A Unified Analysis of Generalization and Sample Complexity for Semi-Supervised Domain Adaptation*

Elif Vural† and Hüseyin Karaca‡

Abstract. Domain adaptation seeks to leverage the abundant label information in a source domain to improve classification performance in a target domain with limited labels. While the field has seen extensive methodological development, its theoretical foundations remain relatively underexplored. Most existing theoretical analyses focus on simplified settings where the source and target domains share the same input space and relate target-domain performance to measures of domain discrepancy. Although insightful, these analyses may not fully capture the behavior of modern approaches that align domains into a shared space via feature transformations. In this paper, we present a comprehensive theoretical study of domain adaptation algorithms based on *domain alignment*. We consider the joint learning of domain-aligning feature transformations and a shared classifier in a semi-supervised setting. We first derive generalization bounds in a broad setting, in terms of covering numbers of the relevant function classes. We then extend our analysis to characterize the sample complexity of domain-adaptive neural networks employing maximum mean discrepancy (MMD) or adversarial objectives. Our results rely on a rigorous analysis of the covering numbers of these architectures. We show that, for both MMD-based and adversarial models, the sample complexity admits an upper bound that scales quadratically with network depth and width. Furthermore, our analysis suggests that in semi-supervised settings, robustness to limited labeled target data can be achieved by scaling the target loss proportionally to the square root of the number of labeled target samples. Experimental evaluation in both shallow and deep settings lends support to our theoretical findings.

Key words. Domain adaptation, generalization bounds, domain-adaptive neural networks, maximum mean discrepancy, adversarial domain adaptation, sample complexity

MSC codes. 68Q32, 68T05, 68T07

1. Introduction.

Domain adaptation is a subfield of machine learning that aims to improve model performance in a target domain by leveraging the greater availability of labeled samples in a source domain. The main challenge in domain adaptation is to address the discrepancy between the source and target distributions, which can take various forms such as covariate shift [29], label shift [2, 42], as well as more challenging heterogeneous settings with source and target samples originating from different data spaces [39]. Early work in domain adaptation explored instance reweighting methods for covariate shift [26, 41], feature augmentation approaches [11, 12, 14], and techniques for learning feature projections or transformations [3, 37, 56]. More recently, in line with broader advances in data science, domain adaptation research over the last decade has largely shifted towards deep learning-

*Submitted to the editors February 17, 2026.

Funding: This work is supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK) 1515 Frontier R&D Laboratories Support Program for Türk Telekom 6G R&D Lab under project number 5249902 and 2210 National Graduate Scholarship Program.

†Department of Electrical and Electronics Engineering, METU, Ankara, Türkiye (velif@metu.edu.tr, http://blog.metu.edu.tr/velif/).

‡Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Türkiye and Türk Telekom, Ankara, Türkiye (huseyin.karaca@bilkent.edu.tr, hkaraca@turktelekom.com.tr, https://huseyin-karaca.github.io).

based techniques [39, 49]. Metrics such as maximum mean discrepancy (MMD) [31, 46, 22] lead to efficient solutions for aligning source and target domains across various applications [58, 52, 54, 55]. Adversarial architectures [21, 45, 43, 61] and reconstruction-based approaches using encoder-decoder structures [23, 6, 62] are also commonly employed.

Despite the variety of models and the diversity of solutions, the basic paradigm in domain adaptation - whether using shallow methods or neural networks- often boils down to first aligning the source and target domains by mapping them to a common space through feature transformations, followed by learning a hypothesis function, typically a classifier, in that shared domain. The alignment of the source and target distributions is achieved by minimizing a suitably defined *distribution distance* (also referred to as *domain discrepancy* or *distribution divergence*), with common choices including MMD [31], covariance-based metrics [40], and the Wasserstein distance [7, 9, 16]. Although domain adaptation algorithms have been successfully applied across a wide range of fields including computer vision, time-series analysis, and natural language processing [39, 61], surprisingly, the literature still lacks a thorough theoretical characterization of their performance. In particular, there is a notable gap in understanding the behavior of *domain alignment algorithms*, which we define as methods that explicitly map source and target domains to a common representation through feature transformations. In this paper, we focus on this important class of algorithms, and aim to provide a rigorous theoretical analysis of their performance.

Most existing theoretical analyses focus on understanding how the discrepancy between source and target domains affects the target-domain performance of classifiers trained to perform well on the source domain [38, 5, 33, 59, 13, 51]. While these studies provide useful insight into how models trained with abundant source labels generalize to a target domain with limited or no labeled data, they inherently assume that source and target data reside in the same space. Consequently, their results do not straightforwardly extend to the prevalent framework where source and target domains are aligned through feature transformations or mappings -whether shallow or deep- prior to classification. Only a few studies have investigated the performance of domain alignment algorithms [60, 17, 50]; however, these works rather focus on specific transformation types, such as linear mappings [60] or location and scale changes [50]. Some literature has investigated the performance and sample complexity of transfer learning via deep learning approaches [19, 35, 27]. However, domain adaptation and transfer learning remain distinct problems: transfer learning deals with differing source and target tasks, unlike domain adaptation. Notably, the characterization of the sample complexity of domain-adaptive neural networks remains an important yet largely unexplored subject in current learning theory. It is well established that the amount of data required to successfully train a neural network increases with the size of the network to prevent overfitting, and many studies have addressed this issue in classical single-domain settings [1, 36, 53, 47, 10]. To the best of our knowledge, however, the scaling of labeled and unlabeled source and target sample requirements with respect to the width and depth of domain-adaptive networks has not been extensively studied yet.

In this work, we aim to fill this gap by providing a comprehensive theoretical analysis of domain adaptation in the widely used setting where the source and target domains are mapped to a common space through feature transformations, and a hypothesis is learnt in that shared space after alignment. We consider a semi-supervised setting where labels are

largely available for the source samples but limited (or unavailable) for the target samples. The structure of the paper along with our main contributions are summarized below:

- In Section 2, we study a general setting that involves learning a source feature transformation $f^s \in \mathcal{F}^s$, a target feature transformation $f^t \in \mathcal{F}^t$ and a hypothesis $h \in \mathcal{H}$ in the common domain. The learning objective minimizes a loss function composed of a weighted (convex) combination of the source and target classification losses, along with a distribution distance term that measures the discrepancy between the aligned domains. At this stage, our analysis remains general and does not assume any specific structure for the learning algorithm. In Section 2.2 (Theorem 2.4), we present a probabilistic bound on the expected target loss in terms of the empirical weighted loss and the expected distribution discrepancy.
- In Section 2.3 we develop these results for the setting where the distribution distance is selected as the popular maximum mean discrepancy (MMD) metric. In Theorem 2.7, we show that the expected target loss can be effectively bounded in terms of the empirical classification and distribution losses alone. This bound holds provided that the number of labeled source samples M_s scales logarithmically with the covering number of the composite hypothesis class $\mathcal{H} \circ \mathcal{F}^s$, while the total number of source and target samples, N_s and N_t, must scale logarithmically with the covering numbers of the feature transformation classes $\mathcal{F}^s$ and $\mathcal{F}^t$.
- In Sections 3.1 and 3.2 we extend our analysis to domain-adaptive deep learning algorithms and, in particular, investigate their sample complexity. We consider two pioneering approaches that have inspired a large body of follow-up work: MMD-based domain adaptation networks [31, 46, 22] and adversarial domain adaptation networks [21, 45, 43]. Our results in Theorems 3.6 and 3.11 show that, in both MMD-based and adversarial domain adaptation settings, the sample complexities for the number of labeled source samples M_s and the total number of source and target samples, N_s and N_t, scale quadratically with the width d and the depth L of the network. Our results also offer insight into the optimal choice for the weight α of the target classification loss, indicating it should decrease at rate $\alpha = O(\sqrt{M_t})$ to effectively handle the scarcity of labeled target samples. Our proof technique extends Theorem 2.7 by thoroughly analyzing the covering numbers of the relevant function classes. To the best of our knowledge, these are the first results to provide a comprehensive characterization of the sample complexity of domain-adaptive neural networks.

We defer a detailed discussion of closely related literature to Section SM2, where we also compare and contrast our results with previous findings. Section 4 presents some simulation results for the experimental validation of our findings, and Section 5 concludes the paper. A preliminary version of our study was presented in [48], which laid the groundwork for the results in Section 2.2.

2. General performance bounds for domain alignment.

2.1. Problem formulation. Let $\mathcal{X}^s$ and $\mathcal{X}^t$ denote two compact metric spaces representing respectively a source domain and a target domain, and let $\mathcal{Y} \subset \mathbb{R}^m$ be a label set. Let μ_s be a source Borel probability measure and μ_t be a target Borel probability measure respectively on the sets $\mathcal{Z}^s = \mathcal{X}^s \times \mathcal{Y}$ and $\mathcal{Z}^t = \mathcal{X}^t \times \mathcal{Y}$. We consider the family of learning algorithms that aim to learn two mappings (transformations) $f^s : \mathcal{X}^s \to \mathcal{X}$ and $f^t : \mathcal{X}^t \to \mathcal{X}$ from the source and target domains to a common set $\mathcal{X}$ together with a hypothesis function $h : \mathcal{X} \to \mathcal{Y}$ estimating class labels on $\mathcal{X}$. The expected losses of the transformations f^s, f^t, and the hypothesis h at the source and target are respectively given by

$$\mathcal{L}^s(f^s, h) = \int_{\mathcal{Z}^s} \ell(h \circ f^s(x^s), \mathbf{y}^s)\, d\mu_s \qquad \mathcal{L}^t(f^t, h) = \int_{\mathcal{Z}^t} \ell(h \circ f^t(x^t), \mathbf{y}^t)\, d\mu_t$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ is a loss function. Assuming that f^s and f^t are measurable mappings, the probability measures μ_s and μ_t on the source and target domains induce corresponding probability measures ν_s and ν_t on the domain $\mathcal{X}$. Let D be a function such that $D(f^s, f^t)$ represents the distance between the measures ν_s and ν_t on $\mathcal{X}$ induced via the mappings f^s and f^t with respect to some distribution discrepancy criterion.

Let $\{x_i^s\}_{i=1}^{N_s}$ be a set of source samples and $\{x_j^t\}_{j=1}^{N_t}$ be a set of target samples drawn independently from the probability measures μ_s and μ_t, where $\{x_i^s\}_{i=1}^{M_s}$ are the M_s labeled samples in the source with labels $\{\mathbf{y}_i^s\}_{i=1}^{M_s}$, and $\{x_j^t\}_{j=1}^{M_t}$ are the M_t labeled samples in the target with labels $\{\mathbf{y}_j^t\}_{j=1}^{M_t}$. We consider learning algorithms that minimize a convex combination of the source and target empirical losses, while minimizing the distance between the transformed source and target samples in the domain $\mathcal{X}$ as

$$\min_{f^s \in \mathcal{F}^s,\ f^t \in \mathcal{F}^t,\ h \in \mathcal{H}} (1-\alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha \hat{\mathcal{L}}^t(f^t, h) + \beta \hat{D}(f^s, f^t). \tag{2.1}$$

Here $\mathcal{F}^s$ and $\mathcal{F}^t$ are function classes consisting of a family of transformations, respectively from the source and target domains $\mathcal{X}^s$ and $\mathcal{X}^t$ to $\mathcal{X}$; $\mathcal{H}$ is a hypothesis class consisting of hypotheses; α is a weight parameter with $0 \le \alpha \le 1$; $\hat{\mathcal{L}}^s(f^s, h)$ and $\hat{\mathcal{L}}^t(f^t, h)$ are the empirical source and target losses given by

$$\hat{\mathcal{L}}^s(f^s, h) = \frac{1}{M_s} \sum_{i=1}^{M_s} \ell(h \circ f^s(x_i^s), \mathbf{y}_i^s) \qquad \hat{\mathcal{L}}^t(f^t, h) = \frac{1}{M_t} \sum_{j=1}^{M_t} \ell(h \circ f^t(x_j^t), \mathbf{y}_j^t) \tag{2.2}$$

and the distance $\hat{D}$ is an estimate of the distribution distance $D(f^s, f^t)$ computed with all (labeled and unlabeled) samples $\{x_i^s\}_{i=1}^{N_s}$ and $\{x_j^t\}_{j=1}^{N_t}$. As discussed in Section 1, the distribution distance $D(f^s, f^t)$ has been chosen in different ways in previous works such as the MMD or Wasserstein distance along with the corresponding estimates $\hat{D}(f^s, f^t)$ that lead to practical learning algorithms. In Section 2.2, we provide generalization bounds for learning algorithms with an arbitrary distribution distance function. Then in Section 2.3, we focus on the kernel mean matching (KMM) methods in particular, and propose bounds for algorithms using a KMM-based distribution distance.

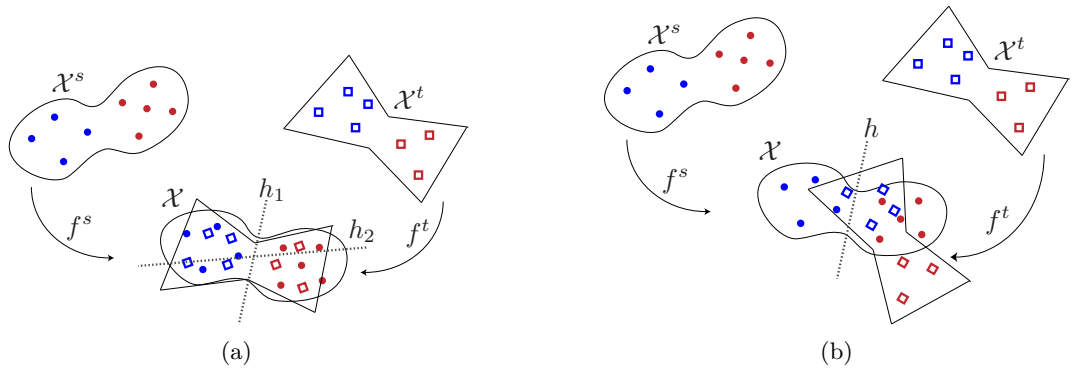


Figure 1. *Illustration of Assumption 2.1. Red and blue colors represent two different classes in the source and target domains $\mathcal{X}^s$ and $\mathcal{X}^t$. In (a), the two domains are well-aligned by the learnt transformations; therefore, the source and target losses are similar. In (b), the learnt transformations do not align the domains well; therefore, the difference between the source and target losses can be high.*

2.2. Generalization bounds for arbitrary distribution distances. In order to analyze the performance of algorithms that aim to solve (2.1), we first assume that the expected loss has a bounded rate of variation with respect to the chosen distribution distance:

Assumption 2.1. *There exists a constant $R > 0$ such that, for any transformations $f^s \in \mathcal{F}^s$, $f^t \in \mathcal{F}^t$ and any hypothesis $h \in \mathcal{H}$, we have*

$$|\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)| \leq R\ D(f^s, f^t). \tag{2.3}$$

Assumption 2.1 imposes the presence of a relation between the source and target distributions: The source and target distributions must be "related" in such a way that, when their distance is reduced in the common domain after going through the transformations in $\mathcal{F}^s$, $\mathcal{F}^t$, their resulting losses should not differ too much compared to the distribution distance $D(f^s, f^t)$. This assumption is illustrated in Figure 1. The figure depicts a simple setting where the source and target domains are aligned by geometric transformations f^s, f^t, which are respectively in the geometric transformation families $\mathcal{F}^s$ and $\mathcal{F}^t$. The hypothesis family $\mathcal{H}$ consists of linear classifiers h. In Figure 1a, the learnt transformations f^s and f^t suitably align the two domains, so that the distribution distance $D(f^s, f^t)$ is small. Consequently, a hypothesis h_1 that yields a small loss $\mathcal{L}^s(f^s, h_1)$ in the source domain also yields a small loss $\mathcal{L}^t(f^t, h_1)$ in the target domain; and a hypothesis h_2 that yields a large loss $\mathcal{L}^s(f^s, h_2)$ in the source domain also yields a large loss $\mathcal{L}^t(f^t, h_2)$ in the target domain. Meanwhile, in Figure 1b the learnt transformations f^s and f^t do not align the two domains well. In this case, the distribution distance $D(f^s, f^t)$ is large, which allows the loss difference $|\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)|$ also to be large by Assumption 2.1. Indeed, one may find a hypothesis h that yields a small loss $\mathcal{L}^s(f^s, h)$ in the source domain, but a large loss $\mathcal{L}^t(f^t, h)$ in the target domain. Since the loss difference $|\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)|$ can be bounded in terms of the distribution distance $D(f^s, f^t)$, the transformation families $\mathcal{F}^s, \mathcal{F}^t$, and the hypothesis family $\mathcal{H}$ considered in this example satisfy Assumption 2.1. In brief, the assumption dictates that there should be a sufficiently strong relation between the source and target domains, the function classes $\mathcal{F}^s$

and $\mathcal{F}^t$ must be chosen suitably to respect this relation, and the hypothesis family $\mathcal{H}$ must also be compatible with the problem. In the following, we first bound the expected target loss in terms of the expected weighted loss and the distribution distance.

We use the above relation to bound the expected target loss in terms of the empirical losses given by the learning algorithm. We characterize the complexity of the transformation and hypothesis classes in terms of their covering numbers, defined as follows [8]:

Definition 2.2. *Let $\mathcal{F}$ be a compact metric space with metric $\mathfrak{d}$, and let $B_\epsilon(f)$ denote an open ball of radius ϵ around $f \in \mathcal{F}$. Then the covering number $\mathcal{N}(\mathcal{F}, \epsilon, \mathfrak{d})$ of $\mathcal{F}$ is defined as*

$$\mathcal{N}(\mathcal{F}, \epsilon, \mathfrak{d}) \triangleq \min\{k : \exists f_1, \dots f_k \in \mathcal{F},\ \mathcal{F} \subset \cup_{i=1}^k B_\epsilon(f_i)\}.$$

In order to study the discrepancy between the expected and the empirical losses, we next make the following assumptions.

Assumption 2.3. *The composite function classes $\mathcal{H} \circ \mathcal{F}^s \triangleq \{g^s = h \circ f^s : h \in \mathcal{H}, f^s \in \mathcal{F}^s\}$ and $\mathcal{H} \circ \mathcal{F}^t \triangleq \{g^t = h \circ f^t : h \in \mathcal{H}, f^t \in \mathcal{F}^t\}$ are compact metric spaces with respect to the metrics*

$$\mathfrak{d}^s(g_1^s, g_2^s) \triangleq \sup_{x^s \in \mathcal{X}^s} \|g_1^s(x^s) - g_2^s(x^s)\| \qquad \mathfrak{d}^t(g_1^t, g_2^t) \triangleq \sup_{x^t \in \mathcal{X}^t} \|g_1^t(x^t) - g_2^t(x^t)\| \tag{2.4}$$

where $\|\cdot\|$ denotes the l_2-norm in $\mathbb{R}^m$. Also, the loss function ℓ is bounded by A_ℓ and Lipschitz continuous with respect to the first argument with constant L_ℓ, such that

$$\begin{aligned} \ell(\mathbf{y}_1, \mathbf{y}_2) &\le A_\ell,\ \forall \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y} \\ |\ell(\mathbf{y}_1, \mathbf{y}) - \ell(\mathbf{y}_2, \mathbf{y})| &\le L_\ell \|\mathbf{y}_1 - \mathbf{y}_2\|,\ \forall \mathbf{y}_1, \mathbf{y}_2, \mathbf{y} \in \mathcal{Y}. \end{aligned}$$

To establish generalization guarantees, we first relate the expected target loss to a weighted combination of source and target losses. By weighting the losses with parameter α, we obtain an upper bound on the expected target loss in terms of the expected weighted loss and the distribution discrepancy (Lemma SM1.1). Next, to connect this to empirical quantities, we bound the deviation between expected and empirical weighted losses using covering number arguments combined with Hoeffding's inequality (Lemma SM1.2). Combining these two results yields the following generalization bound for the expected target loss.

Theorem 2.4. *Let Assumptions 2.1, 2.3 hold. Then for any transformations $f^s \in \mathcal{F}^s$, $f^t \in \mathcal{F}^t$ and hypothesis $h \in \mathcal{H}$, with probability at least*

$$1 - 2\mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t) e^{-\frac{M_t \epsilon^2}{8\alpha^2 A_\ell^2}} - 2\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \frac{\epsilon}{8(1-\alpha) L_\ell}, \mathfrak{d}^s) e^{-\frac{M_s \epsilon^2}{8(1-\alpha)^2 A_\ell^2}} \tag{2.5}$$

the expected target loss is bounded as

$$\mathcal{L}^t(f^t, h) \le \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1-\alpha) RD(f^s, f^t) + \epsilon.$$

The main result in Theorem 2.4 states the following: For any algorithm that computes transformations f^s, f^t, and a hypothesis h by attempting to solve a problem such as in (2.1), the

actual expected loss obtained at the target by applying the learnt transformation f^t and hypothesis h to target test samples cannot differ from the empirical weighted loss $\hat{\mathcal{L}}_\alpha(f^s, f^t, h)$ obtained over training samples by more than ϵ plus an error term involving the distance $D(f^s, f^t)$. This statement holds with probability approaching 1 at an exponential rate with the increase in number of labeled samples M_s. Note that in the very typical case where M_t is limited, the target term in the probability expression (2.5) can be controlled by suitably scaling down the weight parameter α proportionally to $O(\sqrt{M_t})$. [1]

2.3. Generalization bounds for maximum mean discrepancy measures. We now extend the results of Section 2.2 for a setting where the distribution discrepancy in the common domain of transformation is measured with respect to the maximum mean discrepancy (MMD) criterion. The MMD criterion is widely used in domain adaptation. In particular, a popular family of methods called kernel mean matching (KMM) algorithms aim to map the source and target data to a shared domain via a kernel function such that the distance between the source and target samples measured with respect to the MMD criterion is minimized.

KMM methods set the source and target mappings $f^s : \mathcal{X}^s \to \mathcal{X}$ and $f^t : \mathcal{X}^t \to \mathcal{X}$ as a kernel-induced feature map ϕ. The source and target domains $\mathcal{X}^s = \mathcal{X}^t$ are often assumed to be the same and the transformations are set as $f^s = f^t = \phi$. The shared domain $\mathcal{X}$ is typically a Hilbert space with a kernel $k : \mathcal{X}^s \times \mathcal{X}^t \to \mathbb{R}$ satisfying $k(x^s, x^t) = \langle \phi(x^s), \phi(x^t) \rangle_{\mathcal{X}}$ with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ in $\mathcal{X}$.

Given the source and target probability measures μ_s, μ_t on the sets $\mathcal{Z}^s = \mathcal{X}^s \times \mathcal{Y}$ and $\mathcal{Z}^t = \mathcal{X}^t \times \mathcal{Y}$; and the probability measures ν_s, ν_t these respectively induce over the domain $\mathcal{X}$; KMM algorithms characterize the distance between ν_s and ν_t via the MMD given by

$$D(f^s, f^t) = \| E_{x^s}[f^s(x^s)] - E_{x^t}[f^t(x^t)] \|_{\mathcal{X}} \tag{2.6}$$

where $\| \cdot \|_{\mathcal{X}}$ stands for the inner-product-induced norm in the Hilbert space $\mathcal{X}$. [2]Given the source and target sample sets $\{x_i^s\}_{i=1}^{N_s}$ and $\{x_j^t\}_{j=1}^{N_t}$, the empirical estimate of the MMD is given by

$$\hat{D}(f^s, f^t) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} f^s(x_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} f^t(x_j^t) \right\|. \tag{2.7}$$

[1]An important question is how much the learning algorithm is expected to reduce the distribution distance $D(f^s, f^t)$. This depends on the chosen distance; nevertheless, in many practical learning problems, the number of unlabeled samples N_s, N_t is much larger than the number of labeled samples M_s, M_t. If we assume that $N = \min(N_s, N_t)$ is sufficiently large, then we may expect the deviation between the expected and empirical distribution distances to decay such that $P(|D(f^s, f^t) - \hat{D}(f^s, f^t)| \geq \epsilon) \leq (\mathcal{N}_{\mathcal{F}^s,\epsilon} + \mathcal{N}_{\mathcal{F}^t,\epsilon})\, O\left(e^{-N\epsilon^2}\right)$ $\leq O\left(e^{-M_t\epsilon^2}\right) + O\left(e^{-M_s\epsilon^2}\right)$for some appropriate complexity measures $\mathcal{N}_{\mathcal{F}^s,\epsilon}$, $\mathcal{N}_{\mathcal{F}^t,\epsilon}$ for the transformation function classes. In this case, the result in Theorem 2.4 would imply that with probability $1 - O(e^{-M_t\epsilon^2}) - O(e^{-M_s\epsilon^2})$, the expected target loss would be bounded in terms of the empirical losses and the empirical distribution distance as $\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1-\alpha)R\hat{D}(f^s, f^t) + \epsilon + (1-\alpha)R\epsilon$. Our purpose in Section 2.3 is to establish such a result for the particular setting where the distribution distance is chosen as the MMD.

[2]For notational simplicity, we will drop the subscript $(\cdot)_{\mathcal{X}}$ when there is no ambiguity over the space in consideration. The notation $E_{x^s}[\cdot]$ and $E_{x^t}[\cdot]$ indicates that the expectations are taken with respect to the probability measures μ_s and μ_t in the source and the target domains, respectively. We will simply write $E[\cdot]$ whenever the meaning is clear.

In order to study the performance of KMM algorithms, we first derive a bound on the deviation between the actual distribution discrepancy $D(f^s, f^t)$ and its empirical estimate $\hat{D}(f^s, f^t)$.[3] We make the following assumption on the data distributions:

Assumption 2.5. *The random variables $f^s(x_i^s)i = 1^{N_s}$ and $f^t(x_j^t)j = 1^{N_t}$ have bounded expected deviations from their respective means $E[f^s(x^s)]$ and $E[f^t(x^t)]$; that is, there exist constants σ_s^2 and σ_t^2 such that*

$$E\left[\|f^s(x_i^s) - E[f^s(x^s)]\|^2\right] \leq \sigma_s^2 \qquad E\left[\|f^t(x_j^t) - E[f^t(x^t)]\|^2\right] \leq \sigma_t^2. \tag{2.8}$$

Also, for the higher order powers of the deviation, there exist constants C_s and C_t satisfying

$$E\left[\|f^s(x_i^s) - E[f^s(x^s)]\|^k\right] \leq \frac{k!}{2}\sigma_s^2\, C_s^{k-2} \qquad E\left[\|f^t(x_j^t) - E[f^t(x^t)]\|^k\right] \leq \frac{k!}{2}\sigma_t^2\, C_t^{k-2}. \tag{2.9}$$

The condition (2.8) can be seen as a finite variance assumption for a distribution over a Hilbert space, and the condition (2.9) bounds the growth of the k-th central moment by a rate of $O(k!\, C^k)$. These assumptions hold for many common data distributions in practice.

To analyze the MMD estimator, we first establish concentration of sample means in Hilbert spaces. Using a Bernstein-type inequality due to Yurinskii [57], we show that deviations of empirical means from expectations decay exponentially in the sample size (Lemma SM1.3). Building on this, we derive uniform bounds on $|D(f^s, f^t) - \hat{D}(f^s, f^t)|$ that hold simultaneously for all transformation pairs in $\mathcal{F}^s \times \mathcal{F}^t$ by constructing finite covers of these function classes and applying union bounds (Lemma SM1.4). This uniformity requires the following compactness assumption.

Assumption 2.6. *The function classes $\mathcal{F}^s$ and $\mathcal{F}^t$ are compact metric spaces with respect to the metrics*

$$\mathfrak{d}^s_{\mathcal{X}}(f_1^s, f_2^s) \triangleq \sup_{x^s \in \mathcal{X}^s} \|f_1^s(x^s) - f_2^s(x^s)\| \qquad \mathfrak{d}^t_{\mathcal{X}}(f_1^t, f_2^t) \triangleq \sup_{x^t \in \mathcal{X}^t} \|f_1^t(x^t) - f_2^t(x^t)\|. \tag{2.10}$$

Having established concentration properties for the MMD estimator, we can now extend Theorem 2.4 by replacing the expected distribution discrepancy $D(f^s, f^t)$ with its empirical estimate $\hat{D}(f^s, f^t)$, yielding a fully empirical generalization bound.

Theorem 2.7. *Consider a domain adaptation algorithm where the distribution discrepancy is taken as the MMD measure, and the loss function and data distributions satisfy Assumptions 2.1 and 2.6. For $\epsilon > 0$, let the number of source and target samples satisfy $N_s > \frac{16\sigma_s^2}{\epsilon^2}$ and $N_t > \frac{16\sigma_t^2}{\epsilon^2}$. Then for any transformations $f^s \in \mathcal{F}^s$, $f^t \in \mathcal{F}^t$, and hypothesis $h \in \mathcal{H}$, with probability at least*

$$\begin{aligned} &1 - 2\mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t) e^{-\frac{M_t \epsilon^2}{8\alpha^2 A_\ell^2}} - 2\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \frac{\epsilon}{8(1-\alpha)L_\ell}, \mathfrak{d}^s) e^{-\frac{M_s \epsilon^2}{8(1-\alpha)^2 A_\ell^2}} \\ &- \mathcal{N}(\mathcal{F}^s, \frac{\epsilon}{8}, \mathfrak{d}^s_{\mathcal{X}}) e^{-a_s(N_s,\epsilon)} - \mathcal{N}(\mathcal{F}^t, \frac{\epsilon}{8}, \mathfrak{d}^t_{\mathcal{X}}) e^{-a_t(N_t,\epsilon)} \end{aligned} \tag{2.11}$$

[3] Although most KMM methods assume $\mathcal{X}^s = \mathcal{X}^t$ and $f^s = f^t = \phi$, we do not make these assumptions. We only assume that the distribution discrepancy between ν_s and ν_t is taken as in (2.6) and the empirical estimate is computed as in (2.7).

the expected target loss is upper bounded as

$$\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1-\alpha)R\hat{D}(f^s, f^t) + (1-\alpha)R\epsilon + \epsilon. \tag{2.12}$$

The proof follows from Theorem 2.4 and Lemma SM1.4 by the union bound.

The result in Theorem 2.7 states that the target loss can be bounded in terms of the empirical weighted loss and the empirical distribution discrepancy, with probability approaching 1 at an exponential rate as the number of labeled and unlabeled samples increases. The dependence of this rate on the number of unlabeled samples follows from the relations $a_s(N_s, \epsilon) = O(N_s\epsilon^2)$ and $a_t(N_t, \epsilon) = O(N_t\epsilon^2)$. In particular, our result points to the following practical fact: If a domain adaptation algorithm efficiently minimizes the empirical weighted loss and the empirical distribution discrepancy, the true loss obtained in the target domain will also be small, provided that the number of samples is sufficiently high with respect to the complexity of the transformation and hypothesis classes, characterized by their covering numbers.

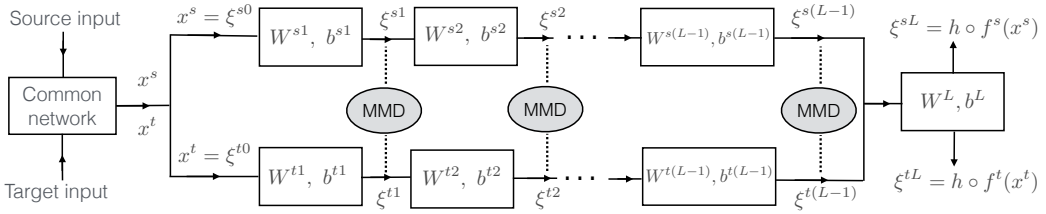


Figure 2. *Illustration of MMD-based domain adaptation networks. Source and target samples first pass through a common network (convolutional and fully connected layers), then through domain-specific networks of $L-1$ fully connected layers, with the L-th layer being a shared classifier. The common network parameters are often adopted from pre-trained networks or fine-tuned using source samples [31, 46, 22]; hence we consider feature representations at its output as our domain samples.*

3. Sample complexity of domain-adaptive neural networks.

In this section, we build on the results in Section 2 and extend our analysis to examine the performance of domain-adaptive neural networks. In particular, we study the sample complexity of two common neural network types, namely, MMD-based and adversarial architectures, respectively in Sections 3.1 and 3.2.

3.1. MMD-based domain adaptation networks.

We study the implications of Theorem 2.7 on deep domain adaptation networks that learn domain-invariant features based on the MMD distance measure. We consider the network model depicted in Figure 2, a commonly adopted foundation for many MMD-based architectures. Defining $\boldsymbol{\xi}^{s0} \triangleq x^s \in \mathbb{R}^{d_0}$ and $\boldsymbol{\xi}^{t0} \triangleq x^t \in \mathbb{R}^{d_0}$, the relation between the features of layers l and $l-1$ is given by

$$\boldsymbol{\xi}^{sl} = \eta^l(\mathbf{W}^{sl}\boldsymbol{\xi}^{s(l-1)} + \mathbf{b}^{sl}) \qquad \boldsymbol{\xi}^{tl} = \eta^l(\mathbf{W}^{tl}\boldsymbol{\xi}^{t(l-1)} + \mathbf{b}^{tl}) \tag{3.1}$$

for $l = 1, \ldots, L$, where $\boldsymbol{\xi}^{sl}, \boldsymbol{\xi}^{tl} \in \mathbb{R}^{d_l}$ are d_l-dimensional source and target features in layer l; the parameters $\mathbf{W}^{sl}, \mathbf{W}^{tl} \in \mathbb{R}^{d_l \times d_{l-1}}$ are source and target weight matrices; the parameters $\mathbf{b}^{sl}, \mathbf{b}^{tl} \in \mathbb{R}^{d_l}$ are source and target bias vectors; $\eta^l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ is a nonlinear activation function; L is the depth of the network; and d_l is the width of the network at layer l. We assume that the parameters of the output layer L are common between the source and the target domains, such that $\mathbf{W}^{sL} = \mathbf{W}^{tL} = \mathbf{W}^L \in \mathbb{R}^{m \times d_{L-1}}$ and $\mathbf{b}^{sL} = \mathbf{b}^{tL} = \mathbf{b}^L \in \mathbb{R}^m$, where $m = d_L$ is the number of classes.

Let $\boldsymbol{\Theta}^{sl} = [\mathbf{W}^{sl}\ \mathbf{b}^{sl}] \in \mathbb{R}^{d_l \times (d_{l-1}+1)}$ and $\boldsymbol{\Theta}^{tl} = [\mathbf{W}^{tl}\ \mathbf{b}^{tl}] \in \mathbb{R}^{d_l \times (d_{l-1}+1)}$ denote the matrices containing the network parameters of layer l. Let us also define the overall parameter structures $\boldsymbol{\Theta}^s = (\boldsymbol{\Theta}^{s1}, \ldots, \boldsymbol{\Theta}^{sL})$ and $\boldsymbol{\Theta}^t = (\boldsymbol{\Theta}^{t1}, \ldots, \boldsymbol{\Theta}^{tL})$ containing the parameters of the entire source and target networks, respectively. We model the source and target domains to be compact sets and the network parameters to be bounded.

Assumption 3.1. *The source and target domains are*

$$\mathcal{X}^s = \{x^s \in \mathbb{R}^{d_0} : \|x^s\| \le A_x\} \qquad \mathcal{X}^t = \{x^t \in \mathbb{R}^{d_0} : \|x^t\| \le A_x\} \tag{3.2}$$

for some bound $A_x > 0$. Also, the network parameters $\boldsymbol{\Theta}^{sl}$, $\boldsymbol{\Theta}^{tl}$ in each layer belong to a closed and bounded set in $\mathbb{R}^{d_l \times (d_{l-1}+1)}$ such that

$$|\boldsymbol{\Theta}^{sl}_{ij}|, |\boldsymbol{\Theta}^{tl}_{ij}| \le A_\Theta \tag{3.3}$$

for some magnitude bound parameter $A_\Theta > 0$*, for* $l = 1, \dots, L$ *and* $i = 1, \dots, d_l$*;* $j = 1, \dots, d_{l-1} + 1$.

Clearly, the features $\boldsymbol{\xi}^{sl}$, $\boldsymbol{\xi}^{tl}$ in all layers depend on both the input vectors x^s, x^t and the network parameters $\boldsymbol{\Theta}^s$, $\boldsymbol{\Theta}^t$. In the following, with a slight abuse of notation we write $\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s}$ when we would like emphasize the dependence of $\boldsymbol{\xi}^{sl}$ on the network parameters $\boldsymbol{\Theta}^s$, and we write $\boldsymbol{\xi}^{sl}(x^s)$ when we would like to refer to the dependence of $\boldsymbol{\xi}^{sl}$ on the input x^s. The notation is set similarly for the target domain variables.

MMD-based deep domain adaptation networks employ a feature mapping $\phi^l : \mathbb{R}^{d_l} \to \mathcal{X}^l$ between the hidden layer feature vectors $\boldsymbol{\xi}^{sl}, \boldsymbol{\xi}^{tl}$ and a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{X}^l$ [31, 25]. The RKHS $\mathcal{X}^l$ of each layer l has a symmetric, positive definite characteristic kernel $k^l : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathbb{R}$ such that $k^l(\boldsymbol{\xi}^l_1, \boldsymbol{\xi}^l_2) = \langle \phi^l(\boldsymbol{\xi}^l_1), \phi^l(\boldsymbol{\xi}^l_2) \rangle_{\mathcal{X}^l}$ for any $\boldsymbol{\xi}^l_1, \boldsymbol{\xi}^l_2 \in \mathbb{R}^{d_l}$, where $\langle \cdot, \cdot \rangle_{\mathcal{X}^l}$ denotes the inner product in the RKHS $\mathcal{X}^l$ [25]. The feature mapping ϕ^l and the characteristic kernel k^l are related as $\phi^l(\boldsymbol{\xi}^l) = k^l(\boldsymbol{\xi}^l, \cdot) : \mathbb{R}^{d_l} \to \mathbb{R}$ [25]. The feature mapping ϕ^l has the property that $\langle \phi^l(\boldsymbol{\xi}^l), \psi \rangle_{\mathcal{X}^l} = \psi(\boldsymbol{\xi}^l)$ for any $\psi \in \mathcal{X}^l$ and $\boldsymbol{\xi}^l \in \mathbb{R}^{d_l}$.

In order to study this common framework within the setting of Section 2.3, let us first define the functions $f^{sl} : \mathcal{X}^s \to \mathcal{X}^l$ and $f^{tl} : \mathcal{X}^t \to \mathcal{X}^l$ as

$$f^{sl}(x^s) \triangleq \phi^l(\boldsymbol{\xi}^{sl}(x^s)) \in \mathcal{X}^l \qquad f^{tl}(x^t) \triangleq \phi^l(\boldsymbol{\xi}^{tl}(x^t)) \in \mathcal{X}^l \tag{3.4}$$

for $l = 1, \dots, L-1$. Note that the direct sum $\mathcal{X} = \bigoplus_{l=1}^{L-1} \mathcal{X}^l = \{(f^1, f^2, \dots, f^{L-1}) : f^l \in \mathcal{X}^l,\ l = 1, \dots, L-1\}$ of the RKHSs $\mathcal{X}^1, \dots, \mathcal{X}^{L-1}$ is also a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ given by [15]

$$\langle (f^1, \dots, f^{L-1}), (g^1, \dots, g^{L-1}) \rangle_{\mathcal{X}} = \sum_{l=1}^{L-1} \langle f^l, g^l \rangle_{\mathcal{X}^l}. \tag{3.5}$$

Let us use the notation $f^{sl}_{\boldsymbol{\Theta}^s}(x^s)$ and $f^{tl}_{\boldsymbol{\Theta}^t}(x^t)$ for the functions $f^{sl}(x^s)$ and $f^{tl}(x^t)$ defined in (3.4) whenever we would like to emphasize their dependence on the network parameters. We can now define the function spaces

$$\begin{aligned} \mathcal{F}^s &= \{f^s : \mathcal{X}^s \to \mathcal{X} \mid f^s(x^s) = \left(f^{s1}_{\boldsymbol{\Theta}^s}(x^s), \dots, f^{s(L-1)}_{\boldsymbol{\Theta}^s}(x^s)\right) \in \mathcal{X},\ |\boldsymbol{\Theta}^{sl}_{ij}| \leq A_\Theta, \forall i, j\} \\ \mathcal{F}^t &= \{f^t : \mathcal{X}^t \to \mathcal{X} \mid f^t(x^t) = \left(f^{t1}_{\boldsymbol{\Theta}^t}(x^t), \dots, f^{t(L-1)}_{\boldsymbol{\Theta}^t}(x^t)\right) \in \mathcal{X},\ |\boldsymbol{\Theta}^{tl}_{ij}| \leq A_\Theta, \forall i, j\} \end{aligned} \tag{3.6}$$

which define the mapping from the source and target domains to the feature representations composed of all layers from $l = 1$ up to $l = L-1$. As these features are passed through layer $l = L$ for the final classification stage, we can regard the network outputs $\boldsymbol{\xi}^{sL}$, $\boldsymbol{\xi}^{tL}$ as the composition of the mappings f^s, f^t with the hypothesis function h, i.e.,

$$g^s(x^s) = (h \circ f^s)(x^s) \triangleq \boldsymbol{\xi}^{sL}(x^s) \qquad g^t(x^t) = (h \circ f^t)(x^t) \triangleq \boldsymbol{\xi}^{tL}(x^t). \tag{3.7}$$

Let us also define the corresponding function spaces

$$\begin{aligned} \mathcal{G}^s &= \mathcal{H} \circ \mathcal{F}^s = \{g^s : \mathcal{X}^s \to \mathcal{Y} \mid g^s(x^s) = \boldsymbol{\xi}^{sL}_{\boldsymbol{\Theta}^s}(x^s) \in \mathcal{Y} \subset \mathbb{R}^m,\ |\boldsymbol{\Theta}^{sl}_{ij}| \leq A_\Theta, \forall i, j\} \\ \mathcal{G}^t &= \mathcal{H} \circ \mathcal{F}^t = \{g^t : \mathcal{X}^t \to \mathcal{Y} \mid g^t(x^t) = \boldsymbol{\xi}^{tL}_{\boldsymbol{\Theta}^t}(x^t) \in \mathcal{Y} \subset \mathbb{R}^m,\ |\boldsymbol{\Theta}^{tl}_{ij}| \leq A_\Theta, \forall i, j\}. \end{aligned} \tag{3.8}$$

In the following, we first assume the continuity of the kernels and the activations.

Assumption 3.2. *The kernels $k^l(\cdot,\cdot)$ for layers $l = 1,\dots,L-1$ and the activation functions $\eta^l(\cdot)$ for layers $l = 1,\dots,L$ are continuous.*

As stated in Lemma SM1.5 and proved in Proof 5, this assumption ensures that $E[f^s(x^s)]$ and $E[f^t(x^t)]$ are in $\mathcal{X}$.

We next revisit the distribution discrepancy definition in Section 2.3 for MMD-based neural networks. Let us define the distribution discrepancy in layer l as $D^l(f^{sl}, f^{tl}) \triangleq \|E_{x^s}[f^{sl}(x^s)] - E_{x^t}[f^{tl}(x^t)]\|_{\mathcal{X}^l}$. MMD-based domain adaptation algorithms typically seek to minimize the empirical estimate $\hat{D}^l$ of D^l at each layer [31, 46, 22]. The empirical distribution discrepancy $\hat{D}^l$ is obtained from the source and target sample sets $\{x_i^s\}_{i=1}^{N_s}$ and $\{x_j^t\}_{j=1}^{N_t}$ as

$$
\begin{aligned}
(\hat{D}^l)^2(f^{sl}, f^{tl}) &= \left\| \frac{1}{N_s}\sum_{i=1}^{N_s} f^{sl}(x_i^s) - \frac{1}{N_t}\sum_{j=1}^{N_t} f^{tl}(x_j^t) \right\|_{\mathcal{X}^l}^2 \\
&= \frac{1}{N_s^2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} k^l(\boldsymbol{\xi}_i^{sl}, \boldsymbol{\xi}_j^{sl}) - \frac{2}{N_s N_t}\sum_{i=1}^{N_s}\sum_{j=1}^{N_t} k^l(\boldsymbol{\xi}_i^{sl}, \boldsymbol{\xi}_j^{tl}) + \frac{1}{N_t^2}\sum_{i=1}^{N_t}\sum_{j=1}^{N_t} k^l(\boldsymbol{\xi}_i^{tl}, \boldsymbol{\xi}_j^{tl})
\end{aligned}
\tag{3.9}
$$

where $\boldsymbol{\xi}_i^{sl}$ and $\boldsymbol{\xi}_j^{tl}$ denote the source and target features in layer l corresponding respectively to the samples x_i^s and x_j^t. The second equality follows from the relations $f^{sl}(x_i^s) = \phi^l(\boldsymbol{\xi}_i^{sl})$ and $f^{tl}(x_j^t) = \phi^l(\boldsymbol{\xi}_j^{tl})$. The overall distribution discrepancy between the source and the target domains defined in (2.6) is given by

$$
D(f^s, f^t) = \|E_{x^s}[f^s(x^s)] - E_{x^t}[f^t(x^t)]\|_{\mathcal{X}}
$$

following the definitions in Lemma SM1.5. Its empirical estimate $\hat{D}(f^s, f^t)$ defined in (2.7) is then obtained as

$$
\begin{aligned}
\hat{D}^2(f^s, f^t) &= \left\| \frac{1}{N_s}\sum_{i=1}^{N_s} f^s(x_i^s) - \frac{1}{N_t}\sum_{j=1}^{N_t} f^t(x_j^t) \right\|_{\mathcal{X}}^2 \\
&= \frac{1}{N_s^2}\sum_{i,j=1}^{N_s} \langle f^s(x_i^s), f^s(x_j^s)\rangle_{\mathcal{X}} - \frac{2}{N_s N_t}\sum_{i=1}^{N_s}\sum_{j=1}^{N_t} \langle f^s(x_i^s), f^t(x_j^t)\rangle_{\mathcal{X}} \\
&\quad + \frac{1}{N_t^2}\sum_{i,j=1}^{N_t} \langle f^t(x_i^t), f^t(x_j^t)\rangle_{\mathcal{X}} \\
&= \sum_{l=1}^{L-1} (\hat{D}^l)^2(f^{sl}, f^{tl}).
\end{aligned}
\tag{3.10}
$$

where the last equality follows from the definition (3.5) of the inner product in $\mathcal{X}$.

Most MMD-based deep domain adaptation networks rely on aligning the source and the target domains by minimizing the total MMD distance (3.10) summed over all layers [49, 31, 46, 22]. We thus consider a learning algorithm that minimizes the overall loss

$$
\min_{f^s\in\mathcal{F}^s,\ f^t\in\mathcal{F}^t,\ h\in\mathcal{H}} (1-\alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha\hat{\mathcal{L}}^t(f^t, h) + \beta\sum_{l=1}^{L-1}(\hat{D}^l)^2(f^{sl}, f^{tl}).
\tag{3.11}
$$

Hence, the above analysis provides the bridge between the results in Section 2.3 and the current setting with MMD-based domain adaptation networks, so that the statement of Theorem 2.7 applies to the current problem. Before we proceed with the implications of Theorem 2.7, we need two additional assumptions.

Assumption 3.3. *The symmetric kernel $k^l(\cdot,\cdot) : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathbb{R}$ is Lipschitz continuous with constant L_K in each argument, such that*

$$|k^l(\boldsymbol{\xi}_1, \boldsymbol{\xi}) - k^l(\boldsymbol{\xi}_2, \boldsymbol{\xi})| \leq L_K \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\| \tag{3.12}$$

for all $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi} \in \mathbb{R}^{d_l}$. Also, the nonlinear activation functions η^l in (3.1) are Lipschitz-continuous with constant L_η, such that

$$\|\eta^l(\mathbf{u}) - \eta^l(\mathbf{v})\| \leq L_\eta \, \|\mathbf{u} - \mathbf{v}\| \tag{3.13}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_l}$, for $l = 1, \ldots, L$.

Assumption 3.4. *The nonlinear activation functions η^l in (3.1) are bounded either in value (e.g., sigmoid, softmax) or as an operator (e.g., ReLU). In the former case, we assume that there exists a constant $C_\eta > 0$ with*

$$|\eta_i^l(\mathbf{u})| \leq C_\eta \tag{3.14}$$

for all $\mathbf{u} \in \mathbb{R}^{d_l}$, for $l = 1, \ldots, L-1$ and $i = 1, \ldots, d_l$, where $\eta_i^l(\mathbf{u})$ denotes the i-th component of $\eta^l(\mathbf{u})$. In the latter case, we assume that there exists $A_\eta > 0$ such that

$$\|\eta^l(\mathbf{u})\| \leq A_\eta \|\mathbf{u}\| \tag{3.15}$$

for all $\mathbf{u} \in \mathbb{R}^{d_l}$, for $l = 1, \ldots, L-1$.

The Lipschitz continuity condition (3.12) holds for many widely used kernels such as Gaussian kernels. As for condition (3.13), the Lipschitz constants of the commonly used rectified linear unit, softmax and softplus activation functions are derived in Section SM4. Under these assumptions, the transformation function classes $\mathcal{F}^s, \mathcal{F}^t$ and the composite function classes $\mathcal{G}^s$, $\mathcal{G}^t$ are compact metric spaces with respect to the metrics defined in (2.10) and (2.4), respectively. This compactness result (Lemma SM1.6) is established by showing that the bounded parameter space is compact and the mapping from parameters to functions is continuous. Having established compactness, we can now characterize the covering numbers of these function classes.

To upper bound the covering numbers, we construct finite covers by discretizing the network parameter space into regular grids and leveraging the Lipschitz continuity of network components to control the induced function space distances. This analysis yields explicit covering number bounds in terms of network depth, width, and the problem-dependent constants (Lemmas SM1.7 and SM1.8). From these technical results, we obtain the following characterization of the growth rates of covering numbers with network depth and width.

Corollary 3.5. *Consider that the feature dimensions d_l are such that $d_l = O(d)$ for $l = 1, \ldots, L$, for some common network width parameter d. Then, the rate of growth of the*

covering numbers for the function spaces $\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}^s_{\mathcal{X}})$, $\mathcal{N}(\mathcal{F}^t, \epsilon, \mathfrak{d}^t_{\mathcal{X}})$, $\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s)$, $\mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \epsilon, \mathfrak{d}^t)$ *with the width* d *and the depth* L *of the network is upper bounded by*

$$O\left(\left(\frac{L}{\epsilon}\right)^{d^2 L} (cd)^{d^2 L^2}\right)$$

where c *denotes a constant.*

Corollary 3.5 is proved in Section SM5 of the supplement. Combining Corollary 3.5 and Theorem 2.7, we are now ready to state our main result about the sample complexity of MMD-based domain adaptation networks in Theorem 3.6 below, whose proof is presented in Section SM6 of the supplement.

Theorem 3.6. *Consider a learning algorithm relying on the minimization of a loss function of the form* (3.11) *via an MMD-based domain adaptation network. Assume that the classification loss function* ℓ *is bounded by a constant* A_ℓ *and Lipschitz continuous with respect to the first argument with constant* L_ℓ*. Suppose that the source and target data distributions satisfy Assumptions 2.1 and 2.5. Assume also that the network parameters, activation functions and the kernels satisfy Assumptions 3.1-3.4. Consider that the weight parameter* α *in the loss function is chosen such that*

$$\alpha = O\left(\left(\frac{M_t \epsilon^2}{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}\right)^{1/2}\right)$$

according to the number M_t *of available labeled target samples. Then in order to bound the expected target loss with a generalization gap of* $O(\epsilon)$ *as*

$$\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1-\alpha) R\hat{D}(f^s, f^t) + (1-\alpha)R\epsilon + \epsilon, \tag{3.16}$$

the sample complexities in terms of the number M_s *of labeled source samples, the number* N_s *of all (labeled and unlabeled) source samples, and the number* N_t *of all target samples are upper bounded by*

$$O\left(\frac{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right). \tag{3.17}$$

Theorem 3.6 shows that sample complexities M_s, N_s, and N_t must increase at rate $O(d^2 L^2)$ as network depth L and width d increase (ignoring logarithmic terms), indicating quadratic growth with network size to prevent overfitting.[4] For limited labeled target samples M_t, the weight α must shrink at rate $\alpha = O(\sqrt{M_t})$ to avoid overfitting, and similarly at rate $\alpha = O((dL)^{-1})$ as network size grows. Sample sizes scale as $O(\epsilon^{-2})$ for an $O(\epsilon)$ bound on loss difference.

[4]The assumption of the existence of constants A_ℓ and L_ℓ is satisfied in many settings; we derive these for cross-entropy loss in Appendix SM7.

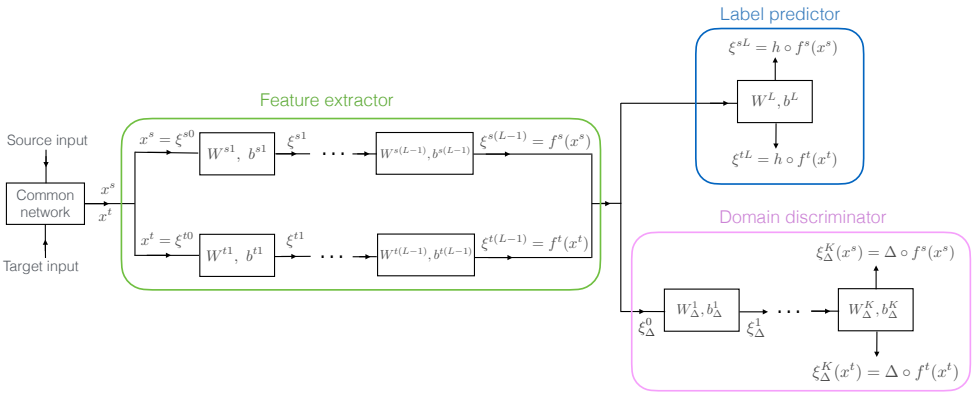


Figure 3. *Illustration of adversarial domain adaptation networks*

3.2. Adversarial domain adaptation networks. In this section, we extend our results to analyze the sample complexity of adversarial domain adaptation networks. Domain-adversarial neural networks aim to compute domain-invariant representations $f^s : \mathcal{X}^s \to \mathcal{X}$, $f^t : \mathcal{X}^t \to \mathcal{X}$ through a feature extractor network, followed by a label predictor $h : \mathcal{X} \to \mathcal{Y}$ (Figure 3). The domain-invariance of the learnt features is ensured by a domain discriminator network trained to distinguish source from target features. The feature extractor and discriminator are trained adversarially: the extractor learns representations indistinguishable to the discriminator. The domain discriminator $\Delta : \mathcal{X} \to \mathbb{R}$ minimizes the domain discrimination loss $\mathcal{L}^s_{\mathcal{D}}(f^s, \Delta) + \mathcal{L}^t_{\mathcal{D}}(f^t, \Delta)$ where $\mathcal{L}^s_{\mathcal{D}}(f^s, \Delta) = E[\ell_{\mathcal{D}}(\Delta \circ f^s(x^s), l^s)]$, and $\mathcal{L}^t_{\mathcal{D}}(f^t, \Delta) = E[\ell_{\mathcal{D}}(\Delta \circ f^t(x^t), l^t)]$ respectively denote the expected domain discrimination losses in the source and the target domains; $\ell_{\mathcal{D}} : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ is a domain discrimination loss function; and $l^s, l^t \in \mathbb{R}$ denote the domain labels of the source and the target domains. It is common practice to set the domain discrimination loss $\ell_{\mathcal{D}}$ as a logarithmic penalty on the deviation between the estimated domain labels and the true domain labels $l^s = 0$, $l^t = 1$ as [21, 45, 32]:

$$\begin{aligned} \ell_{\mathcal{D}}(\Delta \circ f^s(x^s), l^s) &= -\log(1 - \Delta \circ f^s(x^s)) \\ \ell_{\mathcal{D}}(\Delta \circ f^t(x^t), l^t) &= -\log(\Delta \circ f^t(x^t)). \end{aligned} \tag{3.18}$$

Meanwhile, the feature extractor network is trained to maximize the domain classification loss so that the learnt features are domain-invariant, leading to the overall optimization problem

$$\min_{f^s, f^t, h, \Delta} (1 - \alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha\hat{\mathcal{L}}^t(f^t, h) - \beta\big(\hat{\mathcal{L}}^s_{\mathcal{D}}(f^s, \Delta) + \hat{\mathcal{L}}^t_{\mathcal{D}}(f^t, \Delta)\big) \tag{3.19}$$

where $\hat{\mathcal{L}}^s, \hat{\mathcal{L}}^t$ denote the empirical source and target classification losses defined in (2.2). Here $\hat{\mathcal{L}}^s_{\mathcal{D}}, \hat{\mathcal{L}}^t_{\mathcal{D}}$ are the empirical domain discrimination losses given by

$$\hat{\mathcal{L}}^s_{\mathcal{D}}(f^s, \Delta) = \frac{1}{N_s}\sum_{i=1}^{N_s} \ell_{\mathcal{D}}(\Delta \circ f^s(x^s_i), l^s_i) \qquad \hat{\mathcal{L}}^t_{\mathcal{D}}(f^t, \Delta) = \frac{1}{N_t}\sum_{j=1}^{N_t} \ell_{\mathcal{D}}(\Delta \circ f^t(x^t_j), l^t_j).$$

where l_i^s and l_j^t respectively denote the domain labels of the source samples x_i^s and the target samples x_j^t.

In order to study domain-adversarial network models within our framework, we consider that the transformations f^s, f^t are given by the feature representations at layer $L-1$ of the feature extractor network. The corresponding function spaces are then

$$\begin{aligned}\mathcal{F}^s &= \{f^s : \mathcal{X}^s \to \mathbb{R}^{d_{L-1}} \,|\, f^s(x^s) = \boldsymbol{\xi}_{\boldsymbol{\Theta}^s}^{s(L-1)}(x^s),\ |\boldsymbol{\Theta}_{ij}^{sl}| \leq A_\Theta, \forall i,j\} \\ \mathcal{F}^t &= \{f^t : \mathcal{X}^t \to \mathbb{R}^{d_{L-1}} \,|\, f^t(x^t) = \boldsymbol{\xi}_{\boldsymbol{\Theta}^t}^{t(L-1)}(x^t),\ |\boldsymbol{\Theta}_{ij}^{tl}| \leq A_\Theta, \forall i,j\}.\end{aligned} \tag{3.20}$$

Similarly, the hypotheses $h\circ f^s$ and $h\circ f^t$ are given by the output of the last layer L $h\circ f^s(x^s) = \boldsymbol{\xi}^{sL}(x^s)$, and $h\circ f^t(x^t) = \boldsymbol{\xi}^{tL}(x^t)$ with the function spaces $\mathcal{H}\circ\mathcal{F}^s$ and $\mathcal{H}\circ\mathcal{F}^t$ defined[5] in (3.8). Here, the features between layers $l-1$ and l are related as in (3.1) through the network parameters $\mathbf{W}^{sl}, \mathbf{W}^{tl}, \mathbf{b}^{sl}, \mathbf{b}^{tl}$ and the nonlinear activation functions η^l. [6]

The domain discriminator network typically consists of several fully connected layers [21, 45]. Denoting the weight parameters of these layers as $\mathbf{W}_\Delta^l \in \mathbb{R}^{d_l^\Delta \times d_{l-1}^\Delta}$, $\mathbf{b}_\Delta^l \in \mathbb{R}^{d_l^\Delta}$, the relation between the responses $\boldsymbol{\xi}_\Delta^{l-1} \in \mathbb{R}^{d_{l-1}^\Delta}, \boldsymbol{\xi}_\Delta^l \in \mathbb{R}^{d_l^\Delta}$ at layers $l-1$ and l is given by $\boldsymbol{\xi}_\Delta^l = \eta_\Delta^l(\mathbf{W}_\Delta^l \boldsymbol{\xi}_\Delta^{l-1} + \mathbf{b}_\Delta^l)$ for $l = 1, \ldots, K$, where K denotes the number of layers and $\eta_\Delta^l : \mathbb{R}^{d_l^\Delta} \to \mathbb{R}^{d_l^\Delta}$ denotes the activation function of the domain discriminator network at layer l. Here, the input $\boldsymbol{\xi}_\Delta^0$ to the domain discriminator network corresponds to the outputs $\boldsymbol{\xi}^{s(L-1)}, \boldsymbol{\xi}^{t(L-1)}$ of the feature extractor networks. The domain discriminator output is then given by $\Delta \circ f^s(x^s) = \boldsymbol{\xi}_\Delta^K(x^s)$, and $\Delta \circ f^t(x^t) = \boldsymbol{\xi}_\Delta^K(x^t)$ for the source and the target domains, where the dimension of the output layer of the domain discriminator is $d_K^\Delta = 1$. Still using Assumption 3.1 and extending it to the domain discriminator network as well, we define the function class of domain discriminators with bounded network weights as

$$\mathcal{D} = \{\Delta : \mathbb{R}^{d_{L-1}} \to \mathbb{R} \mid \Delta(\boldsymbol{\xi}_\Delta^0) = \boldsymbol{\xi}_\Delta^K,\ |(\mathbf{W}_\Delta^l)_{ij}| \leq A_\Theta,\ |(\mathbf{b}_\Delta^l)_i| \leq A_\Theta, \forall i,j\}. \tag{3.21}$$

Provided that the adversarial domain adaptation network is well-trained, the mappings $f^s(x^s)$, $f^t(x^t)$ specialize in the extraction of domain-invariant features such that the domain discriminator cannot distinguish between the source and the target samples. The discriminator outputs $\Delta \circ f^s(x^s)$ and $\Delta \circ f^t(x^t)$ then take similar values. Based on this observation, we build our analysis on the following definition of the distribution distance

$$D_\Delta(f^s, f^t) \triangleq \left| E[\Delta \circ f^s(x^s)] - E[\Delta \circ f^t(x^t)] \right|.$$

[5] Note that, the definitions of the function spaces $\mathcal{F}^s, \mathcal{F}^t$ in this section are different from those in Section 3.1, as they take different roles between MMD-based and adversarial networks. Nevertheless, the composite function spaces $\mathcal{G}^s = \mathcal{H}\circ\mathcal{F}^s$ and $\mathcal{G}^t = \mathcal{H}\circ\mathcal{F}^t$ in this section are the same as those of Section 3.1, since the functions g^s, g^t are defined through the classification layer output in both the MMD-based and the adversarial settings.

[6] Buraya bir ayar verilecek. While feature extractor networks typically consist of several convolutional layers followed by fully connected layers in many common architectures [39]; in domain adaptation applications it is a common strategy to adopt convolutional layer weights from pretrained networks or to train or fine-tune them using only source data [45]. Therefore, we leave the training of convolutional layers out of the scope of our analysis. We consider the input source and target samples $x^s, x^t \in \mathbb{R}^{d_0}$ to be the response generated at the output of the convolutional network common between the two domains as illustrated in Figure 3 and focus on the action of the fully connected layers of the feature extractor networks.

The distribution distance $D_\Delta(f^s, f^t)$ measures how well the source and target distributions are aligned once they are mapped to the shared feature space by the mappings f^s and f^t. Note that the above definition of the distribution distance $D_\Delta(f^s, f^t)$ depends also on the domain discriminator Δ. We make the following assumption about the domain discriminator.

Assumption 3.7. *The domain discriminator output is bounded, i.e., there exists a constant $C_\mathcal{D} > 0$ such that $|\Delta(\boldsymbol{\xi}^0_\Delta)| = |\boldsymbol{\xi}^K_\Delta| \leq C_\mathcal{D}$ for all $\boldsymbol{\xi}^0_\Delta \in \mathbb{R}^{d_{L-1}}$.*

Note that Assumption 3.7 is satisfied for many domain-adversarial networks, as the activation function η^K_Δ of the final domain discriminator layer is often selected as a bounded function such as the sigmoid [21] or the softmax function [44]. Let us denote the composition of the domain discriminator and the feature extractor as $v^s(x^s) \triangleq \Delta \circ f^s(x^s)\, v^t(x^t) \triangleq \Delta \circ f^t(x^t)$, and the corresponding function spaces as $\mathcal{V}^s = \mathcal{D} \circ \mathcal{F}^s = \{v^s : v^s = \Delta \circ f^s, \Delta \in \mathcal{D}, f^s \in \mathcal{F}^s\}$, and $\mathcal{V}^t = \mathcal{D} \circ \mathcal{F}^t = \{v^t : v^t = \Delta \circ f^t, \Delta \in \mathcal{D}, f^t \in \mathcal{F}^t\}$.

In order to study the sample complexity of adversarial domain adaptation networks, we characterize the deviation between the expected distribution distance $D_\Delta(f^s, f^t)$ and its finite-sample estimate

$$\hat{D}_\Delta(f^s, f^t) = \left| \frac{1}{N_s} \sum_{i=1}^{N_s} \Delta \circ f^s(x^s_i) - \frac{1}{N_t} \sum_{j=1}^{N_t} \Delta \circ f^t(x^t_j) \right|$$

in Lemma SM1.9, which is the counterpart of Lemma SM1.4 in the domain-adversarial setting. Before stating the main result of this section, we formalize the following conditions.

Assumption 3.8. *The activation functions $\eta^l(\cdot)$ for layers $l = 1, \ldots, L$ and the activation functions $\eta^l_\Delta(\cdot)$ for layers $l = 1, \ldots, K$ are continuous and also Lipschitz-continuous with constant L_η, such that $\|\eta^l(\mathbf{u}) - \eta^l(\mathbf{v})\| \leq L_\eta \|\mathbf{u} - \mathbf{v}\|$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_l}$, for $l = 1, \ldots, L$ and $\|\eta^l_\Delta(\mathbf{u}) - \eta^l_\Delta(\mathbf{v})\| \leq L_\eta \|\mathbf{u} - \mathbf{v}\|$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d^\Delta_l}$, for $l = 1, \ldots, K$.*

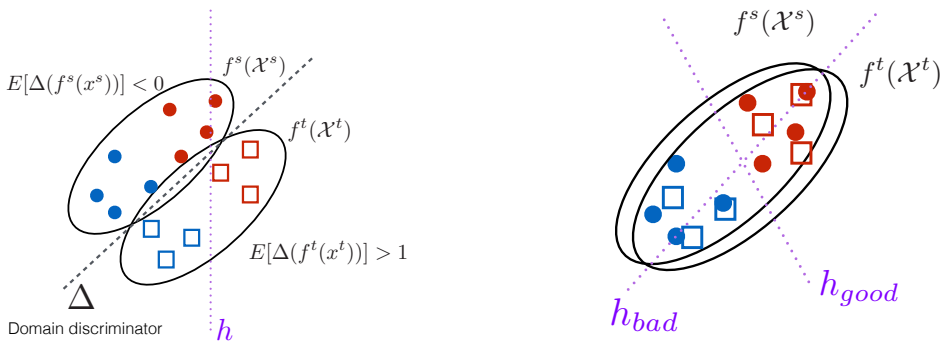
Assumption 3.9. *The nonlinear activation functions η^l_Δ are bounded either in value or as an operator, for $l = 1, \ldots, K-1$. In the former case, there exists a constant $C_\eta > 0$ with $|(\eta^l_\Delta)_i(\mathbf{u})| \leq C_\eta$ for all $\mathbf{u} \in \mathbb{R}^{d^\Delta_l}$, where $(\eta^l_\Delta)_i(\mathbf{u})$ denotes the i-th component of $\eta^l_\Delta(\mathbf{u})$. In the latter case, there exists $A_\eta > 0$ such that $\|\eta^l_\Delta(\mathbf{u})\| \leq A_\eta \|\mathbf{u}\|$ for all $\mathbf{u} \in \mathbb{R}^{d^\Delta_l}$.*

Note that Assumption 3.8 is an adaptation of the conditions in Assumptions 3.2 and 3.3 to the domain-adversarial setting in consideration. Similarly, Assumption 3.9 simply adapts the condition in Assumption 3.4 to the domain discriminator network. We lastly make the following assumption about the link between the distribution distance and the deviation between the source and target losses.

Assumption 3.10. *There exists a constant $R_A > 0$ such that, for the domain discriminator $\Delta \in \mathcal{D}$ learnt by the algorithm, we have*

$$|\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)| \leq R_A\, D_\Delta(f^s, f^t)$$

for any transformations $f^s \in \mathcal{F}^s$, $f^t \in \mathcal{F}^t$, and any hypothesis $h \in \mathcal{H}$.



(a) Poor alignment (b) Proper alignment

Figure 4. *Illustration of Assumption 3.10. Red and blue colors represent two different classes in the source and target domains. In (a), the two domains are poorly aligned by the mappings f^s and f^t, therefore, the algorithm learns a domain discriminator Δ that can separate the two domains well. The domain distance $D_\Delta(f^s, f^t)$ is then high, and consequently, there may exist hypotheses h yielding a small loss in one domain and a large loss in the other domain. In (b), the domains are well-aligned and the domain distance $D_\Delta(f^s, f^t)$ is small. The source and target losses are then similar for any hypothesis h.*

Assumption 3.10 is the counterpart of Assumption 2.1 in the context of adversarial domain adaptation networks, which is illustrated in Figure 4. The assumption asserts that the source and the target distributions be related in such a way that, when efficiently aligned via the feature mappings f^s and f^t so as to minimize the domain discrepancy $D_\Delta(f^s, f^t)$, the classification losses arising in the source and the target domains are also comparable. [7]

We can now state our main result about the sample complexity of adversarial domain adaptation networks.

Theorem 3.11. *Consider a learning algorithm relying on the minimization of a loss function of the form (3.19) via an adversarial domain adaptation network. Assume that the classification loss function ℓ is bounded by a constant A_ℓ and Lipschitz continuous with respect to the first argument with constant L_ℓ. Suppose that the source and target data distributions satisfy Assumption 3.10 and the network parameters and activation functions satisfy Assumption 3.1 and Assumptions 3.4 to 3.9.*

Let the feature dimensions be such that $d_l = O(d)$ for $l = 1, \dots, L$ and $d_l^\Delta = O(d)$ for $l = 1, \dots, K$ for some common width parameter d. Consider that the weight parameter α in the loss function is chosen such that

$$\alpha = O\left(\left(\frac{M_t \epsilon^2}{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}\right)^{1/2}\right) \tag{3.22}$$

[7] Note that the assumption is not limited to the ideal scenario where the domains are well-aligned: In case of poor alignment, $D_\Delta(f^s, f^t)$ may be high, possibly leading to significantly different losses in the two domains. We, however, assume that the domain discriminator network is sufficiently well-trained; i.e., the learnt discriminator Δ is able to distinguish between the source and target domains if the mappings f^s and f^t result in poor feature alignment.

according to the number M_t of available labeled target samples. Then, in order to bound the expected target loss with a generalization gap of $O(\epsilon)$ as

$$\mathcal{L}^t(f^t,h) \le \hat{\mathcal{L}}_\alpha(f^s,f^t,h) + (1-\alpha)R_A\hat{D}_\Delta(f^s,f^t) + (1-\alpha)R_A\epsilon + \epsilon, \tag{3.23}$$

the sample complexities in terms of the number M_s of labeled source samples, the number N_s of all (labeled and unlabeled) source samples, and the number N_t of all target samples are upper bounded by

$$\begin{aligned} M_s &= O\left(\frac{d^2L\log\left(\frac{L}{\epsilon}\right) + d^2L^2\log(d)}{\epsilon^2}\right) \\ N_s, N_t &= O\left(\frac{d^2(L+K)\log\left(\frac{L+K}{\epsilon}\right) + d^2(L+K)^2\log(d)}{\epsilon^2}\right). \end{aligned} \tag{3.24}$$

The proof of Theorem 3.11 is presented in Section SM8. The findings of Theorem 3.11 on the sample complexity of domain-adversarial networks are in line with those of Theorem 3.6, which studied MMD-based networks. The optimal choice for the weight parameter α scales as $O(\sqrt{M_t})$ as the number of labeled target samples varies. In order to prevent overfitting, M_s must increase at rate $M_s = O(d^2L^2)$ with d and L, which indicates that the number of labeled source samples must increase quadratically with the width d and the depth L of the feature extractor network, ignoring the logarithmic factors. Likewise, the number of source and target samples N_s and N_t must also increase at a quadratic rate $O(d^2(L+K)^2)$ with the width d and the depth $L+K$ of the combination of feature extractor and domain discriminator networks, in order to avoid overfitting to the empirical domain discrimination loss of training samples. Similarly to the result in Theorem 3.6, for the difference between the expected target loss and the sum of the empirical losses to be bounded by an amount of $O(\epsilon)$, the number of samples M_s, N_s, N_t must scale at rate $O(\epsilon^{-2})$. While we analyze single-layer label predictors as in Figure 3 following common practice, our results extend to multi-layer predictors with depth P by replacing L with $L+P$ in the complexity bounds. The optimal choice of the weight parameter α in (3.22) can similarly be obtained by replacing the number of layers L with $L+P$ in this case.[8] [9]

[8] This is due to the fact that our analysis is based on the covering numbers of the function spaces $\mathcal{G}^s, \mathcal{G}^t$ and $\mathcal{V}^s, \mathcal{V}^t$, where $\mathcal{N}(\mathcal{G}^s,\epsilon,\mathfrak{d}^s)$, $\mathcal{N}(\mathcal{G}^t,\epsilon,\mathfrak{d}^t)$ depend on only the total number of layers in the cascade of the feature extractor and the label predictor networks, and $\mathcal{N}(\mathcal{V}^s,\epsilon,\mathfrak{d}^s_\mathcal{V})$, $\mathcal{N}(\mathcal{V}^t,\epsilon,\mathfrak{d}^t_\mathcal{V})$ depend only on the total number of layers in the cascade of the feature extractor and the domain discriminator networks.

[9] In our analysis, we have considered the label predictor network to consist of a single layer as illustrated in Figure 3, as common practice in adversarial domain adaptation networks. Nevertheless, it is straightforward to adapt our results to the case where the label predictor network consists of more than one layer. This is due to the fact that our analysis is based on the covering numbers of the function spaces $\mathcal{G}^s, \mathcal{G}^t$ and $\mathcal{V}^s, \mathcal{V}^t$, where $\mathcal{N}(\mathcal{G}^s,\epsilon,\mathfrak{d}^s)$, $\mathcal{N}(\mathcal{G}^t,\epsilon,\mathfrak{d}^t)$ depend on only the total number of layers in the cascade of the feature extractor and the label predictor networks, and $\mathcal{N}(\mathcal{V}^s,\epsilon,\mathfrak{d}^s_\mathcal{V})$, $\mathcal{N}(\mathcal{V}^t,\epsilon,\mathfrak{d}^t_\mathcal{V})$ depend only on the total number of layers in the cascade of the feature extractor and the domain discriminator networks. Denoting the depth of the label predictor network as P in this alternative setting, the resulting sample complexities would be obtained as $M_s = O(d^2(L+P)^2)$, and $N_s, N_t = O(d^2(L+K)^2)$. The optimal choice of the weight parameter α in (3.22) can similarly be obtained by replacing the number of layers L with $L+P$ in this case.

4. Experimental results. In this section, we present experimental results for the verification of the proposed generalization bounds. We first study the generic bounds using shallow classifier models, then examine the sample complexity of domain-adaptive neural networks. Complete experimental details are provided in Section SM3 of the supplement.

4.1. General domain alignment methods. We validate our findings on a synthetic data set with two classes.[10]

Figure 5(a) shows the variation of the target misclassification rate with the number M_t of labeled target samples for different values of α. The decay in the target error is consistent with the rate $O(\sqrt{1/M_t})$ predicted by Theorem 2.4, as confirmed by the fitted theoretical curves. We also observe that large M_t values favor larger α, while smaller M_t values require smaller α, supporting the scaling $\alpha = O(\sqrt{M_t})$. Figure 5(b) illustrates that the misclassification rate increases approximately linearly with the transformation estimation error τ, which is proportional to $D(f^s, f^t)$, confirming the prediction of Theorem 2.4 that the target loss increases proportionally to the distribution distance.

We next experiment on the MIT-CBCL face data set [34] (Figure 6).[11] Figure 7 shows that misclassification rates are effectively reduced with increasing labeled samples, at rates consistent with $O(\sqrt{1/M_t})$ and $O(\sqrt{1/M_s})$, as predicted by our theory. The fitted theoretical curves closely match the experimental data.

4.2. Domain-adaptive neural networks. We experimentally verify the sample complexity results of Theorems 3.6 and 3.11 using MNIST $\rightarrow$ MNIST-M experiments [30, 20].[12]

4.2.1. MMD-based domain adaptation networks. We adopt the MMD-based architecture from [31], building on our previous experimental study [28].[13] Figure 8 presents the sample complexity with respect to network depth: the left panels show the decrease in target accuracy as L increases for different M_s and N_s values, while the right panels plot the

[10] Source and target data are generated by applying different geometric transformations to 400 samples drawn from the standard normal distribution in $\mathbb{R}^2$. We emulate a setting where the transformations f^s and f^t are learnt with some estimation error τ. The classifier is a regularized ridge regression trained in the common domain. Target misclassification rates are evaluated over 1000 test samples. Complete setup is provided in Section SM3.1.

[11] The dataset contains 3240 synthetic face images of 10 subjects rendered under 36 illumination conditions and 9 poses. We use Pose 1 (frontal) as source and Poses 2, 5, 9 as targets in separate trials. Domain alignment uses the PCA-based method of [18] and an SVM classifier. Complete setup in Section SM3.1.

[12] MNIST (60000 images) serves as source and MNIST-M (59000 colored background images) as target. Networks are trained with varying numbers of labeled and unlabeled samples. Hyperparameters are chosen to keep the network in the overfitting regime, where target accuracy decreases as network complexity grows for fixed sample sizes, enabling the characterization of sample complexity. Target accuracy vs. network complexity curves are obtained via linear extrapolation; sample complexity is characterized by identifying the minimum sample sizes required to maintain reference accuracy levels as complexity increases. Complete details in Section SM3.2.

[13] The network begins with convolutional layers followed by fully connected MMD layers with coupled parameters between domains, and is trained with batch normalization after each layer. The network minimizes cross-entropy loss weighted by $(1-\alpha)$ and α for source and target samples, plus β times the MMD distance across all layers. The parameter β is chosen inversely proportional to L to prevent the MMD term from dominating. Training is conducted with SGD (learning rate 0.001, momentum 0.9, batch size 512). Complete implementation details and hyperparameter values are provided in Section SM3.2.

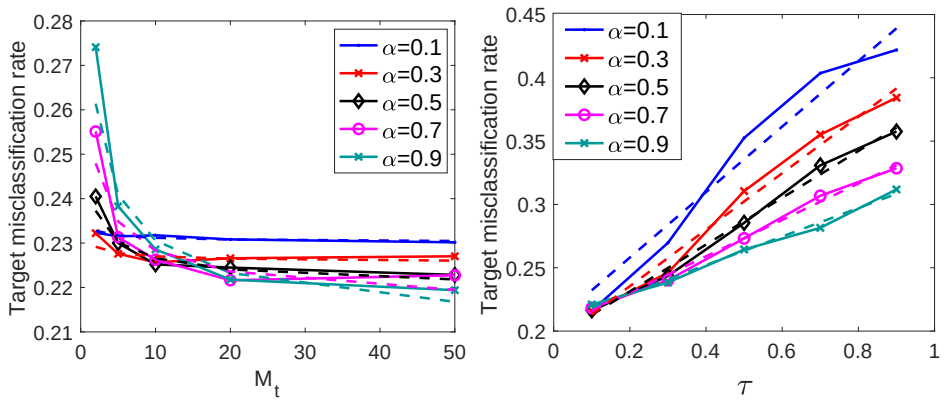


Figure 5. *Variation of the target error on synthetical data with (a) Number of labeled target samples, (b) Distribution distance after transformation. Solid lines indicate experimental data and dashed lines represent theoretical rates of variation.*

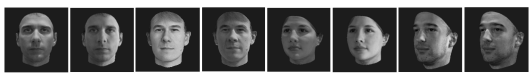


Figure 6. *Sample images from the MIT-CBCL face data set for four different subjects, rendered respectively under poses 1, 2, 5, and 9 for various illumination conditions.*

minimum sample sizes needed to achieve reference accuracy levels, with fitted quadratic polynomials confirming $M_s, N_s = O(L^2)$. Figure 9 demonstrates analogous quadratic growth $M_s = O(d^2)$ with respect to the width parameter d, which scales the width of all MMD layers. These results are consistent with the predictions of Theorem 3.6.

Figure 11 validates the theoretical prediction for the optimal weight parameter. The target accuracy exhibits a non-monotonic variation with α, and the optimal value α_{opt}, identified via polynomial fitting for each M_t, scales as $\alpha_{opt} = O(\sqrt{M_t})$, in agreement with Theorem 3.6.

4.2.2. Adversarial domain adaptation networks. We adopt the domain-adversarial architecture from [21], adapted for our semi-supervised setting.[14]

Figures 10-12 present the sample complexity analysis. The results confirm Theorem 3.11: required sample sizes grow quadratically as $M_s, N_s = O(L^2)$ and $M_s, N_s = O(d^2)$ with respect to network depth and width. Figure 13 further validates that the optimal weight parameter scales as $\alpha_{opt} = O(\sqrt{M_t})$, consistently across different M_s values.

[14]The feature extractor is trained adversarially against a domain discriminator using the objective in Section SM3.2, with negative log likelihood for both classification and domain discrimination losses, and $f^s = f^t = f$. The original implementation [24] was modified to (i) support labeled target samples in the classification loss, and (ii) systematically scale the network capacity via L and d. When studying M_s, the feature extractor and label predictor depths are set equal as L; for N_s, the feature extractor and domain discriminator depths are equated. The width parameter d scales both convolutional channels and fully connected layer widths proportionally. Training uses Adam (learning rate 0.001, batch size 128, 100 epochs). Complete details in Section SM3.2.

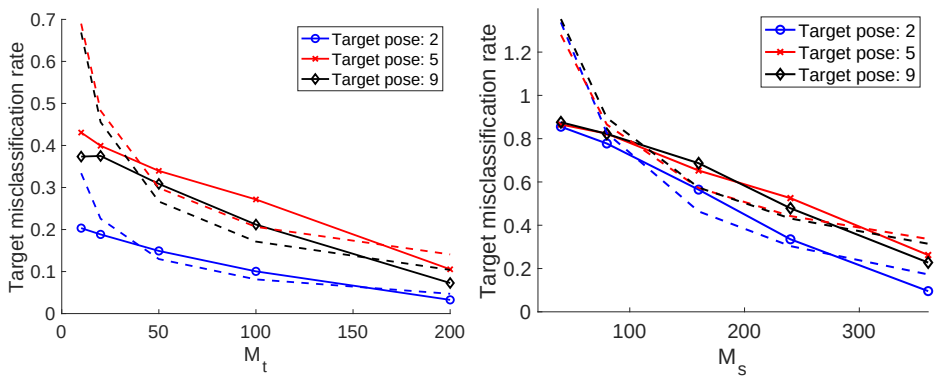


Figure 7. *Variation of the target error on MIT-CBCL face data with (a) Number of labeled target samples, (b) Number of labeled source samples. Solid lines indicate experimental data and dashed lines represent theoretical rates of variation.*

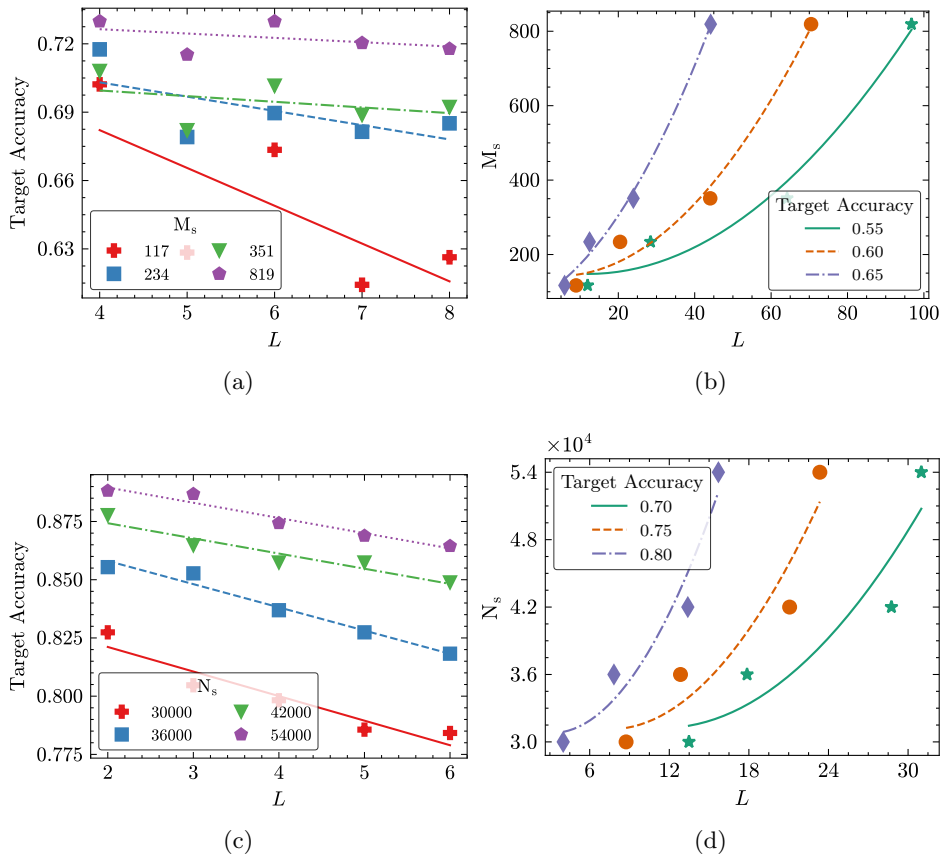


Figure 8. *Sample complexity with respect to depth L for MMD-based networks [31]. Left panels show target accuracy variation with L at different sample sizes. Right panels show quadratic growth $M_s, N_s = O(L^2)$.*

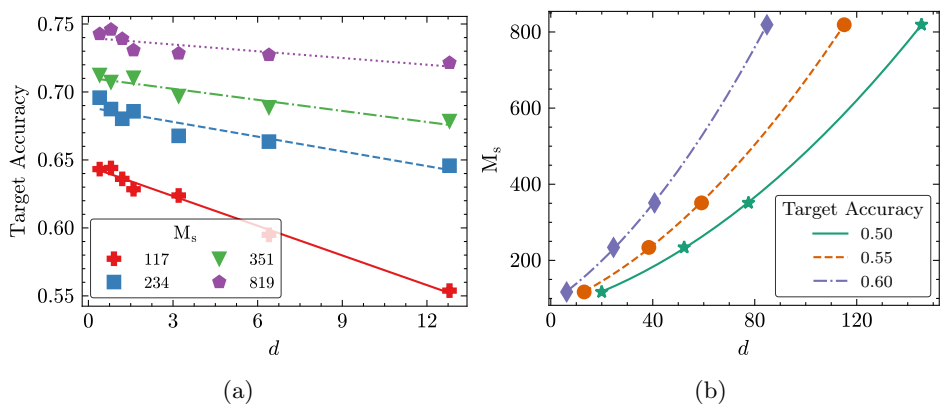


Figure 9. *Sample complexity with respect to width d for MMD-based networks. Quadratic growth $M_s = O(d^2)$ confirmed.*

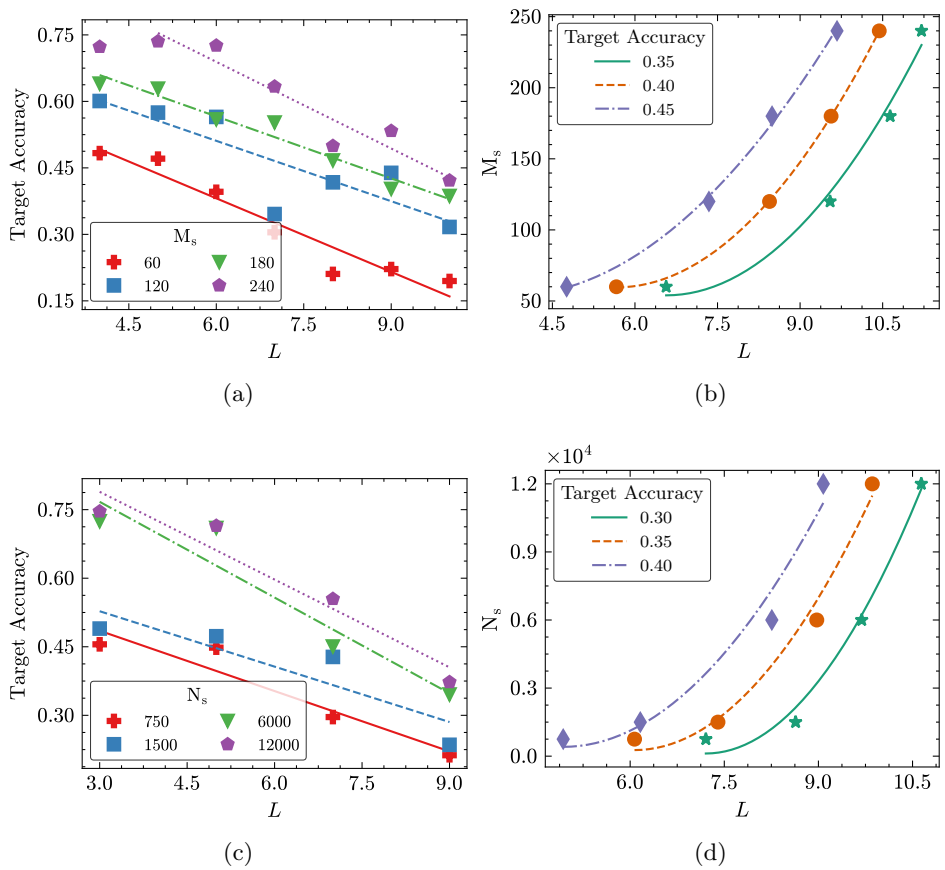


Figure 10. *Sample complexity with respect to depth L for adversarial networks. Left: accuracy vs depth. Right: quadratic growth $M_s, N_s = O(L^2)$.*

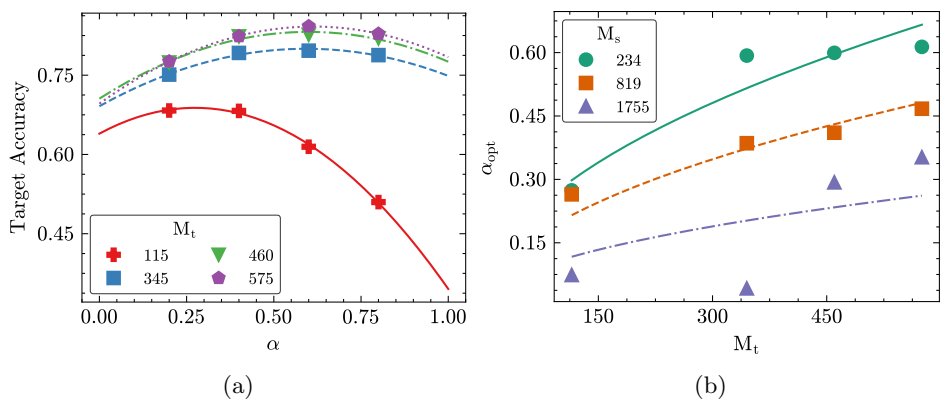


Figure 11. *Optimal weight parameter for MMD-based networks. (a) Target accuracy variation with α shows non-monotonic behavior. (b) Optimal α_{opt} scales as $O(\sqrt{M_t})$.*

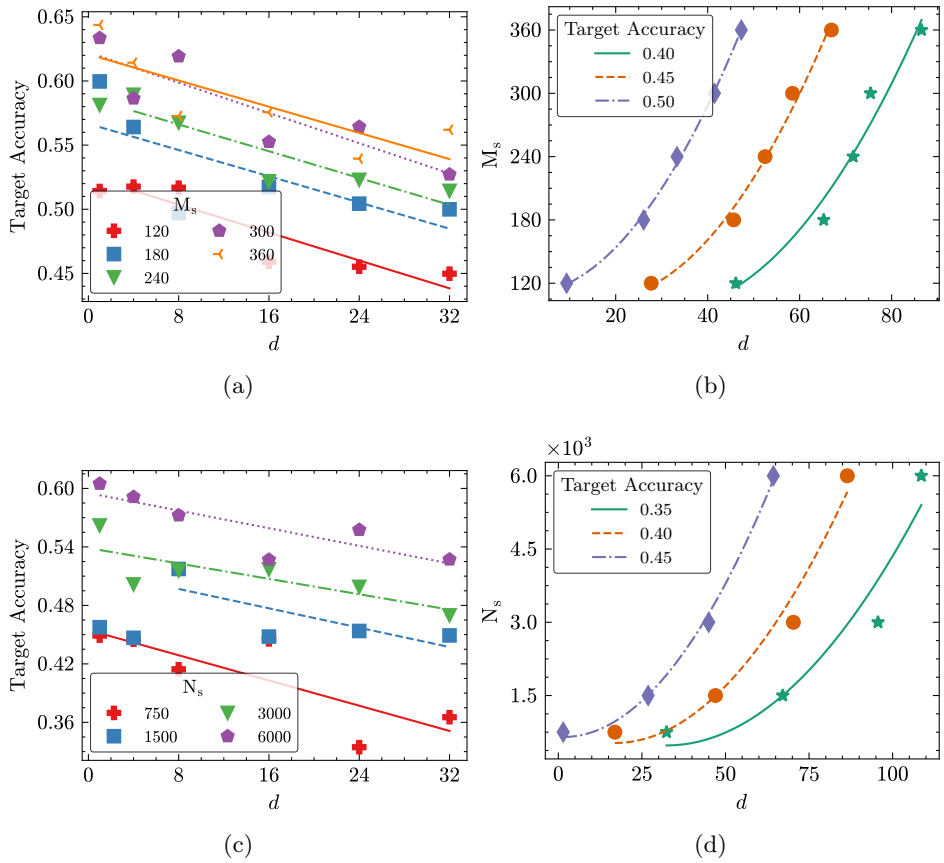


Figure 12. *Sample complexity with respect to width d for adversarial networks. Left: accuracy vs width. Right: quadratic growth* $M_s, N_s = O(d^2)$.

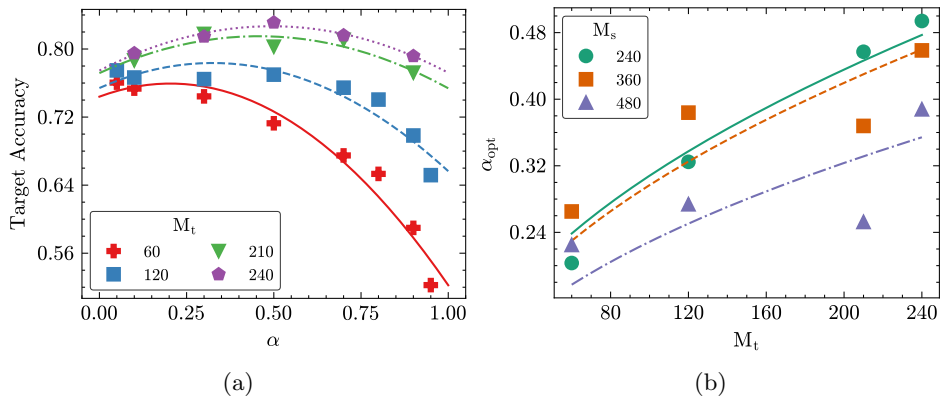


Figure 13. *Optimal weight for adversarial networks. (a) Accuracy vs* α. *(b)* α_{opt} *scales as* $O(\sqrt{M_t})$.

5. Conclusion. We have presented a theoretical analysis of semi-supervised domain adaptation methods that jointly learn feature transformations that map the source and target domains to a shared space, along with a classifier defined in that space. We have first derived general performance bounds applicable to arbitrary function classes and domain discrepancy measures. We have then specialized these results under the assumption that the domain alignment is measured using the maximum mean discrepancy (MMD) metric. Our results show that the number of labeled source samples must scale logarithmically with the covering number of the combined hypothesis class comprising the feature transformation and the classifier, while the total sample sizes must scale logarithmically with the covering numbers of the feature transformation classes alone.

Building on these results, we have then extended our analysis to characterize the sample complexity of domain-adaptive neural networks. Our treatment relies on a detailed examination of the covering numbers of the corresponding function classes in deep architectures. We have focused on two types of neural networks, which perform domain alignment via MMD-based transformations or through adversarial objectives. In both cases, our analysis indicates that the sample complexities for both labeled and unlabeled data grow quadratically with the network depth and width. We have also shown that the scarcity of labeled target data can be effectively mitigated by scaling the weight of the target classification loss proportionally to the square root of the number of labeled target samples.

To the best of our knowledge, our study provides the first comprehensive theoretical characterization of the sample complexity of domain-adaptive neural networks.

Relation to prior work.[15] Previous theoretical analyses of domain adaptation have primarily focused on how domain discrepancy affects generalization when learning a classifier in the original source and target domains, without considering domain-aligning transformations [5, 4, 33]. These works establish generalization bounds in terms of VC-dimensions or Rademacher complexities of hypothesis classes, combined with various distribution divergence measures. While theoretically insightful, many of these divergence measures are difficult to estimate in practice. In contrast, our results in Theorems 2.7-3.11 provide practical generalization bounds based on empirical losses and distribution distances computed directly on aligned training data. Research on neural network sample complexity in single-domain settings [36, 53] has shown dependencies on network size; our work extends these insights to the domain adaptation setting, demonstrating quadratic scaling with both network depth and width.

Acknowledgement. The authors would like to thank Özlem Akgül, Ömer Faruk Arslan, Atilla Can Aydemir, Firdevs Su Aydın and Enes Ata Ünsal for their help with the experiments in Section 4.2.1.

[15] A detailed discussion is provided in the supplement.

REFERENCES

[1] P. L. Anthony, M. Bartlett, *Neural Network Learning - Theoretical Foundations*, Cambridge University Press, Cambridge, UK, 2002.

[2] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar, *Regularized learning for domain adaptation under label shifts*, in Int. Conf. Learning Representations, 2019.

[3] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, *Unsupervised domain adaptation by domain invariant projection*, in IEEE International Conference on Computer Vision, 2013, pp. 769–776.

[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, *A theory of learning from different domains*, Machine Learning, 79 (2010), pp. 151–175.

[5] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, *Analysis of representations for domain adaptation*, in Proc. Advances in Neural Information Processing Systems 19, 2006, pp. 137–144.

[6] K. Bousmalis et al., *Domain separation networks*, in Adv. Neural Information Processing Systems, 2016, pp. 343–351.

[7] N. Courty et al., *Optimal transport for domain adaptation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (2017), pp. 1853–1865.

[8] F. Cucker and S. Smale, *On the Mathematical Foundations of Learning*, Bulletin of the American Mathematical Society, 39 (2002), pp. 1–49.

[9] B. B. Damodaran et al., *Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation*, in European Conf. Comp. Vision, vol. 11208, 2018, pp. 467–483.

[10] A. Daniely and E. Granot, *On the sample complexity of two-layer networks: Lipschitz vs. element-wise Lipschitz activation*, in International Conference on Algorithmic Learning Theory, vol. 237, 2024, pp. 505–517.

[11] H. Daumé III, *Frustratingly easy domain adaptation*, in Annual Meeting-Association for Computational Linguistics, 2007.

[12] H. Daumé III, A. Kumar, and A. Saha, *Co-regularization based semi-supervised domain adaptation*, in Proc. Advances in Neural Information Processing Systems 23, 2010, pp. 478–486.

[13] S. Dhouib, I. Redko, and C. Lartizien, *Margin-aware adversarial domain adaptation with optimal transport*, in Proc. Int. Conf. Machine Learning,, vol. 119, 2020, pp. 2514–2524.

[14] L. Duan, D. Xu, and I. W. Tsang, *Learning with augmented features for heterogeneous domain adaptation*, in Proc. 29th International Conference on Machine Learning, 2012.

[15] N. Dunford and J. Schwartz, *Linear Operators, Part 1: General Theory*, Wiley Classics Library, Interscience Publishers Inc., New York, 1988.

[16] M. El Hamri, Y. Bennani, and I. Falih, *Theoretical guarantees for domain adaptation with hierarchical optimal transport*, Mach. Learn., 114 (2025), p. 119.

[17] Z. Fang, J. Lu, F. Liu, and G. Zhang, *Semi-supervised heterogeneous domain adaptation: Theory and algorithms*, IEEE Trans. Pattern Anal. Mach. Intell., 45 (2023), pp. 1087–1105.

[18] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, *Unsupervised visual domain adaptation using subspace alignment*, in IEEE International Conference on Computer Vision, 2013, pp. 2960–2967.

[19] T. Galanti, L. Wolf, and T. Hazan, *A theoretical framework for deep transfer learning*, Information and Inference: A Journal of the IMA, 5 (2016), pp. 159–209.

[20] Y. Ganin and V. Lempitsky, *Unsupervised domain adaptation by backpropagation*, in Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015, pp. 1180–1189.

[21] Y. Ganin et al., *Domain-adversarial training of neural networks*, J. Mach. Learn. Res., 17 (2016), pp. 59:1–59:35.

[22] M. Ghifary, W. B. Kleijn, and M. Zhang, *Domain adaptive neural networks for object recognition*, in Int. Conf. Artificial Intelligence, vol. 8862, 2014, pp. 898–904.

[23] M. Ghifary et al., *Deep reconstruction-classification networks for unsupervised domain adaptation*, in European Conf. Comp. Vision,, vol. 9908, 2016, pp. 597–613.

[24] GitHub repository, *Dann_py3*, 2023. [Online]. Available: https://github.com/fungtion/DANN_py3.git.

[25] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, *A kernel two-sample test*, J. Mach. Learn. Res., 13 (2012), pp. 723–773.

[26] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, *Correcting sample selection bias by unlabeled data*, in Proc. Advances in Neural Information Processing Systems 19, 2006, pp. 601–608.
[27] Y. Jiao, H. Lin, Y. Luo, and J. Z. Yang, *Deep transfer learning: Model framework and error analysis*, arXiv preprint: http://arxiv.org/abs/2410.09383, (2024).
[28] H. Karaca et al., *An experimental study of the sample complexity of domain adaptation*, in IEEE Signal Processing and Communications Applications Conference, 2023, pp. 1–4.
[29] W. M. Kouw and M. Loog, *A review of domain adaptation without target labels*, IEEE Trans. Pattern Anal. Mach. Intell., 43 (2021), pp. 766–785.
[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.
[31] M. Long, Y. Cao, J. Wang, and M. I. Jordan, *Learning transferable features with deep adaptation networks*, in Proc 32nd International Conference on Machine Learning, vol. 37, pp. 97–105.
[32] M. Long, Z. Cao, J. Wang, and M. I. Jordan, *Conditional adversarial domain adaptation*, in Advances in Neural Information Processing Systems, 2018, pp. 1647–1657.
[33] Y. Mansour, M. Mohri, and A. Rostamizadeh, *Domain adaptation: Learning bounds and algorithms*, in The 22nd Conference on Learning Theory, 2009.
[34] Massachusetts Institute of Technology, *MIT-CBCL face recognition database*. Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html.
[35] D. McNamara and M. Balcan, *Risk bounds for transferring representations with and without fine-tuning*, in Proc. Int. Conf. Machine Learning,, vol. 70, 2017, pp. 2373–2381.
[36] B. Neyshabur, R. Tomioka, and N. Srebro, *Norm-based capacity control in neural networks*, in Prof. 28th Conference on Learning Theory, vol. 40, 2015, pp. 1376–1401.
[37] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, *Domain adaptation via transfer component analysis*, IEEE Trans. Neural Networks, 22 (2011), pp. 199–210.
[38] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani, *A survey on domain adaptation theory*, arXiv preprint: http://arxiv.org/abs/2004.11829, (2020).
[39] P. Singhal, R. Walambe, S. Ramanna, and K. Kotecha, *Domain adaptation: Challenges, methods, datasets, and applications*, IEEE Access, 11 (2023), pp. 6973–7020.
[40] B. Sun and K. Saenko, *Deep CORAL: correlation alignment for deep domain adaptation*, in European Conf. Comp. Vision, vol. 9915, 2016, pp. 443–450.
[41] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, *A two-stage weighting framework for multi-source domain adaptation*, in Proc. Advances in Neural Information Processing Systems 24, 2011, pp. 505–513.
[42] R. Tachet des Combes et al., *Domain adaptation with conditional distribution matching and generalized label shift*, in Neural Inf. Proc. Systems, 2020.
[43] H. Tang and K. Jia, *Discriminative adversarial domain adaptation*, in AAAI Conference on Artificial Intelligence, 2020, pp. 5940–5947.
[44] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, *Simultaneous deep transfer across domains and tasks*, in IEEE International Conference on Computer Vision, 2015, pp. 4068–4076.
[45] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, *Adversarial discriminative domain adaptation*, in IEEE Conference on Computer Vision and Pattern Recognition,, 2017, pp. 2962–2971.
[46] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, *Deep domain confusion: Maximizing for domain invariance*, arXiv preprint: http://arxiv.org/abs/1412.3474, (2014).
[47] G. Vardi, O. Shamir, and N. Srebro, *The sample complexity of one-hidden-layer neural networks*, in Advances in Neural Information Processing Systems 35, 2022.
[48] E. Vural, *Generalization bounds for domain adaptation via domain transformations*, in IEEE Int. Workshop Machine Learning for Signal Processing, 2018, pp. 1–6.
[49] M. Wang and W. Deng, *Deep visual domain adaptation: A survey*, Neurocomputing, 312 (2018), pp. 135–153.
[50] X. Wang and J. Schneider, *Generalization bounds for transfer learning under model shift*, in Proc. Conf. Uncertainty in Artificial Intelligence, 2015, pp. 922–931.
[51] Z. Wang and Y. Mao, *On f-divergence principled domain adaptation: An improved framework*, in Advances in Neural Information Processing Systems, 2024.

[52] P. Wang et al., *Information maximizing adaptation network with label distribution priors for unsupervised domain adaptation*, IEEE Transactions on Multimedia, 25 (2023), pp. 6026–6039.
[53] C. Wei and T. Ma, *Data-dependent sample complexity of deep neural networks via Lipschitz augmentation*, in Advances in Neural Information Processing Systems 32, 2019, pp. 9722–9733.
[54] Z. Xia et al., *Meta domain adaptation approach for multi-domain ranking*, IEEE Access, 13 (2025), pp. 92921–92931.
[55] B. Yang et al., *Point-to-set metric-gated mixture of experts for multisource domain adaptation fault diagnosis*, IEEE Transactions on Neural Networks and Learning Systems, (2025), pp. 1–15.
[56] T. Yao, Y. Pan, C. Ngo, H. Li, and T. Mei, *Semi-supervised domain adaptation with subspace learning for visual recognition*, in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2142–2150.
[57] V. V. Yurinskii, *Exponential inequalities for sums of random vectors*, Journal of Multivariate Analysis, 6 (1976), pp. 473–499.
[58] Y. Zeng et al., *Multirepresentation dynamic adaptive network for cross-domain rolling bearing fault diagnosis in complex scenarios*, IEEE Transactions on Instrumentation and Measurement, 74 (2025), pp. 1–16.
[59] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, *Bridging theory and algorithm for domain adaptation*, in Proceedings of the 36th International Conference on Machine Learning, vol. 97, 2019, pp. 7404–7413.
[60] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan, *Multi-class heterogeneous domain adaptation*, Journal of Machine Learning Research, 20 (2019), pp. 1–31.
[61] M. H. Zonoozi and V. Seydi, *A survey on adversarial domain adaptation*, Neural Process. Lett., 55 (2023), pp. 2429–2469.
[62] M. H. P. Zonoozi, V. Seydi, and M. Deypir, *An unsupervised adversarial domain adaptation based on variational auto-encoder*, Mach. Learn., 114 (2025), p. 128.