label information can propagate.

[14] The default batch size and initial learning rate of MDD are 32 and 0.004, we simultaneously scale them by a factor of 2 for using parallel computation.

D. Other Related Works

There are many works designing algorithms based on the idea of distributional matching (Adel et al., 2017; Becker et al., 2013; Pei et al., 2018; Jhuo et al., 2012; Hoffman et al., 2018; Zhao et al., 2019b; Long et al., 2017b). We refer the readers to Zhang (2019); Zhuang et al. (2020); Zhao et al. (2020b) for comprehensive surveys.

Domain generalization is a fundamental extension of domain adaptation; the distinction to domain adaptation is made precisely in, e.g., Gulrajani and Lopez-Paz (2020). Most domain generalization methods aim to incorporate the invariances across all training datasets instead of only being invariant to a specific test domain (Ghifary et al., 2015). Different types of invariances are leveraged through algorithms like invariant risk minimization or its variants (Arjovsky et al., 2019; Ahuja et al., 2020; Parascandolo et al., 2020; Javed et al., 2020; Krueger et al., 2020; Mitrovic et al., 2020), (group) distributional robust optimization (Sagawa et al., 2019), and meta-learning algorithms (Li et al., 2018). Theoretical understandings on the invariant representation have also been stutied (Zhao et al., 2020a). Other works also study how the inductive bias of models helps to generalize or extrapolate (Xu et al., 2021).