Estimating Generalization under Distribution Shifts via Domain-Invariant Representations

Ching-Yao Chuang [1] **Antonio Torralba** [1] **Stefanie Jegelka** [1]

Abstract

When machine learning models are deployed on a test distribution different from the training distribution, they can perform poorly, but overestimate their performance. In this work, we aim to better estimate a model's performance under distribution shift, without supervision. To do so, we use a set of domain-invariant predictors as a proxy for the unknown, true target labels. Since the error of the resulting risk estimate depends on the target risk of the proxy model, we study generalization of domain-invariant representations and show that the complexity of the latent representation has a significant influence on the target risk. Empirically, our approach (1) enables self-tuning of domain adaptation models, and (2) accurately estimates the target error of given models under distribution shift. Other applications include model selection, deciding early stopping and error detection.

1. Introduction

In many applications, machine learning models are deployed on data whose distribution is different from that of the training data. For instance, self-driving cars must be able to adapt to different weather, change of landscape or traffic, i.e., conditions can change at prediction time. But often, collecting large-scale supervised data on the shifted prediction domain is prohibitively expensive or impossible. While we may hope that the model generalizes to this new data distribution, *estimating empirically* how well a given model will actually generalize is challenging without labels.

Indeed, estimating the *adaptability*, i.e., the generalization to the target distribution, and the related potentially uncertain behavior of a prediction model, is a key concern for AI Safety (Amodei et al., 2016), motivating recent work on estimating target performance (Steinhardt & Liang, 2016; Platanios et al., 2014).

In this work, we develop a new idea for estimating performance under distribution shift, by drawing connections with domain adaptation. Necessarily, any method for estimating target performance must make some assumptions. Our method assumes the existence of a domain adaptation model that generalizes well from source (training) to target (test). Given the empirical success of domain adaptation, this assumption is met in many practical settings. A prominent class of domain adaptation models, *domain-invariant representations (DIR)* (Ben-David et al., 2007; Long et al., 2015; Ganin et al., 2016), learns a latent, joint representation of source and target data, and a predictor from the latent space to the output labels. In particular, we use a set of "check" DIR models as a proxy for the unknown, true target labels. If there exist "good" domain adaptation models, i.e., they achieve low source and presumably low target error, and those models disagree with the model h we want to evaluate, then the target risk of h is potentially high, and we should not trust it. Our experiments show that this leads to accurate estimates of target error that outperform previous methods.

This idea relies on good domain adaptation models, i.e., our check models should predict well on the target distribution, and not disagree too much with each other. But, evaluating a domain adaptation model itself on the target distribution is an unsolved problem. Hence, we begin by studying the target error of DIR. We observe that in general, DIR is much more sensitive to model complexity than supervised learning on the source distribution. In particular, the complexity of the representation encoder is key for target generalization and for selecting the set of check models, and points to an important model selection problem. For deep neural networks, this model selection problem essentially means how to optimally divide the network into an encoder and predictor part. Yet, this model selection ideally demands an estimate of target generalization, which we are developing.

We show that, with our framework for estimating target error, it is possible to let DIR models self-tune to find the optimal model complexity. The resulting models achieve good target generalization, and estimate target error of other

[1]CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Ching-Yao Chuang <cychuang@mit.edu>.

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

models well. Our approach applies to estimating the target error of a single or a class of models, and to predicting point-wise error. Hence, it can be used, e.g., for judging reliability and for model selection. Empirically, we examine our theory and algorithms on sentiment analysis (Amazon review dataset), digit classification (MNIST, MNIST-M, SVHN) and general object classification (Office-31). In short, this work makes the following contributions:

- We develop a generic method for estimating the error of a given model on a new data distribution.
- We show, theoretically and empirically, the important role of embedding complexity for domain-invariant representations.
- Our empirical results reflect our analyses and show that the proposed methods work well in practice.

2. Related Work

Estimating risk with distribution shifts. Estimating model risk on distributions different from the training distribution is important, but difficult with unlabeled data. Platanios et al. (2014) construct multiple models based on different views of the data and estimate the risk by calculating agreement rates across models. Steinhardt & Liang (2016) estimate the model's error on distributions very different from the training distribution by assuming a conditional independence structure of the data. Platanios et al. (2017) use logical constraints on the data to estimate classification accuracy. Recently, Elsahar & Gallé (2019) evaluate both confidence score and $\mathcal{H}\Delta\mathcal{H}$-divergence to predict performance drop under domain shift. We compare to those methods in the experiments. Different from previous works, we leverage domain-invariant classifiers as proxy target labels. Our method is general in the sense that it can predict the target risk for both domain adaptation and general supervised models.

Domain-invariant representations. DIRs are learned by minimizing a divergence between the embedding of source and target data, and existing approaches for learning DIRs differ in the divergence measure they use. Examples include domain adversarial learning (Ganin & Lempitsky, 2015; Tzeng et al., 2015; Ganin et al., 2016), maximum mean discrepancy (MMD) (Long et al., 2014; 2015; 2016) and Wasserstein distance (Courty et al., 2016; 2017; Shen et al., 2018; Lee & Raginsky, 2018).

Several theoretical frameworks have been proposed to analyze domain-invariant representations. One approach is to bound the target risk by assuming source and target domain share common support. Wu et al. (2019) show that exact matching of source and target distributions can increase target risk if label distributions differ between source and target. Johansson et al. (2019) propose generalization bounds based on the overlap of the supports of source and target distribution. However, the assumption of common support fails in most standard benchmarks for domain adaptation. Another line of work leverages the $\mathcal{H}\Delta\mathcal{H}$-divergence proposed by Ben-David et al. (2007). Shu et al. (2018) points out that learning domain-invariant representations with disjoint supports can still achieve maximal $\mathcal{H}\Delta\mathcal{H}$-divergence. Recently, Zhao et al. (2019) establish lower and upper bounds on the risk when label distributions between source and target domains differ.

3. Unsupervised Domain Adaptation

For simplicity of exposition, we consider binary classification with input space $\mathcal{X} \subseteq \mathbb{R}^n$ and output space $\mathcal{Y} = \{0, 1\}$. The learning algorithm obtains two datasets: labeled source data $\mathcal{X}_S$ from distribution p_S, and unlabeled target data $\mathcal{X}_T$ from distribution p_T. We will use p_S and p_T to denote the joint distribution on data and labels X, Y and the marginals, i.e., $p_S(X)$ and $p_S(Y)$. Unsupervised domain adaptation seeks a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ in a hypothesis class $\mathcal{H}$ that minimizes the risk in the target domain measured by a loss function ℓ (here, zero-one loss):

$$R_T(h) = \mathbb{E}_{x,y\sim p_T}[\ell(h(x), y)]. \tag{1}$$

We do not assume common support in source and target domain.

3.1. Domain-invariant Representations

A common approach to domain adaptation is to learn a joint embedding $g : \mathcal{X} \to \mathcal{Z}$ of source and target data (Ganin et al., 2016; Tzeng et al., 2017). The idea is that aligning source and target distributions in a latent space $\mathcal{Z}$ results in a domain-invariant representation, and hence a subsequent classifier $f : \mathcal{Z} \to \mathcal{Y}$ will generalize from source to target. Formally, this results in the following objective function on the hypothesis $h = fg := f \circ g$, where we minimize a divergence d between the distributions $p^g_S(Z), p^g_T(Z)$ of source and target after the mapping $Z = g(X) \in \mathcal{Z}$:

$$\min_{f\in\mathcal{F},g\in\mathcal{G}} R_S(fg) + \alpha d(p^g_S(Z), p^g_T(Z)). \tag{2}$$

The divergence d could be, e.g., the Jensen-Shannon (Ganin et al., 2016) or Wasserstein distance (Shen et al., 2018). In this paper, we denote the hypothesis class of the entire model h as $\mathcal{H}$, the class of embeddings by $\mathcal{G}$, and the class of predictors by $\mathcal{F}$.

3.2. Upper Bounds on the Target Risk

Ben-David et al. (2007) introduced the $\mathcal{H}\Delta\mathcal{H}$-divergence to bound the worst-case loss from extrapolating between domains. Let $R_D(h, h') = \mathbb{E}_{x\sim D}[\ell(h(x), h'(x))]$ be the

expected disagreement between two hypotheses and an extension of the notation $R_D(h) = R_T(h, h_{\text{true}})$, where h_{true} are the true labels. The $\mathcal{H}\Delta\mathcal{H}$-divergence measures whether there is any pair of hypotheses whose disagreement (risk) differs a lot between source and target distribution.

Definition 1. ($\mathcal{H}\Delta\mathcal{H}$-divergence) *Given two domain distributions p_S and p_T over $\mathcal{X}$, and a hypothesis class $\mathcal{H}$, the $\mathcal{H}\Delta\mathcal{H}$-divergence between p_S and p_T is*

$$d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T) = \sup_{h,h' \in \mathcal{H}} |R_S(h, h') - R_T(h, h')|.$$

The $\mathcal{H}\Delta\mathcal{H}$-divergence is determined by the discrepancy between source and target distribution and the complexity of the hypothesis class $\mathcal{H}$. This divergence allows us to bound the target risk:

Theorem 2. (Ben-David et al., 2010) *For all hypotheses $h \in \mathcal{H}$, the target risk is bounded as*

$$R_T(h) \leq R_S(h) + d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T) + \lambda_{\mathcal{H}}, \qquad (3)$$

where $\lambda_{\mathcal{H}}$ is the best joint risk

$$\lambda_{\mathcal{H}} \coloneqq \inf_{h' \in \mathcal{H}} [R_S(h') + R_T(h')].$$

Similar results exist for continuous labels (Cortes & Mohri, 2011; Mansour et al., 2009). Theorem 2 has been an influential theoretical result in domain adaptation, and motivated work on domain invariant representations. For example, recent work (Ganin et al. (2016); Johansson et al. (2019)) applied Theorem 2 to the hypothesis class $\mathcal{F}$ that maps the representation space $\mathcal{Z}$ induced by an encoder g to the output space:

$$R_T(fg) \leq R_S(fg) + d_{\mathcal{F}\Delta\mathcal{F}}(p_S^g(Z), p_T^g(Z)) + \lambda_{\mathcal{F}}(g) \qquad (4)$$

where $\lambda_{\mathcal{F}}(g)$ is the best hypothesis risk with fixed g, i.e., $\lambda_{\mathcal{F}}(g) \coloneqq \inf_{f' \in \mathcal{F}} [R_S(f'g) + R_T(f'g)]$. The $\mathcal{F}\Delta\mathcal{F}$ divergence implicitly depends on the fixed g and can be small if g provides a suitable representation. However, if g induces a wrong alignment, then the best hypothesis risk $\lambda_{\mathcal{F}}(g)$ is large with any function class $\mathcal{F}$.

4. Estimating Target Risk: Main Idea

Our goal is to estimate the error of a given, learned model h on a target distribution p_T, without observing true labels on the target. Let h_{true} be the true labeling, and $h^* = \arg\inf_{h \in \mathcal{P}} R_T(h)$. By the triangle inequality, $R_T(h) = R_T(h, h_{\text{true}}) \leq R_T(h, h^*) + R_T(h^*)$. The main idea underlying our approach is to obtain an upper bound on $R_T(h)$ by replacing h^* with candidates from a set of proxy models $\mathcal{P}$ that we also call *check models*.

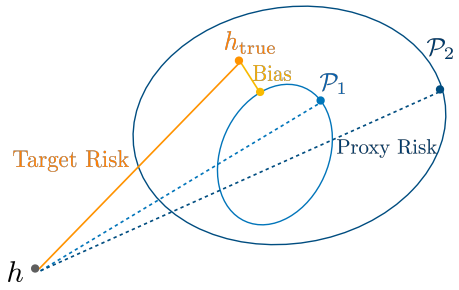


Figure 1. Conceptual illustration of proxy risk: the orange line is the true target risk, the dashed lines are the proxy risks with respect to two sets of check models, $\mathcal{P}_1$ and $\mathcal{P}_2$ where $\mathcal{P}_1 \subseteq \mathcal{P}_2$. By construction, although $h_{\text{true}} \in \mathcal{P}_2$ (zero bias), the proxy risk calculated with $\mathcal{P}_2$ is not tight enough to approximate the target risk well. In contrast, $\mathcal{P}_1$ has a nonzero bias (yellow line) but tighter estimation.

Lemma 3. *Given a hypothesis class $\mathcal{P}$, for all $h \in \mathcal{H}$,*

$$R_T(h) \leq \underbrace{\sup_{h' \in \mathcal{P}} R_T(h, h')}_{\text{Proxy Risk}} + \underbrace{\inf_{h' \in \mathcal{P}} R_T(h')}_{\text{Bias}}. \qquad (5)$$

We prove all theoretical results in the Appendix A. The first term in Lemma 3 measures the maximal disagreement (risk) between the hypothesis h and a check model $h' \in \mathcal{P}$, instead of h^*. The second term measures how good the check models are. For this bound to be tight, $\mathcal{P}$ must contain a good hypothesis. At the same time, $\mathcal{P}$ should not contain any unnecessarily disagreeing hypotheses, otherwise the proxy risk will be too large. Figure 1 provide an illustrative example of the idea.

Connection to Domain Adaptation The proxy risk can be estimated empirically. If the bias term is small, namely, there exists a good hypothesis in the check models, then the proxy risk itself is a good estimate of an upper bound on $R_T(h)$. It remains to determine the set $\mathcal{P}$.

Lemma 4. *Given a hypothesis class $\mathcal{P}$, for all $h \in \mathcal{H}$,*

$$\underbrace{|\sup_{h' \in \mathcal{P}} R_T(h, h') - R_T(h)|}_{\text{Estimation Error}} \leq \sup_{h' \in \mathcal{P}} R_T(h'). \qquad (6)$$

Lemma 4 links our approach with domain adaptation: the target risk of the check models affects the error of estimating risk via the proxy risk. This motivates domain adaptation models as check models, because they are designed to minimize the target risk. In Section 7, where we develop this idea in detail, $\mathcal{P}$ is the set of all DIR models that have low DIR objective. To understand the tightness of our proxy risk-based estimation, we begin with a closer look at what affects the target risk of domain invariant representations.

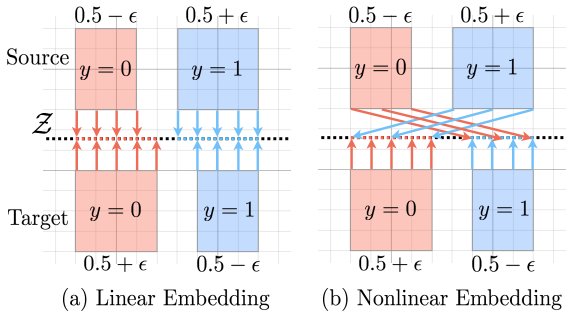


Figure 2. Illustrative example in 2D. The 1D representation space is illustrated as a dotted line, and arrows indicate the embedding from 2D to 1D. (a) Optimal embedding when $\mathcal{G}$ is the class of linear functions. (b) Optimal embedding with a complex nonlinear encoder class.

5. Understanding the Adaptability of DIR

In this section, we aim to better understand what affects target risk and ambiguity on the target for domain invariant representations. The bound (4) highlights the effect of the complexity of the prediction models $\mathcal{F}$, and of the quality of the alignment via the embedding g. But, as the following toy example illustrates, another important component is the complexity of the embedding class $\mathcal{G}$.

Toy Example. Figure 2 shows a binary classification problem in 2D with disjoint support and a slight shift in the label distributions from source to target: $p_S(y=1) = p_T(y=1) + 2\epsilon$. For a 1D latent representation space, if we allow arbitrary maps $g \in \mathcal{G}$, then, e.g., a complicated nonlinear map as in Figure 2(b) can achieve zero DIR objective value (equation (2)), but maximum target risk $R_T(fg) = 1$. If we restrict $\mathcal{G}$ to linear maps, then a map g as in Figure 2(a) achieves optimal DIR objective value of 2ϵ, and minimum target risk. Hence, a too powerful embedding class $\mathcal{G}$ can increase ambiguity, variance and hence target risk.

Empirical Effect of Complexity. In the experiments in Section 5.2, e.g., in Figure 4, we observe that throughout, the complexity of $\mathcal{G}$ has a noticeable effect on the target risk. In contrast, in analogous experiments for the predictor f shown in Figure 3(a), the predictor class $\mathcal{F}$ has a much weaker influence. Likewise, Figure 3(b) demonstrates that generalization on the source domain, i.e., "normal" generalization of supervised learning, is also much less affected by the model complexity. In summary, empirically, the adaptability of domain-invariant representations is more sensitive to model complexity than supervised learning, and in general most sensitive to the complexity of the embedding class $\mathcal{G}$. Hence, we focus on embedding complexity.

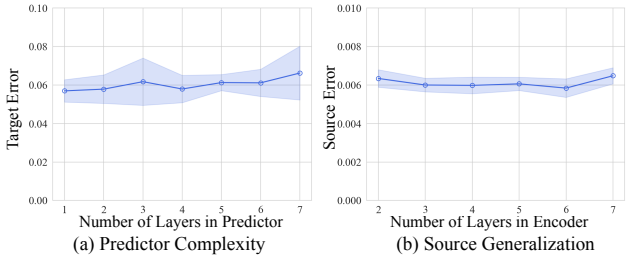


Figure 3. (a) Effect of predictor complexity on target generalization for MNIST→MNIST-M and (b) effect of embedding complexity on source (MNIST) generalization.

5.1. Bounds for Domain-invariant Representations

Motivated by the above observations, we next expose how the bound on the target risk depends on the complexity of the embedding class. Directly applying Theorem 2 to the composition $\mathcal{H} = \mathcal{F}\mathcal{G}$ treats both jointly and does not make the role of the embedding very explicit. Instead, we define a version of the $\mathcal{H}\Delta\mathcal{H}$-divergence that explicitly measures variation of the embeddings in $\mathcal{G}$:

Definition 5. ($\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence) *For two domain distributions p_S and p_T over $\mathcal{X}$, an encoder class $\mathcal{G}$, and predictor class $\mathcal{F}$, the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence between p_S and p_T is*

$$d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S, p_T) = \sup_{f\in\mathcal{F};\, g,g'\in\mathcal{G}} |R_S(fg, fg') - R_T(fg, fg')|.$$

Importantly, the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence is smaller than the $(\mathcal{F}\mathcal{G})\Delta(\mathcal{F}\mathcal{G})$-divergence, since the two hypotheses in the supremum, fg and fg', share the same predictor f.

Theorem 6. *For all $f \in \mathcal{F}$ and $g \in \mathcal{G}$,*

$$R_T(fg) \le R_S(fg) + \underbrace{d_{\mathcal{F}\Delta\mathcal{F}}(p_S^g(Z), p_T^g(Z))}_{\text{Latent Divergence}} + \underbrace{d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S, p_T)}_{\text{Embedding Complexity}} + \lambda_{\mathcal{F}\mathcal{G}}(g). \quad (7)$$

where $\lambda_{\mathcal{F}\mathcal{G}}(g)$ is a variant of the best in-class joint risk:

$$\lambda_{\mathcal{F}\mathcal{G}}(g) = \inf_{f'\in\mathcal{F}, g'\in\mathcal{G}} 2R_S(f'g) + R_S(f'g') + R_T(f'g').$$

This target generalization bound is small if (C1) the source risk is small, (C2) the latent divergence is small, because the domains are well-aligned and/or $\mathcal{F}$ is restricted, (C3) the complexity of $\mathcal{G}$ is restricted to avoid overfitting of alignments, and (C4) good source and target risk is in general achievable with $\mathcal{F}$ and $\mathcal{G}$ and the encoder is good for the source domain. The bound naturally explains the tradeoff we observe in the subsequent experiments between the following terms: the latent divergence (which increases with complexity of $\mathcal{F}$ and decreases with complexity of $\mathcal{G}$), embedding complexity (which increases with complexity of

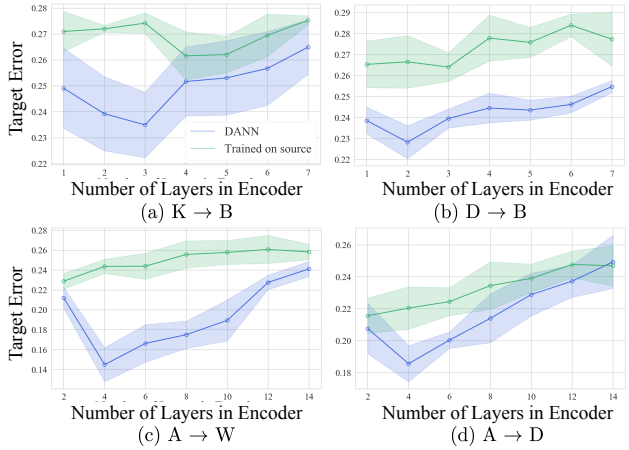


Figure 4. Effect of embedding complexity on target risk. First row: Sentiment Classification. Second row: Object Classification.

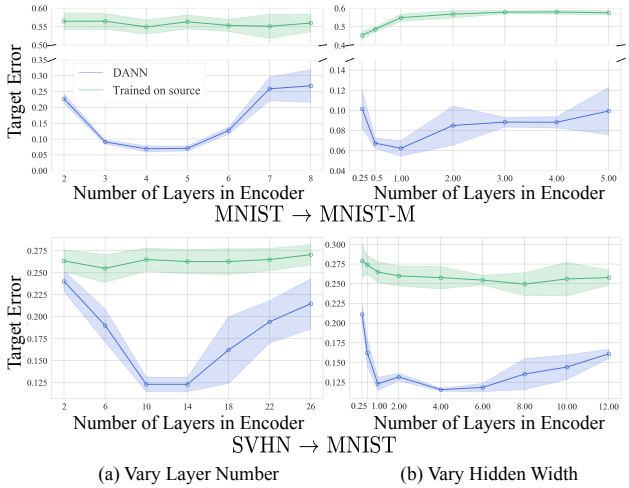


Figure 5. Effect of embedding complexity on target risk for digit classification.

$\mathcal{F}$ and $\mathcal{G}$), and the best in-class joint risk (which decreases with complexity of $\mathcal{F}$ and $\mathcal{G}$). Overly expressive encoders suffer from a larger embedding complexity penalty, while smaller encoders fail to minimize the latent divergence.

5.2. Experiments

Next, we probe the effect of embedding complexity empirically, via experiments with several standard benchmarks: sentiment analysis (Amazon reviews dataset), digit classification (MNIST, MNIST-M, SVHN) and general object classification (Office-31). In all experiments, we train DANN (Ganin et al., 2016), which measures the latent divergence via a domain discriminator (Jensen Shannon divergence). A validation set from the source domain is used as an early stopping criterion during learning. In all experiments, we use a progressive training strategy for the discriminator (Ganin et al., 2016). We primarily consider two types of complexity: number of layers and number of hidden neurons. In all embedding complexity experiments, we retrain each model for 5 times and plot the mean and standard deviation of the target error. Dataset and architecture details may be found in the appendix.

Sentiment Classification. We first examine complexity tradeoffs on the Amazon reviews data, which has four domains: books (B), DVD disks (D), electronics (E), and kitchen appliances (K). The hypothesis class are multi-layer ReLU networks. We show results for K→B, and D→B in Figure 4 and defer the rest to the appendix. To probe the effect of embedding complexity, we fix the predictor class to 4 layers and vary the number of layers of the embedding. Figure 4 shows that the target error decreases initially, and then increases as more layers are added to the encoder.

Object Classification. Office-31 (Saenko et al., 2010), one of the most widely used benchmarks in domain adaptation, contains three domains: Amazon (A), Webcam (W), and DSLR (D) with 4,652 images and 31 categories. We show results for A→W, A→D in Figure 4, and the rest in the Appendix B. To overcome the lack of training data, similar to (Li et al., 2018; Long et al., 2018), we use ResNet-50 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) for feature extraction. With the extracted features, we adopt multi-layer ReLU networks as hypothesis class. Again, we increase the depth of the encoder while fixing the depth of the predictor to 2. Even with a powerful feature extractor, the embedding complexity tradeoff still exists.

Digit Classification. We next verify our findings on standard domain adaptation benchmarks: MNIST→MNIST-M (M→M-M) and SVHN→MNIST (S→M). We use standard CNNs as the hypothesis class.

To analyze the effect of the embedding complexity, we augment the original two-layer CNN encoders with 1 to 6 additional CNN layers for M→M-M and 1 to 24 for S→M, leaving other settings unchanged. Figure 5(a) shows the results. Again, the target error decreases initially and increase as the encoder becomes more complex. Notably, the target error increases by 19.8% in M→M-M and 8.8% in S→M compared to the optimal case, when more layers are added to the encoder. We also consider the width of hidden layers as a complexity measure, while fixing the depth of both encoder and predictor. The results are shown in Figure 5(b). This time, the decrease in target error is not significant compared to increasing encoder depth. This suggests that depth plays a more important role than width in learning domain-invariant representations. In the appendix, we

also investigate the importance of inductive bias and for domain-invariant representations.

6. Division for Multilayer Neural Networks

Next, we adapt the bound in Theorem 6 to multilayer networks. Specifically, we consider the number of layers as a measure of complexity. Assume $\mathcal{H}$ is the class of N-layer feedforward neural networks with a fixed width. The model $h \in \mathcal{H}$ can be decomposed as $h = f_i g_i \in \mathcal{F}_i \mathcal{G}_i = \mathcal{H}$ for $i \in \{1, 2, \ldots, N-1\}$, where the embedding g_i is formed by the first layer to the i-th layer and the predictor f_i is formed by the $(i+1)$-th layer to the last layer. We can then rewrite the bound in Theorem 6 in layer-specific form:

$$R_T(h) \leq R_S(h) + \underbrace{d_{\mathcal{F}_i \Delta \mathcal{F}_i}(p_S^{g_i}(Z), p_T^{g_i}(Z))}_{\text{Latent Divergence in } i\text{-th layer}} + \underbrace{d_{\mathcal{F}_i \mathcal{G}_i \Delta \mathcal{G}_i}(p_S, p_T)}_{\text{Embedding Complexity w.r.t } \mathcal{G}_i} + \lambda_{\mathcal{F}_i \mathcal{G}_i}(g_i). \quad (8)$$

Minimizing the domain-invariant loss in different layers leads to different tradeoffs between fit and complexity penalties. This is reflected by the following inequalities that relate different layer divisions.

Proposition 7. *In an N-layer feedforward neural network $h = f_i g_i \in \mathcal{F}_i \mathcal{G}_i = \mathcal{H}$, the following inequalities hold for all $i \leq j \leq N-1$:*

$$d_{\mathcal{F}_i \mathcal{G}_i \Delta \mathcal{G}_i}(p_S, p_T) \leq d_{\mathcal{F}_j \mathcal{G}_j \Delta \mathcal{G}_j}(p_S, p_T)$$
$$d_{\mathcal{F}_i \Delta \mathcal{F}_i}(p_S^{g_i}(Z), p_T^{g_i}(Z)) \geq d_{\mathcal{F}_j \Delta \mathcal{F}_j}(p_S^{g_j}(Z), p_T^{g_j}(Z)).$$

Proposition 7 states that a deeper embedding allows for better alignments and simultaneously reduces the depth (power) of $\mathcal{F}$; both reduce the latent divergence. At the same time, it incurs a larger $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence. This is a tradeoff within the fixed combined hypothesis class $\mathcal{H}$.

This suggests that there might be an optimal division that minimizes the bound on the target risk. In practice, this translates into the question: *in which intermediate layer should we optimize the domain-invariant loss?*

6.1. Experiments

Next, we examine the embedding complexity tradeoff when the total number of layers is fixed, with the setup of Section 5.2. We first probe the tradeoff when the total number of layers is fixed to 8 for sentiment classification. The results in Figure 6 suggest that there exists an optimal setting for all tasks. Next, we fix the total number of CNN layers of the neural network to 7 and 26 for M→M-M and S→M, respectively, and optimize the domain-invariant loss in different intermediate layers. The results again show a "U-curve", indicating the existence of an optimal division. Even with this fixed total size of the network ($\mathcal{H}$), the performance gap between different divisions can still reach 19.5% in M→M-M and 10.4% in S→M. Similar results can be seen in object classification with fixed total network depth 14. More experimental results may be found in the appendix.

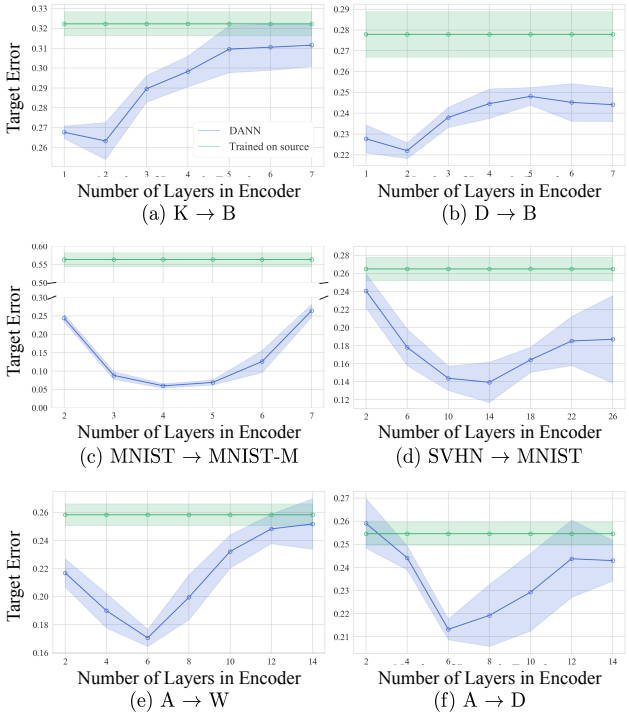


Figure 6. The effect of layer division in fixed-depth neural networks. First row: Sentiment Classification; Second row: Digit Classification; Third row: Object Classification.

In summary, empirically, there is an optimal division with minimum target error, suggesting that for a fixed $\mathcal{H}$, i.e., total network depth, not all divisions are equal. We will provide methods to predict the optimal division in Section 8.1.

7. Estimating Target Risk

Upper bounds such as the ones in Sections 3 and 5 are useful for theoretical insights and intuition about effects, but hard to compute explicitly. Here, we return to the idea in Section 4 to estimate the target risk of a given model by using a selected set $\mathcal{P}$ of check models as proxies. In particular, given the bound in Lemma 3, we use $\sup_{h' \in \mathcal{P}} R_T(h, h')$ as an estimate of the target risk. Our approach works well if $\inf_{h' \in \mathcal{P}} R_T(h')$ is small, i.e., if there is a good target prediction model in $\mathcal{P}$.

We define the set $\mathcal{P}$ of check models to be all domain-invariant classifiers that achieve low DIR objective value, i.e., they achieve low source risk and align the source and target distributions well:

$$\mathcal{P}^{\epsilon}_{\mathcal{FG}} = \{h = fg \in \mathcal{FG} | R_S(h) + \alpha d(p_S^g(Z), p_T^g(Z)) \leq \epsilon\}.$$

Hence, we implicitly assume that there exists some DIR model that achieves low target risk.

7.1. Connection to Embedding Complexity

How good is the resulting proxy risk as an estimate of the target risk of h? Lemma 4 states that the target risk of the check models gives an upper bound on the estimation error:

$$|\sup_{h' \in \mathcal{P}^{\epsilon}_{\mathcal{FG}}} R_T(h, h') - R_T(h)| \leq \sup_{h' \in \mathcal{P}^{\epsilon}_{\mathcal{FG}}} R_T(h'). \quad (9)$$

Recall that the set $\mathcal{P}^{\epsilon}_{\mathcal{FG}}$ comprises all DIR models that achieve low DIR objective value. If $\sup_{h' \in \mathcal{P}^{\epsilon}_{\mathcal{FG}}} R_T(h')$ is large, then the DIR objective is not sufficiently determining to identify a good target classifier, and generalization to the target is impossible. The results in Section 5 suggest that the embedding complexity of the DIR check models plays an important role for target generalization. To minimize the estimation error, we should select a class of DIR models with suitable embedding complexity, i.e., one with an optimal division. As we will show in Section 8.1, it is possible to also use our ideas to let DIR models self-tune, to find the optimal embedding complexity.

7.2. Computing the Target Risk Estimator

To approximate the proxy risk $\sup_{h' \in \mathcal{P}^{\epsilon}_{\mathcal{FG}}} R_T(h, h')$, we aim to maximize the disagreement under model constraints:

$$\max_{f'g' \in \mathcal{FG}} R_T(h, f'g') \quad (10)$$

$$\text{s.t. } R_S(f'g') + \alpha d(p_S^{g'}(Z), p_T^{g'}(Z)) \leq \epsilon \quad (11)$$

Computationally, it is more convenient to replace the constraint with a penalty via Lagrangian relaxation:

$$\max_{f'g' \in \mathcal{FG}} R_T(h, f'g') - \lambda(R_S(f'g') + \alpha d(p_S^{g'}(Z), p_T^{g'}(Z))$$

where $\lambda > 0$. We use empirical estimates for R_T, R_S, and minimize the empirical objective via standard stochastic gradient descent.

Algorithm 1 provides details about approximating the proxy risk[1]. In brief, we first pretrain $h' = f'g'$, and then maximize the disagreement with h under constraints. Empirically we maximize the disagreement on the training set and check the constrains $R_S(f'g') + \alpha d(p_S^{g'}(Z), p_T^{g'}(Z)) \leq \epsilon$ with the validation set.

8. Experiments

We evaluate our method on two broad tasks: model selection for DIR models and estimating target risk of any given model. Throughout, the experimental settings and the model architectures are the same as in Section 6.1.

[1]The code is available at https://github.com/chingyaoc/risk-estimation-dip.

Algorithm 1 Computing Proxy Risk

Require: Target hypothesis h; Check model class $\mathcal{H} = \mathcal{FG}$; S_S and S_T: labeled source dataset and unlabeled target dataset; $\alpha, \lambda, \epsilon$: tradeoff parameters; T_1: Epochs for training domain-invariant classifier; T_2: Epochs for maximizing the disagreement.

▷ *Pretrain check model h'*
Initialize $h' = f'g' \in \mathcal{FG}$
Train h' for T_1 epochs to minimize $R_S(h') + \alpha d(p_S^{g'}(Z), p_T^{g'}(Z))$

▷ *Maximize the disagreement*
Initialize MaxRisk $= 0$
for $i = 1, \ldots, T_2$ **do**
  Train h' for one epoch to minimize $-R_T(h, h') + \lambda(R_S(f'g') + \alpha d(p_S^{g'}(Z), p_T^{g'}(Z))$
  if $R_S(f'g') + \alpha d(p_S^{g'}(Z), p_T^{g'}(Z) \leq \epsilon$ and $R_T(h, h') \geq$ MaxRisk **then**
    Set MaxRisk $= R_T(h, h')$
  end if
end for
return MaxRisk

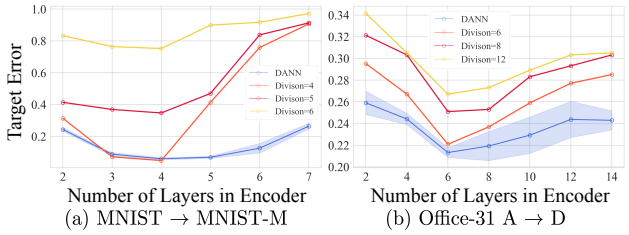
8.1. Model Selection for DIR

Estimating Optimal Network Division We begin with estimating the optimal layer division of a DIR model into encoder and predictor that minimizes target risk. By Lemma 4, this will yield a good class of check models. To estimate the DIR models' target risk, we follow the same strategy as in Section 4, but for a *class* of models: the worst target error for division i can be bounded with a second level of proxy classifiers:

$$\sup_{h \in \mathcal{P}^{\epsilon}_{\mathcal{F}_i\mathcal{G}_i}} R_T(h) \leq \underbrace{\sup_{\substack{h \in \mathcal{P}^{\epsilon}_{\mathcal{F}_i\mathcal{G}_i} \\ h' \in \mathcal{P}^{\epsilon}_{\mathcal{F}'\mathcal{G}'}}} R_T(h, h')}_{\text{Worst In-class Proxy Risk}} + \inf_{h' \in \mathcal{P}^{\epsilon}_{\mathcal{FG}}} R_T(h').$$

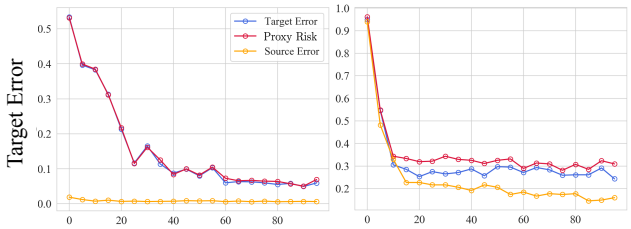
We select the division that minimizes the worst in-class proxy risk.

Computationally, we adopt the approach from Section 7.2 to approximate the worst in-class proxy risks. Figure 7 shows the true target test error for a DIR model, computed with labels (blue line), for different divisions, compared to our in-class proxy risk estimates. The different lines correspond to different second-level check models. The results suggest that (1) we can accurately estimate the best layer division *without supervision*, and (2) this self-tuning strategy is robust to the choice of second-level check models.



(a) MNIST → MNIST-M (b) Office-31 A → D

Figure 7. Estimating optimal network division via in-class proxy risk. Different colors indicate second-level check models with different divisions. "DANN" is the true target test error.



(a) MNIST → MNIST-M (b) Office-31 A → D

Figure 8. Estimating stopping criteria: Target risk, source risk, and the proxy risk across training procedures.

Estimating Stopping Criteria Without access to target labels, it is nontrivial to determine when to perform early stopping for DIR (Prechelt, 1998). Previous works leverage source error or self-labeled validation sets (Ganin & Lempitsky, 2015) to serve as stopping criterion. Figure 8 shows the target error along with the source error and proxy error during the training of DANNs. The check model has the same architecture as the candidate model and the predictions are approximated with optimal division. In both experiments, proxy risks are well aligned with target errors. Notably, the proxy risk is almost the same as the target risk in MNIST→MNIST-M. These results suggest that proxy risk is a good criterion for early stopping.

8.2. Estimating Performance Drop of Supervised Learning under Domain Shift

In this section, we aim to estimate the target risk of non-adaptive models that are trained only on the source, i.e., standard supervised learning. We compare our method with (1) Ben-David et al. (2010)'s bound (Bound (3)) and (2) a method based on confidence scores (Elsahar & Gallé, 2019). For the first approach, we approximate the bound (3) by estimating $R_S(h) + d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T)$ with $\mathcal{H} = \{h \in \mathcal{H} | R_S(h) \leq \epsilon\}$, via an approach similar to that in Section 7.2 (details are in the appendix). For the second approach, let $q_h(x)$ be the probability score (i.e., the max value of the softmax output) of hypothesis h for $x \in \mathcal{X}$. Elsahar & Gallé (2019) compute confidence scores as the drop in average probability scores:

$$\text{CONF}_{S,T}(h) = \mathbb{E}_{x\sim p_S}[q_h(x)] - \mathbb{E}_{x\sim p_T}[q_h(x)]. \quad (12)$$

The target risk is then predicted by $R_S(h) + \text{CONF}_{S,T}(h)$. We examine the methods on object classification, which contains 6 source/target pairs and digit classification, which has 12 source/target pairs after adding USPS. The architectures are fixed for the same task. To estimate risks on the new distribution, we consider domain-invariant classifiers with different divisions as check models. The encoder in the "standard" and "complex" check models have 4/6 and 6/12 layers, respectively, for digit/object classification. The check models and supervised prediction models share the same architecture.

Method	Ours (Standard)		Ben-David et al.		Conf Score	
Metric	Err	PCC	Err	PCC	Err	PCC
SL-Digit	**0.073**	**0.880**	0.450	0.585	0.406	0.103
SL-Object	**0.034**	**0.995**	0.281	0.945	0.106	0.551
DIR-Digit	**0.043**	**0.957**	0.124	0.693	0.242	0.273
DIR-Object	**0.021**	**0.986**	0.114	0.932	0.088	0.459

Table 1. Estimating the target risk. We show the average error (lower is better) and Pearson correlation coefficient (higher is better) on different tasks. We estimate the target risk for supervised learning models (SL) and adaptive models (DIR).

The results in Figure 9 (left) show that our methods consistently provide much better predictions than the baselines. With a complex encoder in the check model, we slightly overestimate the target risk, aligned with our theory: check models with properly controlled embedding complexity result in better prediction. Ben-David et al. (2010)'s bound tends to overestimate the target risk, suggesting that the $\mathcal{H}\Delta\mathcal{H}$-divergence is too pessimistic for empirical estimation. In contrast, the confidence score approach largely underestimates the target risk. Table 1 shows the quantitative results (SL-Digit and SL-Object): the average absolute error over domain pairs and the Pearson correlation coefficient between target risks and predictions. Our methods outperform baselines by a large margin in both metrics.

8.3. Estimating Adaptability between Domains

In this section, we repeat the experiments in Section 8.2, this time for estimating the target risk of adaptive, domain-invariant classifiers (DANNs). Different from the previous section, we tighten Ben-David et al. (2010)'s bound by setting $\mathcal{H} = \mathcal{P}^{\epsilon}_{\mathcal{FG}}$. Figure 9 (right) shows the results. Our method still consistently outperforms baselines in both tasks. Compared to estimating target risk for nonadaptive models, improvements in performance are observed for all methods, as Table 1 (DIR-Digit and DIR-Object) shows. Our methods produce estimates within a few percent of the true accuracy,

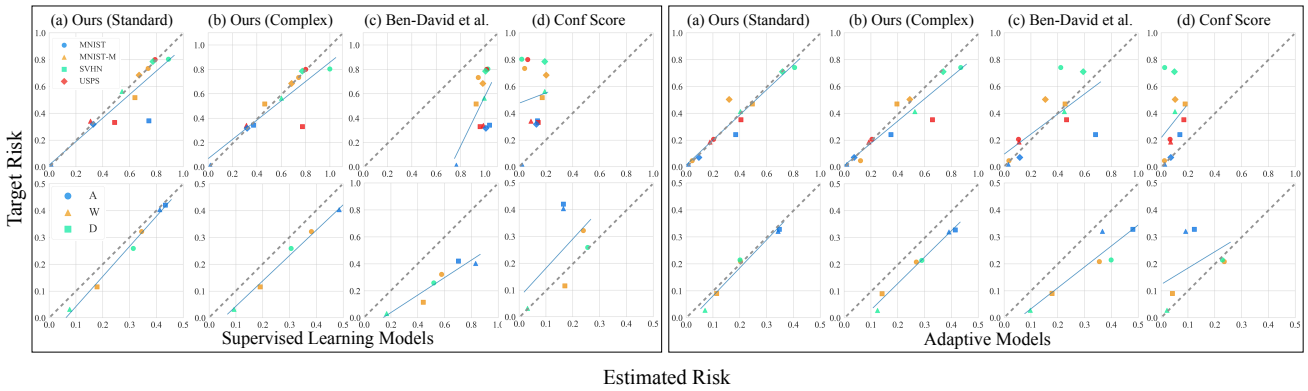


Figure 9. Estimating performance drop. First row: Digit Classification, second row: object classification. The dashed line represents perfect prediction (target risk = predicted risk). Shape and color of points indicates different source and target domain, respectively. Points beneath (above) the dashed line indicate overestimation (underestimation). The solid lines are regression lines.

while the baselines often underestimate (confidence score) or overestimate (Ben-David et al., 2010) the target risk.

8.4. Error Detection

Besides estimating the risk on new distributions, it is also important to know whether a prediction is reliable at a specific new test point. Our approach easily extends to predicting point-wise error, i.e., predicting misclassification at a target point x^*. Recall that to approximate the proxy error of h, we train a check model h' to maximize the disagreement with h. We use h' to predict misclassification: if $h'(x^*) \neq h(x^*)$, then we should not trust $h(x^*)$ and predict an error.

To evaluate this method quantitatively, we formulate error prediction as binary classification, and compute the F1 score of error detection on the target domain for supervised and adaptive models. The results, shown in Table 2, demonstrate that we can not only quantify the expected error in unseen distributions but also estimate the point-wise error accurately.

	Digit Classification		Object Classification	
	M→MM	S→M	A→D	A→W
SL	0.985	0.908	0.925	0.938
DIR	0.980	0.885	0.908	0.928

Table 2. F1 score of error detection on the target domain for supervised learning models (SL) and adaptive models (DIR).

9. Conclusion

In this paper, we made two contributions: (1) We leverage domain-invariant classifiers to empirically estimate the target risk, i.e., performance on a new, shifted, unlabeled dataset, of any given supervised or domain adaptation model. This approach applies to estimating risk on a data set for a single classifier, predicting point-wise error, and estimating the risk for a set of given classifiers, e.g., for model selection. (2) To obtain good estimators, we theoretically and empirically analyze the effect of embedding complexity on the target risk in domain-invariant representations. We observe that the embedding complexity is an important factor for adaptability to the target distribution, much more than the complexity of the predictor part, and more than its roe for non-adaptive, supervised learning.

Interesting directions of future work include adopting other domain adaptation algorithms as check models, and applying our approach to structured tasks, e.g., detection or segmentation.

Acknowledgements

This work was supported by the MIT-IBM Watson AI Lab, NSF CAREER Award 1553284, an NSF BIGDATA award and the MIT-MSR TRAC collaboration. We thank Tongzhou Wang, Joshua Robinson, Wei Fang, Wei-Chiu Ma, and Chen-Ming Chuang for helpful discussions and suggestions.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pp. 137–144, 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(1-2):151–175, 2010.

Cortes, C. and Mohri, M. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pp. 308–323. Springer, 2011.

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 *Advances in Neural Information Processing Systems*, pp. 3730–3739, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Elsahar, H. and Gallé, M. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2163–2173, 2019.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. *International conference on machine learning*, 2015.

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.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Johansson, F. D., Ranganath, R., and Sontag, D. Support and invertibility in domain-invariant representations. *International Conference on Artificial Intelligence and Statistics*, 2019.

Lee, J. and Raginsky, M. Minimax statistical learning with wasserstein distances. In *Advances in Neural Information Processing Systems*, pp. 2687–2696, 2018.

Li, S., Song, S., Huang, G., Ding, Z., and Wu, C. Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Transactions on Image Processing*, 27(9):4260–4273, 2018.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1410–1417, 2014.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. *International conference on machine learning*, 2015.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pp. 136–144, 2016.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.

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

Platanios, E., Poon, H., Mitchell, T. M., and Horvitz, E. J. Estimating accuracy from unlabeled data: A probabilistic logic approach. In *Advances in Neural Information Processing Systems*, pp. 4361–4370, 2017.

Platanios, E. A., Blum, A., and Mitchell, T. Estimating accuracy from unlabeled data. 2014.

Prechelt, L. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pp. 55–69. Springer, 1998.

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

Shen, J., Qu, Y., Zhang, W., and Yu, Y. Wasserstein distance guided representation learning for domain adaptation. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

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

Steinhardt, J. and Liang, P. Unsupervised risk estimation with only structural assumptions. 2016.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4068–4076, 2015.

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.

Wu, Y., Winston, E., Kaushik, D., and Lipton, Z. Domain adaptation with asymmetrically-relaxed distribution alignment. *International conference on machine learning*, 2019.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. On learning invariant representation for domain adaptation. *International conference on machine learning*, 2019.