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 [37], label shift [2], [54], as well as more challenging heterogeneous settings with source and target samples originating from different data spaces [50]. Early work in domain adaptation explored instance reweighting methods for covariate shift [34], [53], feature augmentation approaches [16], [17], [20], and techniques for learning feature projections or transformations [4], [47], [72]. 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 11, 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 [50], [62]. Metrics such as maximum mean discrepancy (MMD) [39], [59], [28] lead to efficient solutions for aligning source and target domains across various applications [75], [67], [70], [71]. Adversarial architectures [27], [58], [55], [78] and reconstruction-based approaches using encoder-decoder structures [29], [10], [79] 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 [39], covariance-based metrics [52], and the Wasserstein distance [12], [14], [22]. Although domain adaptation algorithms have been successfully applied across a wide range of fields including computer vision, time-series analysis, and natural language processing [50], [78], 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 [48], [8], [41], [76], [19], [65]. 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 [77], [23], [63]; however, these works rather focus on specific transformation types, such as linear mappings [77] or location and scale changes [63]. Some literature has investigated the performance and sample complexity of transfer learning via deep learning approaches [25], [43], [35]. 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], [46], [68], [60], [15]. 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 2.9, 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 3.1-3.2 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 [39], [59], [28] and adversarial domain adaptation networks [27], [58], [55]. Our results in Theorems 3.6 and 3.8 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 2.9 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 4, where we also compare and contrast our results with previous findings. Section 5 presents some simulation results for the experimental validation of our findings, and Section 6 concludes the paper. A preliminary version of our study was presented in [61], 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

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

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

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

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.

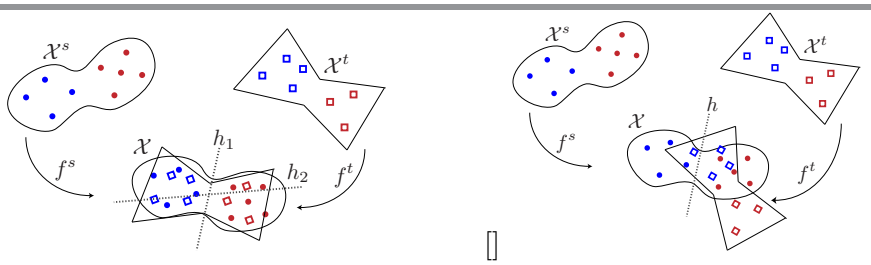

Figure 1. *Illustration of Assumption 2.2. Red and blue colors represent two different classes in the source and target domains $\mathcal{X}^s$ and $\mathcal{X}^t$. In (a), the two domains are well-aligned by the learnt transformations; therefore, the source and target losses are similar. In (b), the 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:

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 2.2 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 1, 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 1 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 2.2. 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 2.2. 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.

Lemma 2.1. *Consider that Assumption 2.2 holds. Let $\mathcal{L}_\alpha(f^s, f^t, h)$ denote the expected weighted loss in the source and target domains given by*

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

Then the expected target loss is bounded as

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

Proof. We have $\mathcal{L}^t(f^t, h) = \alpha\mathcal{L}^t(f^t, h) + (1-\alpha)\mathcal{L}^t(f^t, h)$. From Assumption 2.2, we get

$$\mathcal{L}^t(f^t, h) \leq \mathcal{L}^s(f^s, h) + R\ D(f^s, f^t).$$

Using this above, we obtain

$$\begin{aligned}\mathcal{L}^t(f^t, h) &\leq \alpha\mathcal{L}^t(f^t, h) + (1-\alpha)\left(\mathcal{L}^s(f^s, h) + R\ D(f^s, f^t)\right)\\ &= \mathcal{L}_\alpha(f^s, f^t, h) + (1-\alpha)RD(f^s, f^t).\end{aligned}$$

■

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 [13]:

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*

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

In order to study the discrepancy between the expected and the empirical losses, we next make the following assumptions. The composite function classes $\mathcal{H} \circ \mathcal{F}^s\{g^s = h \circ f^s : h \in \mathcal{H}, f^s \in \mathcal{F}^s\}$ and $\mathcal{H} \circ \mathcal{F}^t\{g^t = h \circ f^t : h \in \mathcal{H}, f^t \in \mathcal{F}^t\}$ are compact metric spaces with respect to the metrics

$$\begin{aligned}\mathfrak{d}^s(g_1^s, g_2^s) &\sup_{x^s \in \mathcal{X}^s} \|g_1^s(x^s) - g_2^s(x^s)\|\\ \mathfrak{d}^t(g_1^t, g_2^t) &\sup_{x^t \in \mathcal{X}^t} \|g_1^t(x^t) - g_2^t(x^t)\|\end{aligned} \tag{2.4}$$

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

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

We can now present the following result that bounds the deviation between the expected and empirical weighted losses.

Lemma 2.3. *Let the conditions in Assumption 2.2 hold. Let*

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

denote the empirical weighted loss. Then, we have

$$P\left(\sup_{f^s\in\mathcal{F}^s, f^t\in\mathcal{F}^t, h\in\mathcal{H}} |\mathcal{L}_\alpha(f^s,f^t,h) - \hat{\mathcal{L}}_\alpha(f^s,f^t,h)| \le \epsilon\right)$$
$$\ge 1 - 2\mathcal{N}(\mathcal{H}\circ\mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t)e^{-\frac{M_t\epsilon^2}{8\alpha^2 A_\ell^2}} - 2\mathcal{N}(\mathcal{H}\circ\mathcal{F}^s, \frac{\epsilon}{8(1-\alpha)L_\ell}, \mathfrak{d}^s)e^{-\frac{M_s\epsilon^2}{8(1-\alpha)^2 A_\ell^2}}.$$

The proof of Lemma 2.3 is given in Appendix A.

We can now simply combine Lemmas 2.1 and 2.3 to bound the expected target loss in terms of the empirical weighted loss and the distribution distance in the following main result.

Theorem 2.4. *Let Assumptions 2.2, 2.2 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 $O(\sqrt{M_t})$.

Remark 2.5. 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

$$P(|D(f^s,f^t) - \hat{D}(f^s,f^t)| \ge \epsilon) \le (\mathcal{N}_{\mathcal{F}^s,\epsilon} + \mathcal{N}_{\mathcal{F}^t,\epsilon})\, O\left(e^{-N\epsilon^2}\right)$$
$$\le O\left(e^{-M_t\epsilon^2}\right) + O\left(e^{-M_s\epsilon^2}\right)$$

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

$$\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1-\alpha)R\hat{D}(f^s, f^t) + \epsilon + (1-\alpha)R\epsilon. \tag{2.6}$$

Our purpose in the next section is to establish such a result for the particular setting where the distribution distance is chosen as the MMD.

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 mathcing (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

$$D(f^s, f^t) = \| E_{x^s}[f^s(x^s)] - E_{x^t}[f^t(x^t)] \|_{\mathcal{X}} \tag{2.7}$$

where $\| \cdot \|_{\mathcal{X}}$ stands for the inner-product-induced norm in the Hilbert space $\mathcal{X}$. 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. 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

$$\hat{D}(f^s, f^t) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} f^s(x_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} f^t(x_j^t) \right\|. \tag{2.8}$$

Remark 2.6. Although most KMM methods assume the source and target domains to be the same ($\mathcal{X}^s = \mathcal{X}^t$), and also the source and target transformations to be the same ($f^s = f^t = \phi$), we do not make use of these assumptions in the analysis presented in this section. Here, we only assume that the distribution discrepancy between ν_s and ν_t is taken as in (2.7) for any two transformations $f^s \in \mathcal{F}^s$ and $f^t \in \mathcal{F}^t$, and the empirical estimate of the MMD is computed as in (2.8).

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

The expected deviations of the random variables $\{f^s(x_i^s)\}_{i=1}^{N_s}$ and $\{f^t(x_j^t)\}_{j=1}^{N_t}$ from their means $E[f^s(x^s)]$ and $E[f^t(x^t)]$ are bounded such that there exist constants σ_s^2 and σ_t^2 satisfying

$$\begin{aligned} E\left[\|f^s(x_i^s) - E[f^s(x^s)]\|^2\right] &\leq \sigma_s^2 \\ E\left[\|f^t(x_j^t) - E[f^t(x^t)]\|^2\right] &\leq \sigma_t^2. \end{aligned} \tag{2.9}$$

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

$$\begin{aligned} E\left[\|f^s(x_i^s) - E[f^s(x^s)]\|^k\right] &\leq \frac{k!}{2}\sigma_s^2\, C_s^{k-2} \\ E\left[\|f^t(x_j^t) - E[f^t(x^t)]\|^k\right] &\leq \frac{k!}{2}\sigma_t^2\, C_t^{k-2}. \end{aligned} \tag{2.10}$$

The condition (2.9) can be seen as a finite variance assumption for a distribution over a Hilbert space, and the condition (2.10) 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.

We first present the following lemma, which bounds the deviation between the expectation and the empirical mean of the source and the target data mapped to the common domain $\mathcal{X}$ via the transformations f^s and f^t.

Lemma 2.7. *Let the source and target distributions and the transformations $f^s : \mathcal{X}^s \to \mathcal{X}$ and $f^t : \mathcal{X}^t \to \mathcal{X}$ be such that Assumption 2.3 holds. Also, for given $\epsilon > 0$, let the number of source and target samples be such that*

$$N_s > \frac{\sigma_s^2}{\epsilon^2}, \quad N_t > \frac{\sigma_t^2}{\epsilon^2}.$$

Then for the source domain we have

$$\begin{aligned} &P\left(\left\|\frac{1}{N_s}\sum_{i=1}^{N_s} f^s(x_i^s) - E[f^s(x^s)]\right\| \geq \epsilon\right) \\ &\leq \exp\left(-\frac{1}{8}\left(\frac{\sqrt{N_s}\epsilon}{\sigma_s} - 1\right)^2 \frac{1}{1 + \left(\frac{\sqrt{N_s}\epsilon}{\sigma_s} - 1\right)\frac{C_s}{2\sqrt{N_s}\sigma_s}}\right) \end{aligned} \tag{2.11}$$

and for the target domain we have

$$\begin{aligned} &P\left(\left\|\frac{1}{N_t}\sum_{j=1}^{N_t} f^t(x_j^t) - E[f^t(x^t)]\right\| \geq \epsilon\right) \\ &\leq \exp\left(-\frac{1}{8}\left(\frac{\sqrt{N_t}\epsilon}{\sigma_t} - 1\right)^2 \frac{1}{1 + \left(\frac{\sqrt{N_t}\epsilon}{\sigma_t} - 1\right)\frac{C_t}{2\sqrt{N_t}\sigma_t}}\right). \end{aligned} \tag{2.12}$$

The proof of Lemma 2.7 is given in Appendix B. Lemma 2.7 provides a bound on the deviation between the sample mean and the expectation of the source and target samples

transformed to the shared Hilbert space $\mathcal{X}$. In particular, it states that as the number N_s, N_t of source and target samples increases, this deviation can be upper bounded with probability improving at an exponential rate with N_s and N_t. We next build on this result to present in Lemma 2.8 a uniform upper bound on the deviation $|D(f^s, f^t) - \hat{D}(f^s, f^t)|$ between the expected and empirical MMD distances, which is valid for any $f^s \in \mathcal{F}^s$ and $f^t \in \mathcal{F}^t$. We first need an assumption on the compactness of the function classes $\mathcal{F}^s$ and $\mathcal{F}^t$:

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

$$\begin{aligned} &\mathfrak{d}^s_{\mathcal{X}}(f^s_1, f^s_2) \sup_{x^s \in \mathcal{X}^s} \|f^s_1(x^s) - f^s_2(x^s)\| \\ &\mathfrak{d}^t_{\mathcal{X}}(f^t_1, f^t_2) \sup_{x^t \in \mathcal{X}^t} \|f^t_1(x^t) - f^t_2(x^t)\|. \end{aligned} \tag{2.13}$$

Lemma 2.8. *Let Assumptions 2.3, 2.3 hold. Given $\epsilon > 0$, let the number of source and target samples be such that*

$$N_s > \frac{16\sigma_s^2}{\epsilon^2}, \quad N_t > \frac{16\sigma_t^2}{\epsilon^2}.$$

Let us define the functions

$$\begin{aligned} a_s(N_s, \epsilon) &\frac{1}{8}\left(\frac{\sqrt{N_s}\epsilon}{4\sigma_s} - 1\right)^2 \frac{1}{1 + \left(\frac{\sqrt{N_s}\epsilon}{4\sigma_s} - 1\right)\frac{C_s}{2\sqrt{N_s}\sigma_s}} \\ a_t(N_t, \epsilon) &\frac{1}{8}\left(\frac{\sqrt{N_t}\epsilon}{4\sigma_t} - 1\right)^2 \frac{1}{1 + \left(\frac{\sqrt{N_t}\epsilon}{4\sigma_t} - 1\right)\frac{C_t}{2\sqrt{N_t}\sigma_t}}. \end{aligned}$$

Then

$$\begin{aligned} &P\left(\sup_{f^s \in \mathcal{F}^s, f^t \in \mathcal{F}^t} |D(f^s, f^t) - \hat{D}(f^s, f^t)| < \epsilon\right) \\ &\geq 1 - \mathcal{N}(\mathcal{F}^s, \frac{\epsilon}{8}, \mathfrak{d}^s_{\mathcal{X}}) \exp(-a_s(N_s, \epsilon)) - \mathcal{N}(\mathcal{F}^t, \frac{\epsilon}{8}, \mathfrak{d}^t_{\mathcal{X}}) \exp(-a_t(N_t, \epsilon)). \end{aligned}$$

Lemma 2.8 is proved in Appendix C. The lemma provides a probabilistic upper bound on the deviation between the actual MMD and its estimate from a finite sample set, which holds for all functions in the transformation function classes $\mathcal{F}^s$ and $\mathcal{F}^t$. We are now ready to combine this bound with our results in Section 2.2. We recall that in Theorem 2.4, the expected target loss $\mathcal{L}^t(f^t, h)$ was bounded in terms of the empirical weighted loss $\mathcal{L}_\alpha(f^s, f^t, h)$ and the true distribution discrepancy $D(f^s, f^t)$ after the transformations. However, in practice, for two transformations f^s, f^t computed by a domain adaptation method, the true distribution discrepancy $D(f^s, f^t)$ is often unknown. We are now in a position to extend Theorem 2.4 in the following result, where we bound the expected target loss in terms of the empirical MMD measure $\hat{D}(f^s, f^t)$.

Theorem 2.9. *Consider a domain adaptation algorithm where the distribution discrepancy is taken as the MMD measure, and the loss function and data distributions satisfy Assumptions*

2.2-2.3. For $\epsilon > 0$, let the number of source and target samples satisfy

$$N_s > \frac{16\sigma_s^2}{\epsilon^2}, \qquad N_t > \frac{16\sigma_t^2}{\epsilon^2}.$$

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

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

the expected target loss is upper bounded as

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

Proof. The stated result follows simply from Theorem 2.4 and Lemma 2.8 by applying the union bound. ∎

The result in Theorem 2.9 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.

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 Section 3.1 and Section 3.2.

3.1. MMD-based domain adaptation networks.

We begin with studying the implications of Theorem 2.9 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, which serves as a commonly adopted foundation for many MMD-based neural network architectures. The source and target samples first pass through a common network, possibly comprising multiple convolutional and fully connected layers. The common network output is then provided to a source network and a target network consisting of $L-1$ fully connected layers in the corresponding domain, with the L-th (output) layer consisting of a classifier that is shared between the two domains. The action of the common network remains out of the scope of our study, as its parameters are often adopted from a pre-trained network or fine-tuned using only a set of source samples in the literature [39], [59], [28]. We hence consider

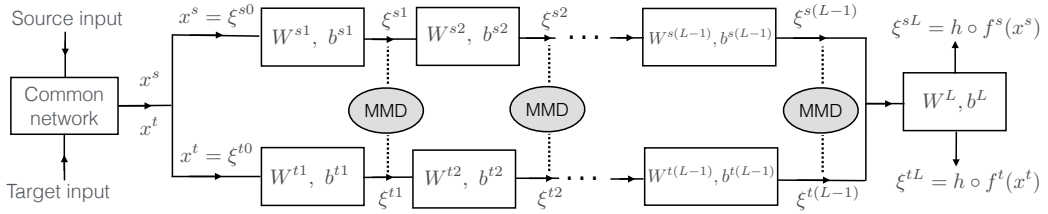

Figure 2. *Illustration of MMD-based domain adaptation networks*

the feature representations at the output of the common network as our source and target domain samples x^s and x^t. Defining $\boldsymbol{\xi}^{s0}x^s \in \mathbb{R}^{d_0}$ and $\boldsymbol{\xi}^{t0}x^t \in \mathbb{R}^{d_0}$, the relation between the features of layers l and $l-1$ is given by

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

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

$$\begin{aligned} \boldsymbol{\Theta}^s &= (\boldsymbol{\Theta}^{s1}, \ldots, \boldsymbol{\Theta}^{sL}) \\ \boldsymbol{\Theta}^t &= (\boldsymbol{\Theta}^{t1}, \ldots, \boldsymbol{\Theta}^{tL}) \end{aligned}$$

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. The source and target domains are given by

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

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

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

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

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) $\mathcal{X}^l$ [39, 32]. 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 $\mathcal{X}^l$ [32]. The feature mapping ϕ^l and the characteristic kernel k^l are related as $\phi^l(\boldsymbol{\xi}^l) = k^l(\boldsymbol{\xi}^l, \cdot) : \mathbb{R}^{d_l} \to \mathbb{R}$ [32]. 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

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

for $l = 1, \ldots, L-1$. Note that the direct sum

$$\mathcal{X} = \bigoplus_{l=1}^{L-1} \mathcal{X}^l = \{(f^1, f^2, \ldots, f^{L-1}) : f^l \in \mathcal{X}^l,\ l = 1, \ldots, L-1\}$$

of the RKHSs $\mathcal{X}^1, \ldots, \mathcal{X}^{L-1}$ is also a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ given by [21]

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

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

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

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.,

$$\begin{aligned} g^s(x^s) &= (h \circ f^s)(x^s)\boldsymbol{\xi}^{sL}(x^s) \\ g^t(x^t) &= (h \circ f^t)(x^t)\boldsymbol{\xi}^{tL}(x^t). \end{aligned} \tag{3.7}$$

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.

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 demonstrated in Lemma 3.1, this assumption ensures that $E[f^s(x^s)]$ and $E[f^t(x^t)]$ are in $\mathcal{X}$, whose proof is presented in Appendix D.

Lemma 3.1. *Let the condition in Assumption 3.1 hold. Then the mappings $f^{sl} : \mathcal{X}^s \to \mathcal{X}^l$ and $f^{tl} : \mathcal{X}^t \to \mathcal{X}^l$ for $l = 1,\dots,L-1$, and the mappings $f^s : \mathcal{X}^s \to \mathcal{X}$ and $f^t : \mathcal{X}^t \to \mathcal{X}$ are measurable. Moreover, assuming that $E[\sqrt{k^l(\boldsymbol{\xi}^{sl},\boldsymbol{\xi}^{sl})}] < \infty$ and $E[\sqrt{k^l(\boldsymbol{\xi}^{tl},\boldsymbol{\xi}^{tl})}] < \infty$, the functions $E[f^{sl}(x^s)] : \mathbb{R}^{d_l} \to \mathbb{R}$ and $E[f^{tl}(x^t)] : \mathbb{R}^{d_l} \to \mathbb{R}$ defined as*

$$\begin{aligned} &E[f^{sl}(x^s)](\cdot)E[f^{sl}(x^s)(\cdot)] \\ &E[f^{tl}(x^t)](\cdot)E[f^{tl}(x^t)(\cdot)] \end{aligned}$$

through the Borel probability measures μ_s and μ_t in the source and target domains are in the RKHSs $\mathcal{X}^l$. Consequently, the functions

$$\begin{aligned} &E[f^s(x^s)](E[f^{s1}(x^s)],\dots,E[f^{s(L-1)}(x^s)]) \\ &E[f^t(x^t)](E[f^{t1}(x^t)],\dots,E[f^{t(L-1)}(x^t)]) \end{aligned}$$

are in the Hilbert space $\mathcal{X}$.

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

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

MMD-based domain adaptation algorithms typically seek to minimize the empirical estimate $\hat{D}^l$ of D^l at each layer [39], [59], [28]. The empirical distribution discrepancy $\hat{D}^l$ is obtained from the source and target sample sets $\{x^s_i\}_{i=1}^{N_s}$ and $\{x^t_j\}_{j=1}^{N_t}$ as

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

where $\boldsymbol{\xi}^{sl}_i$ and $\boldsymbol{\xi}^{tl}_j$ denote the source and target features in layer l corresponding respectively to the samples x^s_i and x^t_j. The second equality follows from the relations $f^{sl}(x^s_i) = \phi^l(\boldsymbol{\xi}^{sl}_i)$ and $f^{tl}(x^t_j) = \phi^l(\boldsymbol{\xi}^{tl}_j)$.

The overall distribution discrepancy between the source and the target domains defined in (2.7) 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 3.1 in the current setting. Its empirical estimate $\hat{D}(f^s, f^t)$ defined in (2.8) is then obtained as

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

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 (3.9) summed over all layers [62], [39], [59], [28]. We thus consider a learning algorithm that minimizes the overall loss

$$\min_{f^s\in\mathcal{F}^s,\ f^t\in\mathcal{F}^t,\ h\in\mathcal{H}} (1-\alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha\hat{\mathcal{L}}^t(f^t, h) + \beta\sum_{l=1}^{L-1}(\hat{D}^l)^2(f^{sl}, f^{tl}). \tag{3.10}$$

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 2.9 applies to the current problem. Before we proceed with the implications of Theorem 2.9, we need two additional assumptions.

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

$$|k^l(\boldsymbol{\xi}_1, \boldsymbol{\xi}) - k^l(\boldsymbol{\xi}_2, \boldsymbol{\xi})| \le L_K\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\| \tag{3.11}$$

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

$$\|\eta^l(\mathbf{u}) - \eta^l(\mathbf{v})\| \le L_\eta\,\|\mathbf{u} - \mathbf{v}\| \tag{3.12}$$

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

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

$$|\eta_i^l(\mathbf{u})| \le C_\eta \tag{3.13}$$

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

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

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

The Lipschitz continuity condition (3.11) holds for many widely used kernels such as Gaussian kernels. As for condition (3.12), the Lipschitz constants of the commonly used rectified linear unit, softmax and softplus activation functions are derived in Appendix E. In the following result we show that the transformation function classes $\mathcal{F}^s, \mathcal{F}^t$ as well as the composite function classes $\mathcal{G}^s$, $\mathcal{G}^t$ are compact metric spaces.

Lemma 3.2. *Let Assumptions 3.1-3.1 hold. Then, the transformation function classes $\mathcal{F}^s, \mathcal{F}^t$ in (3.6) and the composite function classes $\mathcal{G}^s, \mathcal{G}^t$ in (3.8) are compact metric spaces, respectively under the metrics $\mathfrak{d}_{\mathcal{X}}^s, \mathfrak{d}_{\mathcal{X}}^t$ in (2.13), and the metrics $\mathfrak{d}^s, \mathfrak{d}^t$ in (2.4).*

The proof of Lemma 3.2 is presented in Appendix F. Having established the compactness of the function classes, we can now study the corresponding covering numbers.

Lemma 3.3. *Let Assumptions 3.1, 3.1, 3.1 hold. Then, the covering numbers of the function classes $\mathcal{F}^s$ and $\mathcal{F}^t$ are upper bounded as*

$$\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}_{\mathcal{X}}^s) \leq \prod_{l=1}^{L-1} \left(\frac{4A_\Theta L_K Q}{\epsilon^2} + 1 \right)^{d_l(d_{l-1}+1)}$$
$$\mathcal{N}(\mathcal{F}^t, \epsilon, \mathfrak{d}_{\mathcal{X}}^t) \leq \prod_{l=1}^{L-1} \left(\frac{4A_\Theta L_K Q}{\epsilon^2} + 1 \right)^{d_l(d_{l-1}+1)}$$

where the dimension-dependent constant Q is defined as

$$Q \sum_{l=1}^{L-1} Q_l$$

with

$$\begin{aligned} &Q_l(L_\eta R_{l-1}\sqrt{d_l d_{l-1}} + L_\eta \sqrt{d_l}) \\ &+ \sum_{i=1}^{l-1} (L_\eta R_{i-1}\sqrt{d_i d_{i-1}} + L_\eta \sqrt{d_i}) \prod_{k=i+1}^{l} L_\eta A_\Theta \sqrt{d_k d_{k-1}} \end{aligned} \tag{3.15}$$

for $l = 2, \ldots, L$ and $Q_1 L_\eta \sqrt{d_1 d_0}\, R_0 + L_\eta \sqrt{d_1}$. Here

$$\begin{aligned} &R_l (A_\eta A_\Theta)^l (A_x \sqrt{d_0} + 1)\sqrt{d_1} \prod_{k=1}^{l-1} \sqrt{d_{k+1} d_k} \\ &+ \sum_{i=2}^{l-1} (A_\eta A_\Theta)^{l+1-i} \sqrt{d_i} \prod_{k=i}^{l-1} \sqrt{d_{k+1} d_k} + A_\eta A_\Theta \sqrt{d_l} \end{aligned}$$

under condition (3.14) and $R_l C_\eta \sqrt{d_l}$ under condition (3.13) for $l = 2, \ldots, L-1$, where $R_0 A_x$ and $R_1 A_\eta A_\Theta \sqrt{d_1 d_0} A_x + A_\eta A_\Theta \sqrt{d_1}$.

Lemma 3.3 is proved in Appendix G. A similar result is obtained for the function spaces $\mathcal{H} \circ \mathcal{F}^s$ and $\mathcal{H} \circ \mathcal{F}^t$ in the following lemma, which is proved in Appendix H.

Lemma 3.4. *Let Assumptions 3.1, 3.1, 3.1 hold. Then, the covering numbers of the function classes $\mathcal{H} \circ \mathcal{F}^s$ and $\mathcal{H} \circ \mathcal{F}^t$ are upper bounded as*

$$\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s) \leq \prod_{l=1}^{L} \left(\frac{2A_\Theta Q_L}{\epsilon} + 1 \right)^{d_l(d_{l-1}+1)}$$

$$\mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \epsilon, \mathfrak{d}^t) \leq \prod_{l=1}^{L} \left(\frac{2A_\Theta Q_L}{\epsilon} + 1 \right)^{d_l(d_{l-1}+1)}.$$

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 Appendix I. Combining Corollary 3.5 and Theorem 2.9, 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 Appendix J.

Theorem 3.6. *Consider a learning algorithm relying on the minimization of a loss function of the form (3.10) 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 2.2 and 2.3. Assume also that the network parameters, activation functions and the kernels satisfy Assumptions 3.1-3.1.*

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

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

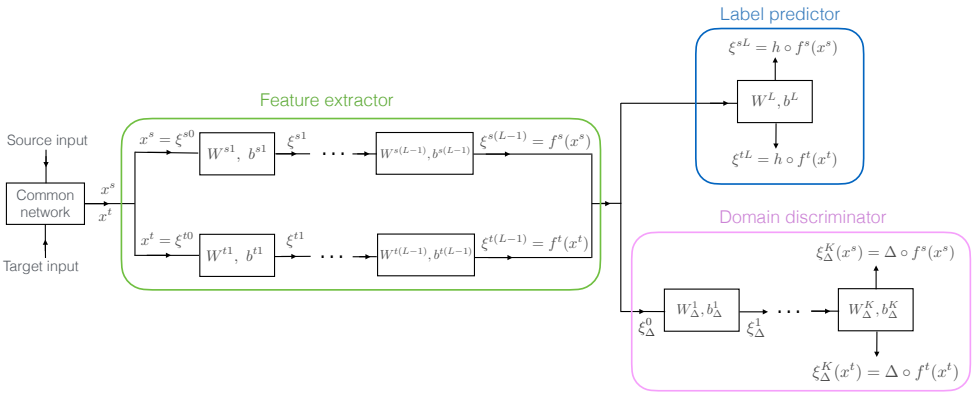

Figure 3. *Illustration of adversarial domain adaptation networks*

Note that the assumption of the existence of the constants A_ℓ and L_ℓ in Theorem 3.6 is satisfied in many common settings. In Appendix K, we derive these constants for the commonly used cross-entropy loss function. We can draw several conclusions from the statement of Theorem 3.6. The sample complexity expressions obtained in the theorem indicate that, as the network depth L and the network width d increase, M_s, N_s, and N_t must increase at rate $O(d^2L^2)$, if the logarithmic terms are ignored for simplicity. This result shows that the number of labeled source samples and the number all source and target samples required for preventing overfitting must grow quadratically with both L and d as the network size increases. On the other hand, the number M_t of available labeled target samples is typically limited in domain adaptation scenarios. Regarding this, Theorem 3.6 also has some implications on the optimal choice of the weight parameter α that finds a suitable balance between the target and source classification losses. As the number M_t of labeled target samples decreases, the weight α of the target classification loss must also shrink at rate $\alpha = O(\sqrt{M_t})$ in order to avoid overfitting the model to the few available target labels. Similarly, as the network size grows, the weight parameter α must also shrink at rate $\alpha = O((dL)^{-1})$ with d and L. The parameter ϵ in the theorem is a probability constant that sets the tradeoff between the desired accuracy level and the number of required training samples. In order for the expected target loss not to exceed the empirical losses by more than $O(\epsilon)$ in (3.16), the number of samples M_s, N_s, N_t must scale at an inverse quadratic rate $O(\epsilon^{-2})$ with ϵ.

3.2. Adversarial domain adaptation networks. In this section, we extend our results to analyze the sample complexity of adversarial domain adaptation networks. Adversarial models have been widely used in domain adaptation since the leading studies [27], [58], [40], and have been applied to a variety of problems in recent works [50]. 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 $h : \mathcal{X} \to \mathcal{Y}$ that provides the class label at its output as illustrated in Figure 3. The domain-invariance of the learnt features is ensured by a domain discriminator network, which is trained to determine whether the features belong to the source domain or the target domain. The feature extractor and the domain discriminator networks are trained in an adversarial fashion, such that the feature

extractor aims to learn domain-invariant representations whose domains are indistinguishable by the domain discriminator. The domain discriminator $\Delta : \mathcal{X} \to \mathbb{R}$ seeks to minimize the domain discrimination loss

$$\mathcal{L}^s_{\mathcal{D}}(f^s, \Delta) + \mathcal{L}^t_{\mathcal{D}}(f^t, \Delta)$$

where

$$\mathcal{L}^s_{\mathcal{D}}(f^s, \Delta) = E[\ell_{\mathcal{D}}(\Delta \circ f^s(x^s), l^s)], \qquad \mathcal{L}^t_{\mathcal{D}}(f^t, \Delta) = E[\ell_{\mathcal{D}}(\Delta \circ f^t(x^t), l^t)]$$

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 [27], [58], [40]

$$\begin{aligned} \ell_{\mathcal{D}}(\Delta \circ f^s(x^s), l^s) &= -\log(1 - \Delta \circ f^s(x^s)) \\ \ell_{\mathcal{D}}(\Delta \circ f^t(x^t), l^t) &= -\log(\Delta \circ f^t(x^t)). \end{aligned} \tag{3.17}$$

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

$$\min_{f^s, f^t, h, \Delta} (1-\alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha\hat{\mathcal{L}}^t(f^t, h) - \beta\big(\hat{\mathcal{L}}^s_{\mathcal{D}}(f^s, \Delta) + \hat{\mathcal{L}}^t_{\mathcal{D}}(f^t, \Delta)\big) \tag{3.18}$$

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

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

where l^s_i and l^t_j respectively denote the domain labels of the source samples x^s_i and the target samples x^t_j.

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

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

Similarly, the hypotheses $h \circ f^s$ and $h \circ f^t$ are given by the output of the last layer L

$$h \circ f^s(x^s) = \boldsymbol{\xi}^{sL}(x^s), \qquad h \circ f^t(x^t) = \boldsymbol{\xi}^{tL}(x^t)$$

with the function spaces $\mathcal{H}\circ\mathcal{F}^s$ and $\mathcal{H}\circ\mathcal{F}^t$ defined[1] 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. While feature extractor networks typically consist of several convolutional layers followed by fully connected layers in many common architectures [50]; 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 [58]. 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 domain discriminator network typically consists of several fully connected layers [27], [58]. 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 $l = 1, \dots, K$, 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

$$\Delta \circ f^s(x^s) = \boldsymbol{\xi}_\Delta^K(x^s), \qquad \Delta \circ f^t(x^t) = \boldsymbol{\xi}_\Delta^K(x^t)$$

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

$$\mathcal{D} = \{\Delta : \mathbb{R}^{d_{L-1}} \to \mathbb{R} \mid \Delta(\boldsymbol{\xi}_\Delta^0) = \boldsymbol{\xi}_\Delta^K,\ |(\mathbf{W}_\Delta^l)_{ij}| \le A_\Theta,\ |(\mathbf{b}_\Delta^l)_i| \le A_\Theta, \forall i, j\}. \tag{3.19}$$

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

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

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.

[1]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.

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.

The domain discriminator output is bounded, i.e., there exists a constant $C_{\mathcal{D}} > 0$ such that

$$|\Delta(\boldsymbol{\xi}_\Delta^0)| = |\boldsymbol{\xi}_\Delta^K| \leq C_{\mathcal{D}}$$

for all $\boldsymbol{\xi}_\Delta^0 \in \mathbb{R}^{d_{L-1}}$.

Note that Assumption 3.2 is satisfied for many domain-adversarial networks, as the activation function η_Δ^K of the final domain discriminator layer is often selected as a bounded function such as the sigmoid [27] or the softmax function [57]. Let us denote the composition of the domain discriminator and the feature extractor as

$$v^s(x^s)\Delta \circ f^s(x^s), \qquad v^t(x^t)\Delta \circ f^t(x^t)$$

and the corresponding function spaces as

$$\begin{aligned}\mathcal{V}^s &= \mathcal{D} \circ \mathcal{F}^s = \{v^s : v^s = \Delta \circ f^s, \Delta \in \mathcal{D}, f^s \in \mathcal{F}^s\} \\ \mathcal{V}^t &= \mathcal{D} \circ \mathcal{F}^t = \{v^t : v^t = \Delta \circ f^t, \Delta \in \mathcal{D}, f^t \in \mathcal{F}^t\}.\end{aligned}$$

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

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

Lemma 3.7. *Let Assumption 3.2 hold. Assume also that the composite function classes $\mathcal{V}^s$ and $\mathcal{V}^t$ are compact with respect to the metrics*

$$\begin{aligned}&\mathfrak{d}_{\mathcal{V}}^s(v_1^s, v_2^s) \sup_{x^s \in \mathcal{X}^s} |v_1^s(x^s) - v_2^s(x^s)| \\ &\mathfrak{d}_{\mathcal{V}}^t(v_1^t, v_2^t) \sup_{x^t \in \mathcal{X}^t} |v_1^t(x^t) - v_2^t(x^t)|\end{aligned}$$

where $v_1^s, v_2^s \in \mathcal{V}^s$ and $v_1^t, v_2^t \in \mathcal{V}^t$. Then,

$$\begin{aligned}&P\left(\sup_{f^s \in \mathcal{F}^s, f^t \in \mathcal{F}^t, \Delta \in \mathcal{D}} |D_\Delta(f^s, f^t) - \hat{D}_\Delta(f^s, f^t)| \leq \epsilon \right) \\ &\geq 1 - 2\mathcal{N}(\mathcal{V}^s, \frac{\epsilon}{6}, \mathfrak{d}_{\mathcal{V}}^s) \exp\left(-\frac{N_s \epsilon^2}{72 C_{\mathcal{D}}^2} \right) - 2\mathcal{N}(\mathcal{V}^t, \frac{\epsilon}{6}, \mathfrak{d}_{\mathcal{V}}^t) \exp\left(-\frac{N_t \epsilon^2}{72 C_{\mathcal{D}}^2} \right).\end{aligned}$$

The proof of Lemma 3.7 is presented in Appendix L. Note that Lemma 3.7 is the counterpart of Lemma 2.8 in the domain-adversarial setting. Before stating the main result of this section, we formalize the following conditions.

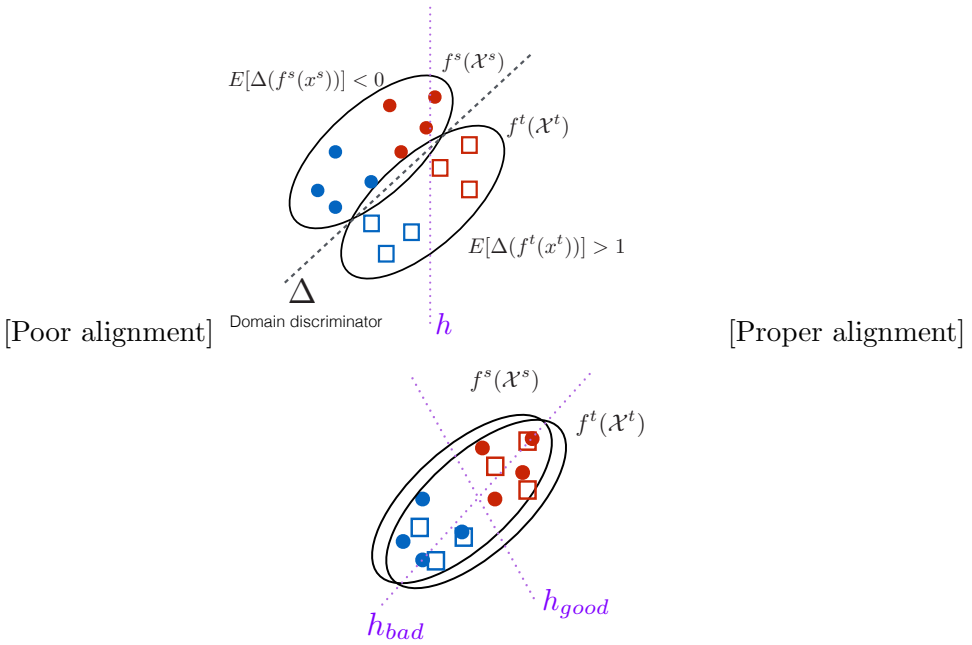

Figure 4. *Illustration of Assumption 3.2. Red and blue colors represent two different classes in the source and target domains. In (a), the two domains are poorly aligned by the mappings f^s and f^t, therefore, the algorithm learns a domain discriminator Δ that can separate the two domains well. The 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.*

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

$$\|\eta^l(\mathbf{u}) - \eta^l(\mathbf{v})\| \leq L_\eta \|\mathbf{u} - \mathbf{v}\| \tag{3.20}$$

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

$$\|\eta^l_\Delta(\mathbf{u}) - \eta^l_\Delta(\mathbf{v})\| \leq L_\eta \|\mathbf{u} - \mathbf{v}\| \tag{3.21}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d^\Delta_l}$, for $l = 1, \ldots, K$.

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

$$|(\eta^l_\Delta)_i(\mathbf{u})| \leq C_\eta \tag{3.22}$$

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

$$\|\eta^l_\Delta(\mathbf{u})\| \leq A_\eta \|\mathbf{u}\| \tag{3.23}$$

for all $\mathbf{u} \in \mathbb{R}^{d^\Delta_l}$.

Note that Assumption 3.2 is an adaptation of the conditions in Assumptions 3.1 and 3.1 to the domain-adversarial setting in consideration. Similarly, Assumption 3.2 simply adapts the condition in Assumption 3.1 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.

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

$$|\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)| \leq R_A \, D_\Delta(f^s, f^t) \tag{3.24}$$

for any transformations $f^s \in \mathcal{F}^s$, $f^t \in \mathcal{F}^t$, and any hypothesis $h \in \mathcal{H}$. Assumption 3.2 is the counterpart of Assumption 2.2 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. 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.

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

Theorem 3.8. *Consider a learning algorithm relying on the minimization of a loss function of the form (3.18) 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 3.2 and the network parameters and activation functions satisfy Assumptions 3.1 and 3.1- 3.2.*

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

$$\alpha = O\left(\left(\frac{M_t \epsilon^2}{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}\right)^{1/2}\right) \tag{3.25}$$

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

$$\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1-\alpha) R_A \hat{D}_\Delta(f^s, f^t) + (1-\alpha) R_A \epsilon + \epsilon, \tag{3.26}$$

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

$$M_s = O\left(\frac{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right)$$

$$N_s, N_t = O\left(\frac{d^2 (L+K) \log\left(\frac{L+K}{\epsilon}\right) + d^2 (L+K)^2 \log(d)}{\epsilon^2}\right).$$

The proof of Theorem 3.8 is presented in Appendix M. The findings of Theorem 3.8 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, similarly to Theorem 3.6. In order to prevent overfitting, M_s must increase at rate $M_s = O(d^2 L^2)$ 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 $O(\epsilon^{-2})$.

Remark 3.9. 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 (3.25) can similarly be obtained by replacing the number of layers L with $L+P$ in this case.

4. Discussion of the results in relation with previous literature. We now discuss our findings in relation with previous literature. To the best of our knowledge, our study is the first to propose an in-depth characterization of the sample complexity of domain-adaptive neural networks. A substantial body of work has focused on the effect of domain discrepancy on generalization performance, while another line of research has examined the sample complexity of neural networks, however, in a single-domain setting. We briefly overview these results below, along with a few relevant studies on the performance of domain alignment methods. For clarity and consistency, we restate the findings of prior work using our own notation. The presence of the parameter δ in the bounds signifies that the result holds with probability at least $1-\delta$.

4.1. Effect of domain discrepancy on generalization performance. One of the earliest analyses examining the effect of the deviation between the source and target distributions is the study by Ben-David et al. [8]. The gap between the expected target loss and the empirical source loss is shown to be bounded by

$$O\left(\sqrt{\frac{\dim_{VC}(\mathcal{H})}{M_s}+\log(\delta^{-1})}\right)+d_{\mathcal{H}}(D_S,D_T)+\lambda$$

ignoring the logarithmic factors, where $\dim_{VC}(\mathcal{H})$ denotes the VC-dimension of the hypothesis space $\mathcal{H}$, M_s is the number of of labeled source samples, and λ is a measure of the proximity of the true label function to the hypothesis class $\mathcal{H}$. Here $d_{\mathcal{H}}(D_S,D_T)$ is the $\mathcal{A}$-distance [8] between the source and target distributions D_S and D_T, given by

$$d_{\mathcal{H}}(D_S,D_T)=2\sup_{A\in\mathcal{A}}|P_{D_S}(A)-P_{D_T}(A)|$$

where $\mathcal{A}$ is the set of domain subsets with characteristic functions in $\mathcal{H}$, and $P_{(\cdot)}$ denotes probability with respect to a distribution.

In a succeeding study [7], this result has been extended to algorithms minimizing a convex combination of source and target losses, where the hypothesis that minimizes the empirical weighted loss is shown to generalize to the target domain within an error of

$$O\Bigg(\sqrt{\frac{\alpha^2}{\gamma}+\frac{(1-\alpha)^2}{1-\gamma}}\sqrt{\frac{\dim_{VC}(\mathcal{H})+\log(\delta^{-1})}{M}} + (1-\alpha)\left(\sqrt{\frac{\dim_{VC}(\mathcal{H})\log(\delta^{-1})}{N}}+\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(D_S,D_T)+\lambda\right)\Bigg).$$

Here the distribution distance $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(D_S,D_T)$ denotes the empirical divergence between the source and the target distributions over the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$, which corresponds to the set of disagreements [7]. $N=N_s=N_t$ denotes the number of all samples in the two domains, and M is the total number of labeled samples, with $M_s=(1-\gamma)M$ source samples and $M_t=\gamma M$ target samples. This result has some implications paralel to our study, in that the optimal weight α of the target loss should decrease with the scarcity of target labels, i.e., as γ decreases. A high domain discrepancy $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(D_S,D_T)$ also drives the weighted loss towards the target loss, by decreasing the weight $1-\alpha$ of the source loss.

Similar findings have been presented in the study of Mansour et al. in terms of the Rademacher complexities of the hypothesis space [41]. However, in [41] the deviation between the source and the target domains has been characterized in terms of the discrepancy $\mathrm{disc}_\ell(D_S,D_T)$, which quantifies how the loss-induced disagreement between any pair of hypotheses may differ across D_S and D_T.

Following these pioneering works, many other domain divergence measures have been proposed in succeeding studies [48]. Deng et al. have explored a robust variant of the discrepancy in [41] based on the adversarial Rademacher complexity definition [18], which has been shown to vary with the number of samples M and the network width d at rate $O(\sqrt{d/M})$ for

two-layer ReLU neural networks. Zhang et al. have proposed an alternative characterization of distribution distance based on the margin disparity discrepancy, leading to generalization bounds in terms of the Rademacher complexities and the covering numbers of hypothesis spaces [76]. Zellinger et al. have presented performance bounds depending on the VC-dimension of the function classes by formulating the domain discrepancy in terms of the difference between the moments of the source and target distributions [74]. Other recent efforts along this line include studies involving margin-aware risks with links to optimal transport distances [19], information-theoretic bounds based on mutual information [64, 69], hypothesis-specific divergence measures [65], and risk definitions based on stochastic predictors [49].

Remark 4.1. We note that all these aforementioned works assume that a common classifier is learnt in the original source and target domains; i.e., their setting is essentially different from ours as they do not at all consider learning a transformation or a mapping that aligns the two domains. The main distinction among these works lies in the specific distribution discrepancy each one proposes to characterize the misalignment between the domains, with the purpose of deriving tighter error bounds. Meanwhile, the reported labeled and unlabeled sample complexities, or otherwise the errors, follow the classical dependence on the VC-dimensions or the Rademacher complexities of the hypothesis classes in consideration, consistent with well-established results in learning theory. From the perspective of domain alignment algorithms, one may want to regard the domain discrepancies in these bounds as the distance obtained after mapping the two domains to a shared domain, an interpretation that arguably extends to transformation learning. While this view holds to some extent, many of the discrepancy measures used in these works (including their empirical approximations) are defined in a theoretical manner, and are difficult to estimate in practice. Although efficient computational techniques may exist for some of these discrepancy measures, they often lack accompanying learning guarantees. In contrast, our main results in Theorems 2.9-3.8 offer a practical means of assessing the generalization capability of domain alignment algorithms, as they are based on the empirical distribution distance computed directly on the aligned training data.

4.2. Performance bounds for domain alignment algorithms. To the best of our knowledge, a very limited number of theoretical analyses have investigated the performance of learning domain-aligning transformations or representations. A multi-task domain adaptation method is proposed in [77], which learns the similarity between source and target samples through a linear transformation $\mathbf{G}$. Assuming the incoherence of the projections corresponding to different tasks, the estimation error of the transformation $\mathbf{G}$ is shown to be bounded by $O(d_T\sqrt{\log(d_S)/n})$, where d_S and d_T denote the dimensions of the source and target Euclidean domains, and n is the number of tasks. While this bound is subsequently leveraged in [77] to design suitable classifiers based on the incoherence principle, the scope of their analysis is limited to linear transformations.

A performance analysis of conditional distribution matching is presented in [63], showing that the generalization gap in the target domain is bounded by

$$O\left(1+\frac{1}{\sqrt{M_t}}+\sqrt{\frac{\log(\delta^{-1})}{M_s+M_t}}\right)$$

when the source domain is mapped to the target domain through a location and scale transform.

Fang et al. have considered semi-supervised domain alignment algorithms as in our work [23]. However, their analysis is significantly different from ours since it does not explore the sample complexity of learning domain transformations, but instead treats the sample complexity as a known problem parameter. Their study aims to demonstrate that the need for labeled target data can be alleviated under certain assumptions by relying on the source and unlabeled target data.

Transferring representations from a source task to a target task is a problem different from but connected to domain adaptation. Wang et al. have provided an extensive analysis of transfer learning and multitask learning through domain-invariant feature representations by minimizing a combined empirical loss under regularization [66]. The performance gap between the source and target losses is shown to vary at rate

$$O\left(\mathrm{dist}_{\mathcal{Y}}(f^s, f^t) + \sqrt{\frac{\log(\delta^{-1})}{M_s + M_t}}\right).$$

Here $\mathrm{dist}_{\mathcal{Y}}(f^s, f^t)$ denotes the $\mathcal{Y}$-discrepancy [44] between the two domains once transformed to a shared domain, which is, however, not easy to estimate in practice.

Galanti et al. have modeled the transfer learning problem in a setting where a target task and multiple source tasks are drawn from the same distribution of distributions, and considered that a neural network architecture is partially transferred to the target task [25]. Their analysis implies that for accurate transfer, the number of source tasks and the number of samples per source task must scale with the number of edges, respectively, in the transferred component and the target-specific component of the network. In a recent work, Jiao et al. have considered a model that distinguishes between shared and domain-specific features in multi-domain deep transfer learning and shown that transferability between tasks improves the convergence rates in the target task [35]. McNamara and Balcan have investigated representation learning on a source task and fine-tuning on a target task [43]. The accuracy on the source task is shown to carry over to the target task within a performance gap of $O(\sqrt{\dim_{VC}(\mathcal{H} \circ \mathcal{F})/M_s} + \sqrt{\dim_{VC}(\mathcal{H})/M_t})$, where $\mathcal{F}$ is the space of feature representations and $\mathcal{H}$ is the space of classifiers. The significance of this result lies in the fact that the number M_t of labeled target samples should scale with the dimension of only the classifier $\mathcal{H}$, rather than the more complex composite hypothesis space $\mathcal{H} \circ \mathcal{F}$. A paralel finding is presented in [56] for the problem of transfer learning in a multi-task setting, demonstrating that the number of labeled samples for a new task needs to scale only with the complexity of its own task-specific map, assuming the abundance of the training data for the previous tasks.

Remark 4.2. Although our domain adaptation setting differs essentially from that considered in these transfer learning studies, they are comparable in their shared focus on handling the scarcity of labeled target samples. Whereas these works tie sample complexity to the richness of the target function class, which can be still large for deep neural networks, our analysis indicates that in a domain adaptation scenario the limitedness of target labels can be tolerated through strategically choosing the weight parameter as $\alpha = O(\sqrt{M_t})$, independently of the complexity of the target function class.

4.3. Sample complexity of neural networks in a single domain. Sample complexity of neural networks is a well-explored topic in statistical learning theory, a comprehensive overview of which can be found in [1], [6]. Although this classical line of research pertains to learning algorithms in a single domain and does not extend to domain adaptation scenarios, we find it instructive to briefly review these results and compare them to our bounds on domain adaptive neural networks.

The sample complexity of a feed-forward network consisting of W weights, L layers and s output units, with fixed piecewise-polynomial activation functions is reported as [1, Theorem 21.5]

$$O\left(\frac{s(WL\log(W)+WL^2)\log(\epsilon^{-1})+\log(\delta^{-1})}{\epsilon^2}\right) \tag{4.1}$$

in order to attain an error of ϵ. Denoting the network width as d, the number of weights W in an L-layer network is obtained as $W=d^2L$. Then, the sample complexity $M=O(d^2L^3)$ in (4.1) points to a quadratic dependence on d and a cubic dependence on L. This polynomial dependence is in line with our results in Theorems 3.6 and 3.8, where the sample complexity of labeled source data has been obtained as $M_s=O(d^2L^2)$. The dependence on L is quadratic, hence slightly tighter in our bounds.

A more recent trend in the exploration of sample complexity of neural networks is the characterization of the complexity in a dimension-independent way under particular assumptions. Neyshabur et al. have shown that the sample complexity depends exponentially on the network depth; nevertheless, its dependence on the network width can be removed under group norm regularization of network weights [46]. In succeeding studies, the exponential dependence on the network size has been reduced to polynomial [68], quadratic [45], linear [31] and logarithmic [5] factors. Harvey et al. have shown that the VC-dimension of neural networks with ReLU activation functions is $O(WL\log(W))$, resulting in comparable bounds to our work [33]. In some more recent works, it has been shown that the dependence on network width can be removed for one-layer networks [60] and reduced to logarithmic factors for two-layer networks [15] under bounded Frobenius norm and spectral norm constraints. We note that these results essentially rely on the condition that the norms of the weight matrices be upper bounded in a dimension-independent manner, and would translate to rather pessimistic sample complexities under the removal of this assumption.

Remark 4.3. While the above studies have contributed to a comprehensive understanding of neural network classifiers, they all focus on the single-domain scenario, assuming identical distributions for training and test data. To the best of our knowledge, our work is the first to provide a detailed analysis of the sample complexity of domain-adaptive neural networks. We note that our analysis does not impose any special constraints on the weight matrices, such as norm regularization. Under the incorporation of norm constraints, we would expect to arrive at tighter bounds consistently with the approaches in single-domain settings, which is left as a potential future direction of our study.

5. Experimental results. In this section, we present experimental results for the verification of the proposed generalization bounds. In Section 5.1, we study the generic bounds presented in Section 2 by considering a shallow (linear) classifier model. Then in Section 5.2,

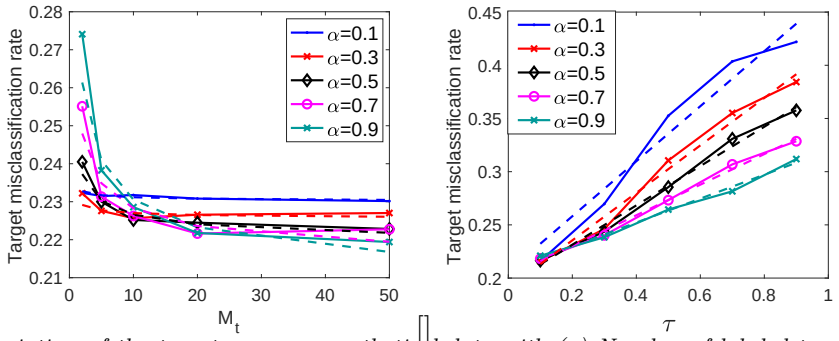

Figure 5. *Variation of the target error on synthetical data with (a) Number of labeled target samples, (b) Distribution distance after transformation. Solid lines indicate experimental data and dashed lines represent theoretical rates of variation.*

we examine the sample complexity results proposed in Section 3 for domain-adaptive neural networks.

5.1. General domain alignment methods. We first validate our findings in Section 2 on a synthetic data set with two classes. The source and target data sets are generated by applying two different geometric transformations to 400 samples drawn from the standard normal distribution in $\mathbb{R}^2$. We simulate a learning algorithm that learns geometric transformations to map the source and target samples to a common domain and then trains a classifier in the shared domain. Here we emulate a setting where the transformations f^s and f^t are treated as if learnt from data, however, with some error. In practice, f^s and f^t are formed by perturbing the ground truth geometric transformations with some transformation estimation error τ. We test a range of estimation error levels τ in the experiments. The classifier trained after mapping the samples to the common domain is chosen as a regularized ridge regression algorithm solving

$$\min_{\mathbf{w}\in\mathbb{R}^2} \quad \frac{1-\alpha}{M_s}\sum_{i=1}^{M_s}(\mathbf{w}^T f^s(x_i^s)-\mathbf{y}_i^s)^2+\frac{\alpha}{M_t}\sum_{j=1}^{M_t}(\mathbf{w}^T f^t(x_j^t)-\mathbf{y}_j^t)^2+\lambda\|\mathbf{w}\|^2.$$

The target misclassification rate is evaluated over 1000 test samples drawn from the target distribution and classified through the learnt hypothesis $\mathbf{w}$ and target transformation f^t.

In Figure 5, the variation of the target misclassification rate with the number M_t of labeled target samples is shown for different values of the weight α for the target loss. In order to interpret these results, it is helpful to recall our theoretical analysis in Section 2: Theorem 2.4 states that the expected target loss $\mathcal{L}^t(f^t,h)$ deviates from its reference value based on the empirical weighted loss $\hat{\mathcal{L}}_\alpha(f^s,f^t,h)$ and the distance $D(f^s,f^t)$ by an amount of ϵ. In order to achieve this with high and fixed probability, the term $M_t\epsilon^2$ in the probability expression (2.5) must be constant[2]. This implies that the expected target loss should decrease at rate $\epsilon = O(\sqrt{1/M_t})$ as M_t increases. Considering the target misclassification rate as an accurate approximation of the expected loss $\mathcal{L}^t(f^t,h)$ in Figure 5, we observe that the decay

[2]We ignore logarithmic factors and assume that the generic covering numbers in Theorem 2.4 grow at a typical geometric rate of increase as the covering radius decreases.

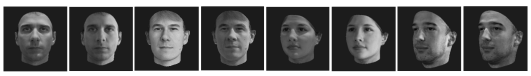

Figure 6. *Sample images from the MIT-CBCL face data set for four different subjects, rendered respectively under poses 1, 2, 5, and 9 for various illumination conditions.*

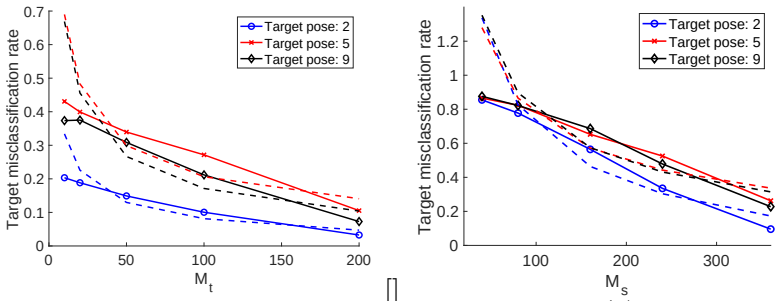

Figure 7. *Variation of the target error on MIT-CBCL face data with (a) Number of labeled target samples, (b) Number of labeled source samples. Solid lines indicate experimental data and dashed lines represent theoretical rates of variation.*

in the target error with M_t is consistent with Theorem 2.4. In particular, the dashed lines in the plots correspond to fitted theoretical rates of decay $O(\sqrt{1/M_t})$, which closely match the experimental data. We can also observe that large M_t values favor larger α values, while α must be chosen smaller at small M_t values. This also aligns with the conclusion drawn from Theorem 2.4 that the parameter α must be chosen as $\alpha = O(\sqrt{M_t})$ in order to control the term $e^{-\frac{M_t\epsilon^2}{8\alpha^2 A_\ell^2}}$ as M_t decreases.

We then study in Figure 5 the variation of the target misclassification rate with the estimation error τ of the geometric transformations. The parameter τ here is taken as the norm of the error matrix that is added to the ground truth transformation matrix. Hence, τ can be regarded as a parameter proportional to the distribution distance $D(f^s, f^t)$. The misclassification rate tends to increase with τ at an approximately linear rate, as confirmed by the dashed lines representing the theoretical linear rate of increase fitted to the experimental data. These results are coherent with the prediction of Theorem 2.4 that the expected target loss should increase proportionally to the distribution distance $D(f^s, f^t)$.

Next, we experiment on the MIT-CBCL image data set [42]. The data set consists of a total of 3240 synthetic face images belonging to 10 subjects. The images of each subject are rendered under 36 different illumination conditions and 9 poses, with Pose 1 corresponding to the frontal view and Pose 9 corresponding to a nearly profile view. Some example images from Poses 1, 2, 5, 9 are shown in Figure 6. We consider the images rendered under Pose 1 as the source domain, and repeat experiments by taking images from Poses 2, 5 and 9 as the target domain in each trial. First, using all labeled and unlabeled images, we compute a mapping between the source and target domains by the method proposed in [24], which finds a transformation that aligns the PCA bases of the source and target domains. We then train an SVM classifier using all labeled samples from the two domains. The unlabeled target samples are finally classified with the learnt transformation and classifier.

The misclassification rates of unlabeled target samples are plotted in Figures 7 and 7, with respect to the number of labeled target and source samples respectively. We observe that in both figures, the misclassification rates are reduced effectively with the increase in the number of labeled samples. As previously discussed, the target loss is expected to asymptotically reduce to an error component resulting from the empirical loss and the distribution distance, at rates $O(\sqrt{1/M_t})$ and $O(\sqrt{1/M_s})$ with increasing M_t and M_s. The experimental results in Figures 7 and 7 seem consistent with this expectation. The theoretical curves fitted to the experimental data with the expected rates of decrease are also indicated with dashed lines in the plots for visual comparison.

5.2. Domain-adaptive neural networks. We next aim to experimentally verify our results in Theorems 3.6 and 3.8 regarding the sample complexity of domain-adaptive neural networks. We present our results for MMD-based and adversarial domain adaptation networks, respectively in Section 5.2.1 and Section 5.2.2. For both architectures, our purpose is to experimentally characterize the sample complexity of the network with respect to the depth L and the width d of the network. We additionally investigate the optimal value of the weight α of the target loss in the objective function for both cases.

In our experiments, the MNIST handwritten digit data set [38] is used as the source data set, which consists of 60000 images. The target data set is taken as MNIST-M [26], which contains 59000 handwritten digit images with colored backgrounds. We train the neural networks with labeled and unlabeled training samples from the source and target domains, and then evaluate the target accuracy of the learnt models, defined as the correct classification rate of test samples from the target domain. In all experiments, algorithm hyperparameters and fixed variables are chosen to keep the neural network in the overfitting regime, enabling the characterization of the sample complexity of the models under consideration.

5.2.1. MMD-based domain adaptation networks. In our analysis of MMD-based domain adaptation networks, we consider the architecture proposed in the pioneering study [39] as our benchmark. We build on our previous experimental study [36] and employ a neural network structure similar to the baseline model in [39], beginning with convolutional layers and followed by several fully connected MMD layers. The MMD layer parameters are coupled between the source and target domains. The dimensions (widths) of all MMD layers are set as equal. Batch normalization is applied after each layer in order to stabilize the performance. We use the PyTorch implementation of the network available in [11] and adapt it for the minimization of the objective function

$$\frac{1-\alpha}{M_s}\sum_{i=1}^{M_s}\ell(h\circ f(x_i^s),\mathbf{y}_i^s)+\frac{\alpha}{M_t}\sum_{i=1}^{M_t}\ell(h\circ f(x_j^t),\mathbf{y}_j^t)+\beta\sum_{l=1}^{L-1}(\hat{D}^l)^2(f^l,f^l) \tag{5.1}$$

where $\ell(\cdot,\cdot)$ is set as the cross-entropy loss function and the source and target feature transformations are coupled as $f^s=f^t=f$ and $f^{sl}=f^{tl}=f^l$.

In Figure 8, we study the sample complexity of labeled source samples M_s and all source samples N_s with respect to the number L of MMD layers in the network. Figures 8 and 8 show the decrease in the target accuracy as the number L of MMD layers increases when the network is in the overfitting regime, for different M_s and N_s values. We aim to characterize

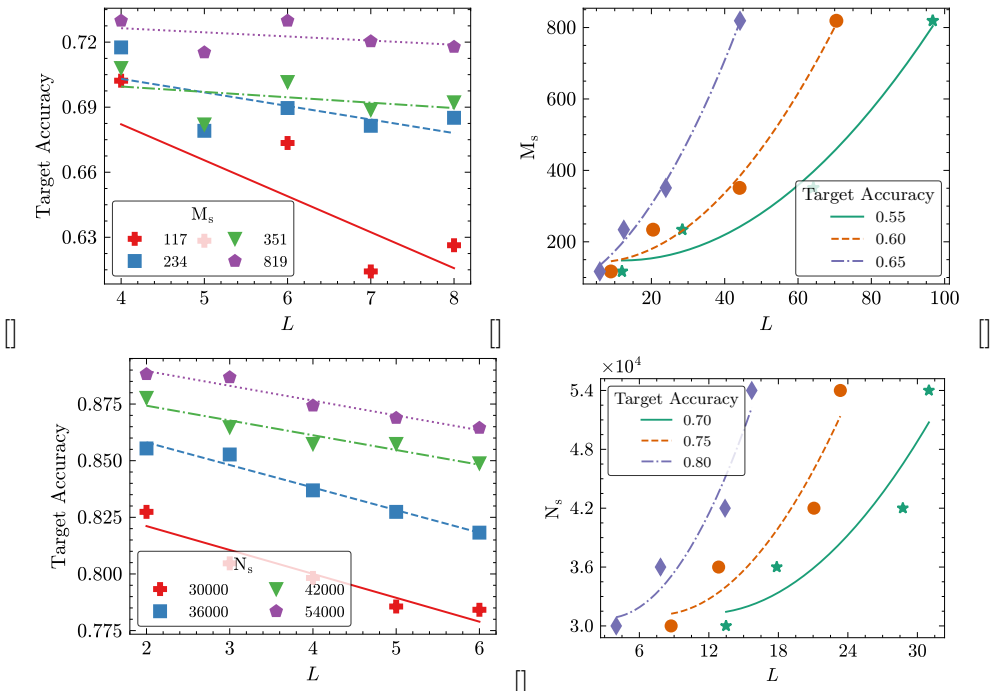

Figure 8. *Sample complexity of labeled samples (M_s) and all samples (N_s) with respect to the depth L of MMD-based domain adaptation networks. Left panels (a),(c): Variation of target accuracy with L. Right panels (b),(d): Variation of the number of samples (M_s, N_s) required for attaining a desired target accuracy level with L.*

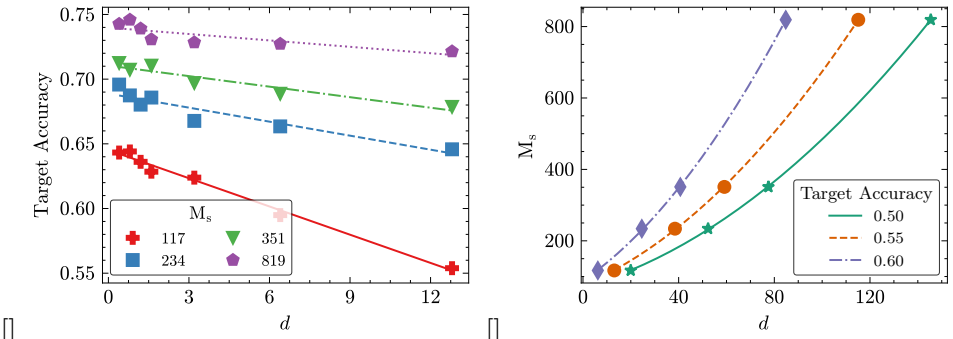

Figure 9. *Sample complexity of labeled samples (M_s) with respect to the width d of MMD-based domain adaptation networks. (a) Variation of target accuracy with d. (b) Variation of the number of samples (M_s) required for attaining a desired target accuracy level with d.*

the sample complexity of M_s and N_s with respect to L in this experiment. Therefore, we determine several desired target accuracy levels for the results in Figures 8 and 8, and identify the smallest M_s and N_s values that ensure this target accuracy as L grows[3], which are plotted respectively in Figures 8 and 8. We recall from Theorem 3.6 that the sample complexities of M_s and N_s are expected to grow at quadratic rates $M_s = O(L^2)$ and $N_s = O(L^2)$ as

[3]In cases where obtaining the exact value of L exceeded our computational resources, we resorted to linear extrapolation of the curves in Figures 8 and 8 to approximately infer the corresponding L value.

the network depth L increases. The experimental findings in Figures 8 and 8 confirm this prediction, as the increase in the required sample size for attaining a reference target accuracy level indeed follows a quadratic increase with L. The curves in 8 and 8 are obtained by fitting quadratic polynomials to the experimental data for visual evaluation.

A similar experiment is conducted in Figure 9, where the sample complexity is studied with respect to the network width this time. The parameter d in Figures 9 and 9 represent the factor by which the network width in the original implementation [11] is multiplied in our experiment. Hence, d is directly proportional to the shared width parameter of the MMD layers. The results in 9 are also consistent with the theoretical findings in Theorem 3.6, which states that the sample complexity must increase at a quadratic rate $M_s = O(d^2)$ as the network width increases.

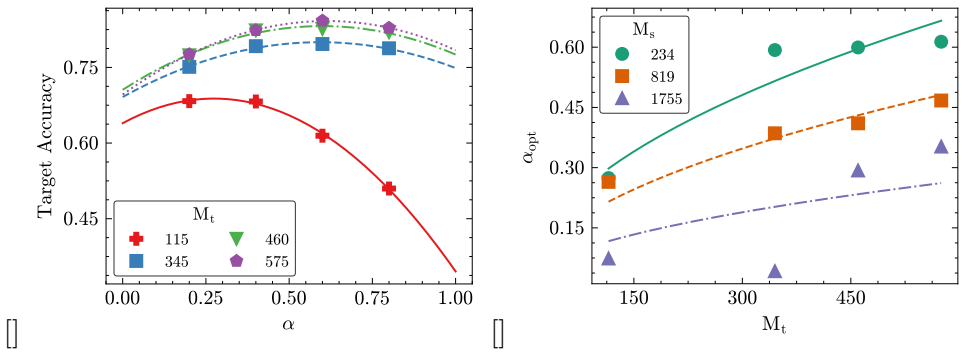

Figure 10. *(a) Variation of target accuracy with target loss weight parameter α for MMD-based domain adaptation networks (obtained at $M_s = 234$). (b) Variation of optimal weight α_{opt} with number of labeled target samples M_t.*

We also recall from Theorem 3.6 that, in order to maximize the target accuracy, the weight parameter α of the target classification loss must scale as $\alpha = O(\sqrt{M_t})$ as the number M_t of labeled target samples varies. We experimentally validate this result in Figure 10. In Figure 10, we examine the variation of the target accuracy with the weight parameter α. Here, the target accuracy follows a non-monotonic variation with α as expected. We approximately identify the optimal value α_{opt} of the weight parameter for each value of M_t by applying polynomial fitting to the plots in Figure 10. The variation of the optimal weight α_{opt} with M_t is then plotted in Figure 10. In order to visually observe the prediction of Theorem 3.6, we also fit a curve of $O(\sqrt{M_t})$ to each data sequence in Figure 10. The experimental data in Figure 10 seems consistent with the fitted curves, which supports the statement of Theorem 3.6 that the optimal weight parameter must scale at rate $\alpha_{opt} = O(\sqrt{M_t})$.

5.2.2. Adversarial domain adaptation networks. In order to experimentally evaluate our findings in Section 3.2, we adopt the model proposed in [27], which is a well-known representative of adversarial domain adaptation architectures. We use the PyTorch implementation of this model available in [30], by adapting it to the semi-supervised setting studied in our

analysis. We train the adversarial network to minimize the objective function

$$\frac{1-\alpha}{M_s}\sum_{i=1}^{M_s}\ell(h\circ f(x_i^s),\mathbf{y}_i^s)+\frac{\alpha}{M_t}\sum_{i=1}^{M_t}\ell(h\circ f(x_j^t),\mathbf{y}_j^t)$$
$$-\frac{\beta}{N_s+N_t}\left(\sum_{i=1}^{N_s}\ell_{\mathcal{D}}(\Delta\circ f(x_i^s),l_i^s)+\sum_{j=1}^{N_t}\ell_{\mathcal{D}}(\Delta\circ f(x_j^t),l_j^t)\right)$$

where the label loss $\ell(\cdot,\cdot)$ and the domain discrimator loss $\ell_{\mathcal{D}}(\cdot,\cdot)$ are selected as the negative log likelihood function, and the source and target feature extractor networks are coupled as $f^s=f^t=f$.

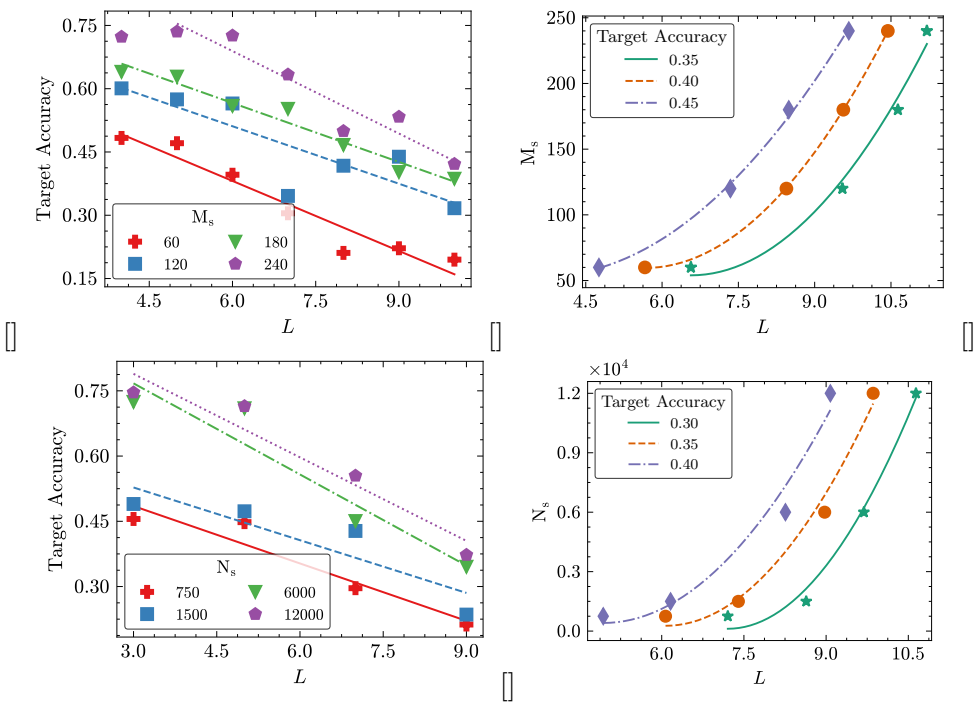

Figure 11. *Sample complexity of labeled samples (M_s) and all samples (N_s) with respect to the depth L of adversarial domain adaptation networks. Left panels (a),(c): Variation of target accuracy with L. Right panels (b),(d): Variation of the number of samples (M_s, N_s) required for attaining a desired target accuracy level with L.*

The feature extractor network contains only convolutional layers, while the label predictor and domain discriminator networks consist of fully connected layers in the implementation in [30]. In order to adapt our experiments to this structure, when analyzing the sample complexity of labeled data (M_s), we set the number of layers in the feature extractor and label predictor networks as equal, which is represented by the parameter L. Likewise, when studying the sample complexity of all data (N_s), the number of layers in the feature extractor and domain discriminator networks are equated and denoted as L. We use a similar strategy to adjust the network width, where we scale the number of convolutional channels and the

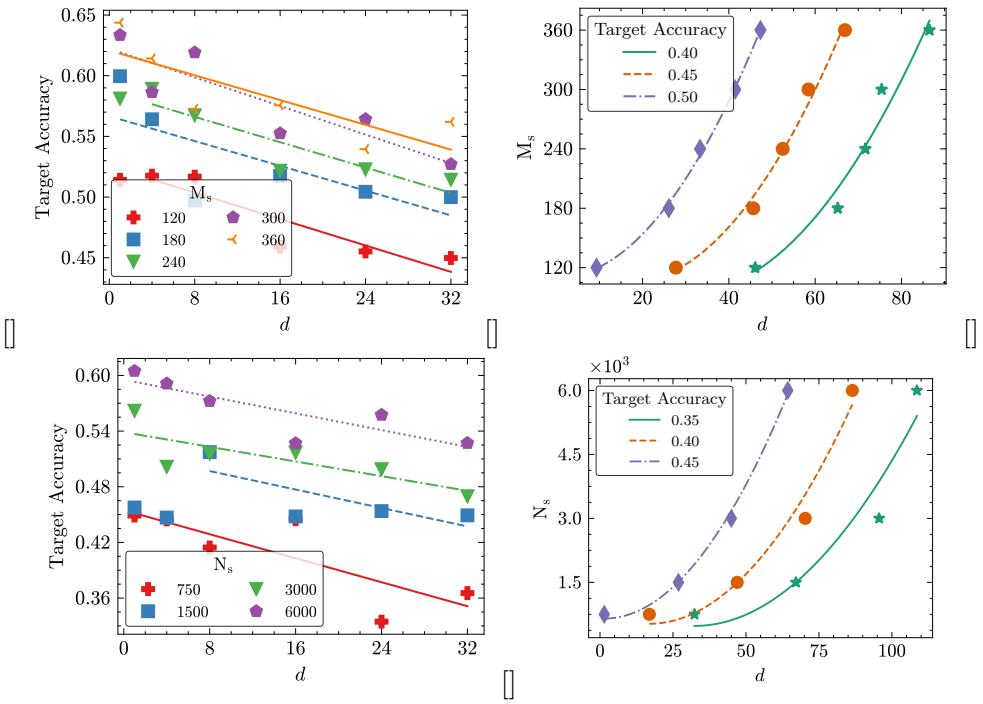

Figure 12. *Sample complexity of labeled samples (M_s) and all samples (N_s) with respect to the width d of adversarial domain adaptation networks. Left panels (a),(c): Variation of target accuracy with d. Right panels (b),(d): Variation of the number of samples (M_s, N_s) required for attaining a desired target accuracy level with d.*

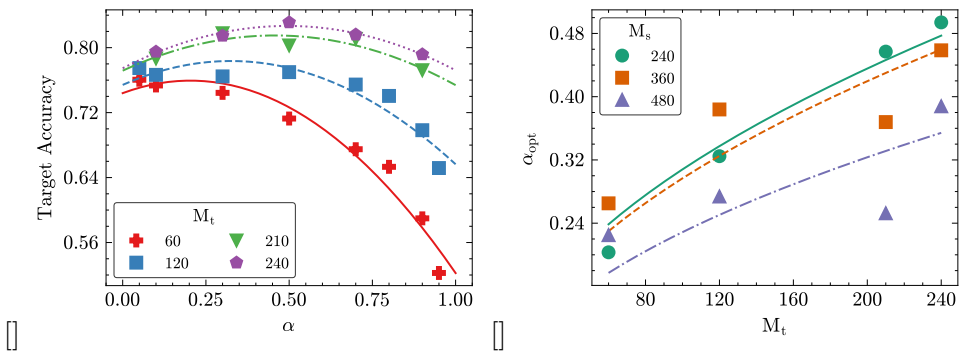

Figure 13. *(a) Variation of target accuracy with target loss weight parameter α for adversarial domain adaptation networks (obtained at $M_s = 240$). (b) Variation of optimal weight α_{opt} with number of labeled target samples M_t.*

fully connected layer width in the original paper [27] with the same factor d. Hence, the number of convolutional channels is scaled proportionally to the width of the label predictor and the domain discriminator networks, respectively, when studying the sample complexities of M_s and N_s. Batch normalization and ReLU layers are included after each convolutional or fully connected layer, following standard practice.

The sample complexities of the number of source samples with the network depth L and width d are presented, respectively in Figures 11 and 12. Similarly to the experiments in Section 5.2.1, left panels (a) and (c) show the variation of the target accuracy with L or d at different M_s and N_s values. The plots in the right panels (b) and (d) are then obtained by investigating the smallest M_s and N_s values ensuring a reference target accuracy level as L or d increases. The results of these experiments align with the theoretical bounds in Theorem 3.8, confirming the quadratic growth in the sample complexities $M_s, N_s = O(L^2)$ and $M_s, N_s = O(d^2)$ as the network depth L and width d increase.

We lastly study the choice of the parameter α weighting the target classification loss in the objective function for the adversarial setting. The results presented in Figure 13 confirm the theoretical prediction that the optimal value of the weight parameter should scale at rate $\alpha_{opt} = O(\sqrt{M_t})$ as the number of labeled samples varies.

Overall, our experimental findings in Section 5.2 are in line with the theoretical bounds presented in Theorems 3.6 and 3.8, supporting our sample complexity and optimal weight choice analyses for both MMD-based and adversarial domain adaptation networks.

6. 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.

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 5.2.1.

Appendix A. Proof of Lemma 2.3.

Proof. We characterize the complexity of function spaces via covering numbers [13]. We first derive a bound for the deviation between the expected and empirical target losses. Let

the open balls of radius $\frac{\epsilon}{8\alpha L_\ell}$ around the functions $\{g_k^t\}_{k=1}^{\kappa^t}$ be a cover for the function space $\mathcal{H} \circ \mathcal{F}^t$ with covering number

$$\kappa^t = \mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t).$$

Take any $g_k^t = h_k \circ f_k^t$, for $k = 1, \ldots, \kappa^t$. The random variables $\ell(g_k^t(x_j^t), \mathbf{y}_j^t)$, $j = 1, \ldots, M_t$ are independent identically distributed, bounded as $|\ell(g_k^t(x_j^t), \mathbf{y}_j^t)| \leq A_\ell$, and they have mean $\mathcal{L}^t(f_k^t, h_k)$. From Hoeffding's inequality, we get that for each k, the deviation between the empirical loss and the expected loss is bounded as

$$P\left(|\hat{\mathcal{L}}^t(f_k^t, h_k) - \mathcal{L}^t(f_k^t, h_k)| \geq \frac{\epsilon}{4\alpha}\right) \leq 2e^{-\frac{M_t\epsilon^2}{8\alpha^2 A_\ell^2}}.$$

Then, from union bound, with probability at least $1 - 2\kappa^t e^{-\frac{M_t\epsilon^2}{8\alpha^2 A_\ell^2}}$, the inequality

$$|\hat{\mathcal{L}}^t(f_k^t, h_k) - \mathcal{L}^t(f_k^t, h_k)| \leq \frac{\epsilon}{4\alpha}$$

holds for all $k = 1, \ldots, \kappa^t$. Now for any $g^t = h \circ f^t \in \mathcal{H} \circ \mathcal{F}^t$, there exists at least one g_k^t such that

$$\mathfrak{d}^t(g^t, g_k^t) < \frac{\epsilon}{8\alpha L_\ell}.$$

This gives

$$\begin{aligned}
&|\mathcal{L}^t(f^t, h) - \mathcal{L}^t(f_k^t, h_k)| = \left|\int_{\mathcal{Z}^t} \left(\ell(g^t(x^t), \mathbf{y}^t) - \ell(g_k^t(x^t), \mathbf{y}^t)\right) d\mu_t\right| \\
&\leq \int_{\mathcal{Z}^t} \left|\ell(g^t(x^t), \mathbf{y}^t) - \ell(g_k^t(x^t), \mathbf{y}^t)\right| d\mu_t \leq \int_{\mathcal{Z}^t} L_\ell \|g^t(x^t) - g_k^t(x^t)\| \, d\mu_t \\
&\leq L_\ell \int_{\mathcal{Z}^t} \mathfrak{d}^t(g^t, g_k^t) \, d\mu_t < \frac{\epsilon}{8\alpha}.
\end{aligned}$$

It is easy to show similarly that

$$|\hat{\mathcal{L}}^t(f^t, h) - \hat{\mathcal{L}}^t(f_k^t, h_k)| < \frac{\epsilon}{8\alpha}.$$

Then with probability at least

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

for any $g^t = h \circ f^t \in \mathcal{H} \circ \mathcal{F}^t$ we have

$$\begin{aligned}
&|\mathcal{L}^t(f^t, h) - \hat{\mathcal{L}}^t(f^t, h)| \\
&\leq |\mathcal{L}^t(f^t, h) - \mathcal{L}^t(f_k^t, h_k)| + |\mathcal{L}^t(f_k^t, h_k) - \hat{\mathcal{L}}^t(f_k^t, h_k) + |\hat{\mathcal{L}}^t(f_k^t, h_k) - \hat{\mathcal{L}}^t(f^t, h)| \\
&< \frac{\epsilon}{8\alpha} + \frac{\epsilon}{4\alpha} + \frac{\epsilon}{8\alpha} = \frac{\epsilon}{2\alpha}.
\end{aligned}$$

Replacing α with $1-\alpha$ and applying the same steps for the function space $\mathcal{H}\circ\mathcal{F}^s$, we similarly obtain that with probability at least

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

the difference between the expected and empirical source losses is bounded for any f^s and h as

$$|\mathcal{L}^s(f^s,h)-\hat{\mathcal{L}}^s(f^s,h)|<\frac{\epsilon}{2(1-\alpha)}.$$

Combining these results, we get that with probability at least

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

the largest difference between the expected and empirical total weighted losses is bounded as

$$\begin{aligned}&\sup_{f^s\in\mathcal{F}^s,f^t\in\mathcal{F}^t,h\in\mathcal{H}}|\mathcal{L}_\alpha(f^s,f^t,h)-\hat{\mathcal{L}}_\alpha(f^s,f^t,h)|\\&\qquad\le\alpha\sup|\mathcal{L}^t(f^t,h)-\hat{\mathcal{L}}^t(f^t,h)|+(1-\alpha)\sup|\mathcal{L}^s(f^s,h)-\hat{\mathcal{L}}^s(f^s,h)|\\&\qquad\le\epsilon.\end{aligned}$$

Appendix B. Proof of Lemma 2.7.

Proof. Our proof is based on the following result by Yurinskii [73].

Theorem B.1. *[73, Theorem 2.1] Let $\zeta_1,\dots,\zeta_N\in\mathcal{B}$ be independent random vectors, where $\mathcal{B}$ is a Banach space. Assume for all $i=1,\dots,N$*

$$E[\|\zeta_i\|^k]\le\frac{k!}{2}\,b_i^2\,C^{k-2},\ \text{for } k=2,3,\cdots. \tag{B.1}$$

If $x>\beta_N/B_N$ where

$$\beta_N\ge E[\|\zeta_1+\cdots+\zeta_N\|],\quad B_N^2=b_1^2+\cdots+b_N^2, \tag{B.2}$$

then

$$P\left(\|\zeta_1+\cdots+\zeta_N\|\ge xB_N\right)\le\exp\left(-\frac{1}{8}\left(x-\frac{\beta_N}{B_N}\right)^2\frac{1}{1+\left(x-\frac{\beta_N}{B_N}\right)\frac{C}{2B_N}}\right).$$

Based on Theorem B.1, we first derive the stated result for the source domain, whose generalization to the target domain is straightforward. First notice that, due to the assumptions (2.9), (2.10), the random vectors $f^s(x_i^s)-E[f^s(x^s)]$ for $i=1,\dots,N_s$ satisy the condition (B.1), for the choices $b_i=\sigma_s$ and $C=C_s$.

Next, we derive a constant β_{N_s} for which the zero-mean random vectors $\zeta_i = f^s(x_i^s) - E[f^s(x^s)]$ for $i = 1, \ldots, N_s$ satisfy the condition (B.2) for $N = N_s$. From (2.9), we have

$$E[\|\zeta_i\|^2] \leq \sigma_s^2.$$

We consider now

$$E\left[\left\|\sum_{i=1}^{N_s} \zeta_i\right\|^2\right] = E\left[\left\langle \sum_{i=1}^{N_s} \zeta_i, \sum_{j=1}^{N_s} \zeta_j \right\rangle\right] = \sum_{i=1}^{N_s}\sum_{j=1}^{N_s} E[\langle \zeta_i, \zeta_j\rangle]$$
$$= \sum_{i=1}^{N_s} E[\langle \zeta_i, \zeta_i\rangle] + \sum_{i=1}^{N_s}\sum_{j\neq i,\, j=1}^{N_s} E[\langle \zeta_i, \zeta_j\rangle] \leq \sigma_s^2 N_s$$

where the last inequality follows from $E[\|\zeta_i\|^2] \leq \sigma_s^2$, and the fact that we have $E[\langle \zeta_i, \zeta_j\rangle] = 0$ for independent and zero-mean ζ_i and ζ_j for $i \neq j$. From the nonnegativity of the variance, we have $(E[Y])^2 \leq E[Y^2]$ for any random variable Y. Taking

$$Y = \left\|\sum_{i=1}^{N_s} \zeta_i\right\|$$

then yields

$$E\left[\left\|\sum_{i=1}^{N_s} \zeta_i\right\|\right] \leq \left(E\left[\left\|\sum_{i=1}^{N_s} \zeta_i\right\|^2\right]\right)^{1/2} \leq \sigma_s\sqrt{N_s}.$$

Hence defining $\beta_{N_s} = \sigma_s\sqrt{N_s}$, we get

$$E[\|\zeta_1 + \cdots + \zeta_{N_s}\|] \leq \beta_{N_s}. \tag{B.3}$$

From the choice $b_i = \sigma_s$, we have $B_{N_s} = \sqrt{N_s}\sigma_s = \beta_{N_s}$. Now for given $\epsilon > 0$, from the assumption $N_s > \sigma_s^2/\epsilon^2$, the following choice for x

$$x = \frac{\sqrt{N_s}\epsilon}{\sigma_s} > 1$$

satisfies the condition $x > \beta_{N_s}/B_{N_s}$ as $\beta_{N_s} = B_{N_s}$. Then from Theorem B.1, we have

$$P\left(\|\zeta_1 + \cdots + \zeta_{N_s}\| \geq N_s\epsilon\right) \leq \exp\left(-\frac{1}{8}\left(\frac{\sqrt{N_s}\epsilon}{\sigma_s} - 1\right)^2 \frac{1}{1 + \left(\frac{\sqrt{N_s}\epsilon}{\sigma_s} - 1\right)\frac{C_s}{2\sqrt{N_s}\sigma_s}}\right).$$

Replacing $\zeta_i = f^s(x_i^s) - E[f^s(x^s)]$ gives the stated result

$$P\left(\left\|\frac{1}{N_s}\sum_{i=1}^{N_s} f^s(x_i^s) - E[f^s(x^s)]\right\| \geq \epsilon\right)$$
$$\leq \exp\left(-\frac{1}{8}\left(\frac{\sqrt{N_s}\epsilon}{\sigma_s} - 1\right)^2 \frac{1}{1 + \left(\frac{\sqrt{N_s}\epsilon}{\sigma_s} - 1\right)\frac{C_s}{2\sqrt{N_s}\sigma_s}}\right).$$

Applying the same analysis for the target domain, it is easy to show similarly that the upper bound for the target domain in (2.12) also holds. ∎

Appendix C. Proof of Lemma 2.8.

Proof. We begin with bounding the deviation $|D(f^s, f^t) - \hat{D}(f^s, f^t)|$ between the MMD and its empirical estimate for a fixed pair of transformations. Let f^s and f^t be a given, fixed pair of transformations. We have

$$\begin{aligned} &|D(f^s, f^t) - \hat{D}(f^s, f^t)| \\ &= \left| \|E[f^s(x^s)] - E[f^t(x^t)]\| - \Big\| \frac{1}{N_s}\sum_{i=1}^{N_s} f^s(x_i^s) - \frac{1}{N_t}\sum_{j=1}^{N_t} f^t(x_j^t) \Big\| \right| \\ &\leq \Big\| \frac{1}{N_s}\sum_{i=1}^{N_s} f^s(x_i^s) - E[f^s(x^s)] \Big\| + \Big\| \frac{1}{N_t}\sum_{j=1}^{N_t} f^t(x_j^t) - E[f^t(x^t)] \Big\|. \end{aligned} \tag{C.1}$$

Replacing ϵ by $\epsilon/4$ in Lemma 2.7, we observe that with probability at least

$$1 - \exp(-a_s(N_s, \epsilon)) - \exp(-a_t(N_t, \epsilon))$$

we have

$$\Big\| \frac{1}{N_s}\sum_{i=1}^{N_s} f^s(x_i^s) - E[f^s(x^s)] \Big\| \leq \frac{\epsilon}{4}, \quad \Big\| \frac{1}{N_t}\sum_{j=1}^{N_t} f^t(x_j^t) - E[f^t(x^t)] \Big\| \leq \frac{\epsilon}{4}$$

which yields from (C.1)

$$|D(f^s, f^t) - \hat{D}(f^s, f^t)| \leq \frac{\epsilon}{2}.$$

In order to extend the above bound to the whole space of transformations, we consider covers of the function classes $\mathcal{F}^s$ and $\mathcal{F}^t$, consisting of open balls of radius $\epsilon/8$ respectively around the functions $\{f_k^s\}_{k=1}^{\kappa^s}$ and $\{f_l^t\}_{l=1}^{\kappa^t}$, where κ^s and κ^t are the covering numbers

$$\kappa^s = \mathcal{N}(\mathcal{F}^s, \frac{\epsilon}{8}, \mathfrak{d}_{\mathcal{X}}^s), \quad \kappa^t = \mathcal{N}(\mathcal{F}^t, \frac{\epsilon}{8}, \mathfrak{d}_{\mathcal{X}}^t).$$

From the union bound, it follows that with probability at least

$$1 - \kappa^s \exp(-a_s(N_s, \epsilon)) - \kappa^t \exp(-a_t(N_t, \epsilon))$$

for all $k = 1, \ldots, \kappa^s$ and $l = 1, \ldots, \kappa^t$,

$$|D(f_k^s, f_l^t) - \hat{D}(f_k^s, f_l^t)| \leq \frac{\epsilon}{2}. \tag{C.2}$$

Now, let us consider an arbitrary pair of transformations $f^s \in \mathcal{F}^s$ and $f^t \in \mathcal{F}^t$. As the balls around $\{f_k^s\}_{k=1}^{\kappa^s}$ and $\{f_l^t\}_{l=1}^{\kappa^t}$ form $\epsilon/8$-covers of the function classes, there exists a source transformation f_k^s and a target transformation f_l^t such that

$$\mathfrak{d}_{\mathcal{X}}^s(f^s, f_k^s) < \frac{\epsilon}{8}, \quad \mathfrak{d}_{\mathcal{X}}^t(f^t, f_l^t) < \frac{\epsilon}{8}.$$

We can then bound the difference between the MMD and its sample mean for f^s and f^t as follows.

$$|D(f^s, f^t) - \hat{D}(f^s, f^t)| \leq |D(f^s, f^t) - D(f^s_k, f^t_l)| + |D(f^s_k, f^t_l) - \hat{D}(f^s_k, f^t_l)| + |\hat{D}(f^s_k, f^t_l) - \hat{D}(f^s, f^t)| \tag{C.3}$$

Next, we bound each one of the terms on the right hand side of the above inequality. The first term can be upper bounded as

$$\begin{aligned} |D(f^s, f^t) - D(f^s_k, f^t_l)| &= \big| \|E[f^s(x^s)] - E[f^t(x^t)]\| - \|E[f^s_k(x^s)] - E[f^t_l(x^t)]\| \big| \\ &\leq \|E[f^s(x^s)] - E[f^s_k(x^s)]\| + \|E[f^t(x^t)] - E[f^t_l(x^t)]\| \\ &= \|E[f^s(x^s) - f^s_k(x^s)]\| + \|E[f^t(x^t) - f^t_l(x^t)]\| \\ &\leq E[\|f^s(x^s) - f^s_k(x^s)\|] + E[\|f^t(x^t) - f^t_l(x^t)\|] \end{aligned} \tag{C.4}$$

where the last inequality follows from Jensen's inequality, observing the fact that a norm over a Hilbert space is a convex function. From the definition of the metrics $\mathfrak{d}^s_{\mathcal{X}}$ and $\mathfrak{d}^t_{\mathcal{X}}$, we have

$$\begin{aligned} \|f^s(x^s) - f^s_k(x^s)\| &\leq \mathfrak{d}^s_{\mathcal{X}}(f^s, f^s_k) \\ \|f^t(x^t) - f^t_l(x^t)\| &\leq \mathfrak{d}^t_{\mathcal{X}}(f^t, f^t_l) \end{aligned}$$

for all $x^s \in \mathcal{X}^s$ and $x^t \in \mathcal{X}^t$. Using this in (C.4), we get

$$|D(f^s, f^t) - D(f^s_k, f^t_l)| \leq \mathfrak{d}^s_{\mathcal{X}}(f^s, f^s_k) + \mathfrak{d}^t_{\mathcal{X}}(f^t, f^t_l) < \frac{\epsilon}{8} + \frac{\epsilon}{8} = \frac{\epsilon}{4}.$$

With a similar analysis by replacing the expectations with the sample means, it is easy to show that the third term in the inequality (C.3) can also be upper bounded as

$$|\hat{D}(f^s_k, f^t_l) - \hat{D}(f^s, f^t)| < \frac{\epsilon}{4}.$$

Now, remembering also the probabilistic upper bound (C.2) that holds for the second term in (C.3) for all k and l, we get that with probability at least

$$1 - \kappa^s \exp(-a_s(N_s, \epsilon)) - \kappa^t \exp(-a_t(N_t, \epsilon))$$

we have for all $f^s \in \mathcal{F}^s$ and $f^t \in \mathcal{F}^t$,

$$|D(f^s, f^t) - \hat{D}(f^s, f^t)| < \frac{\epsilon}{4} + \frac{\epsilon}{2} + \frac{\epsilon}{4} = \epsilon.$$

Hence, we get the stated result

$$\begin{aligned} &P\left(\sup_{f^s \in \mathcal{F}^s, f^t \in \mathcal{F}^t} |D(f^s, f^t) - \hat{D}(f^s, f^t)| < \epsilon \right) \\ &\geq 1 - \kappa^s \exp(-a_s(N_s, \epsilon)) - \kappa^t \exp(-a_t(N_t, \epsilon)) \\ &= 1 - \mathcal{N}(\mathcal{F}^s, \frac{\epsilon}{8}, \mathfrak{d}^s_{\mathcal{X}}) \exp(-a_s(N_s, \epsilon)) - \mathcal{N}(\mathcal{F}^t, \frac{\epsilon}{8}, \mathfrak{d}^t_{\mathcal{X}}) \exp(-a_t(N_t, \epsilon)). \end{aligned}$$

Appendix D. Proof of Lemma 3.1.

Proof. We prove the statements only for the source domain, as the proofs for the target domain are the same. Let $\boldsymbol{\xi}^{sl}(x^s) \in \mathbb{R}^{d_l}$ denote the feature in layer l for the source input $x^s \in \mathbb{R}^{d_0}$, where we regard $\boldsymbol{\xi}^{sl}(\cdot) : \mathbb{R}^{d_0} \to \mathbb{R}^{d_l}$ as a function. In the relation

$$\boldsymbol{\xi}^{sl}(x^s) = \eta^l(\mathbf{W}^{sl}\boldsymbol{\xi}^{s(l-1)}(x^s) + \mathbf{b}^{sl})$$

the expression $\mathbf{W}^{sl}\boldsymbol{\xi}^{s(l-1)}(x^s) + \mathbf{b}^{sl}$ is a continuous mapping of $\boldsymbol{\xi}^{s(l-1)}(x^s)$, and the function η^l is continuous. Hence, based on a simple induction argument it follows that $\boldsymbol{\xi}^{sl}(\cdot) : \mathcal{X}^s = \mathbb{R}^{d_0} \to \mathbb{R}^{d_l}$ is a continuous, thus measurable function (a Borel map).

We next show that the mappings $f^{sl} : \mathcal{X}^s \to \mathcal{X}^l$ are measurable. Let $\mathcal{B}(\cdot)$ denote the Borel σ-algebra of a metric space. We recall from (3.4) that $f^{sl}(x^s) = \phi^l(\boldsymbol{\xi}^{sl}(x^s)) \in \mathcal{X}^l$. Consider a sequence $\{\boldsymbol{\xi}_n^{sl}\} \subset \mathbb{R}^{d_l}$ with $\lim_{n\to\infty} \boldsymbol{\xi}_n^{sl} = \boldsymbol{\xi}_*^{sl}$ for some $\boldsymbol{\xi}_*^{sl} \in \mathbb{R}^{d_l}$. As the kernel $k^l(\cdot,\cdot)$ is assumed to be a continuous function, we have

$$\lim_{n\to\infty} \|\phi^l(\boldsymbol{\xi}_n^{sl}) - \phi^l(\boldsymbol{\xi}_*^{sl})\|^2_{\mathcal{X}^l} = \lim_{n\to\infty} \Big(k^l(\boldsymbol{\xi}_n^{sl}, \boldsymbol{\xi}_n^{sl}) - 2k^l(\boldsymbol{\xi}_n^{sl}, \boldsymbol{\xi}_*^{sl}) + k^l(\boldsymbol{\xi}_*^{sl}, \boldsymbol{\xi}_*^{sl})\Big) = 0$$

where $\|\cdot\|_{\mathcal{X}^l}$ denotes the norm in the RKHS $\mathcal{X}^l$. It thus follows that

$$\lim_{n\to\infty} \phi^l(\boldsymbol{\xi}_n^{sl}) = \phi^l(\boldsymbol{\xi}_*^{sl})$$

and hence $\phi^l : \mathbb{R}^{d_l} \to \mathcal{X}^l$ is a continuous function. ϕ^l is thus measurable with respect to the Borel σ-algebra $\mathcal{B}(\mathcal{X}^l)$ of the RKHS $\mathcal{X}^l$. Since $\boldsymbol{\xi}^{sl}(\cdot) : \mathcal{X}^s \to \mathbb{R}^{d_l}$ is a measurable mapping as well, we conclude that the mapping $f^{sl} = \phi^l(\boldsymbol{\xi}^{sl}(\cdot)) : \mathcal{X}^s \to \mathcal{X}^l$ is measurable with respect to $\mathcal{B}(\mathcal{X}^l)$, for $l = 1, \ldots, L-1$.

We next show that the mappings $f^s \in \mathcal{F}^s$ are measurable. Since the kernel $k^l(\cdot,\cdot)$ is assumed to be continuous, the RKHS $\mathcal{X}^l$ is separable for all l [51]. The separability of the RKHSs ensures that

$$\mathcal{B}(\mathcal{X}) = \bigotimes_{l=1}^{L-1} \mathcal{B}(\mathcal{X}^l)$$

where the right hand side denotes the σ-algebra generated by all finite products of Borel sets in $\mathcal{B}(\mathcal{X}^l)$'s [9]. Hence, denoting the set product of some collection of Borel sets $B^1 \in \mathcal{B}(\mathcal{X}^1), \cdots, B^{L-1} \in \mathcal{B}(\mathcal{X}^{L-1})$ as

$$B^1 \times B^2 \times \cdots \times B^{L-1} = \{(f^1, f^2, \ldots, f^{L-1}) : f^l \in B^l,\ l = 1, \ldots, L-1\},$$

the σ-algebra generated by

$$B = \{B^1 \times \cdots \times B^{L-1} : B^1 \in \mathcal{B}(\mathcal{X}^1), \cdots, B^{L-1} \in \mathcal{B}(\mathcal{X}^{L-1})\}$$

is equal to the Borel σ-algebra $\mathcal{B}(\mathcal{X})$. Then, in order to show that $f^s : \mathcal{X}^s \to \mathcal{X}$ is measurable, it is sufficient to show that the inverse image $(f^s)^{-1}(B)$ of the set B is contained in $\mathcal{B}(\mathcal{X}^s)$. For any element $B^1 \times \cdots \times B^{L-1}$ in B, we have

$$\begin{aligned}(f^s)^{-1}(B^1 \times \cdots \times B^{L-1}) &= \{x^s \in \mathcal{X}^s : f^s(x^s) \in B^1 \times \cdots \times B^{L-1}\} \\ &= \{x^s \in \mathcal{X}^s : f^{s1}(x^s) \in B^1,\ \cdots,\ f^{s(L-1)}(x^s) \in B^{L-1}\} \\ &= \bigcap_{l=1}^{L-1} (f^{sl})^{-1}(B^l).\end{aligned}$$

Since each f^{sl} is measurable, $(f^{sl})^{-1}(B^l) \in \mathcal{B}(\mathcal{X}^s)$. Hence, $(f^s)^{-1}(B^1 \times \cdots \times B^{L-1}) \in \mathcal{B}(\mathcal{X}^s)$ and we conclude that $f^s : \mathcal{X}^s \to \mathcal{X}$ is a measurable mapping.

In order to prove the second part of the lemma, let us fix $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$, and for fixed $\boldsymbol{\xi}$ consider the function $f^{sl}(\cdot)(\boldsymbol{\xi}) : \mathcal{X}^s = \mathbb{R}^{d_0} \to \mathbb{R}$ given by

$$f^{sl}(\cdot)(\boldsymbol{\xi}) = k^l(\boldsymbol{\xi}^{sl}(\cdot), \boldsymbol{\xi}).$$

From the continuity of the kernel k^l and the measurability of the function $\boldsymbol{\xi}^{sl}(\cdot)$, it is easy to conclude that the function $f^{sl}(\cdot)(\boldsymbol{\xi})$ is measurable for any fixed $\boldsymbol{\xi}$. Hence, based on the Borel probability measure μ_s in the source domain, the expectation $E_{x^s}[f^{sl}(x^s)(\boldsymbol{\xi})]$ for fixed $\boldsymbol{\xi}$ is well defined, as well as the function $E_{x^s}[f^{sl}(x^s)] : \mathbb{R}^{d_l} \to \mathbb{R}$ given by

$$E_{x^s}[f^{sl}(x^s)](\boldsymbol{\xi}) E_{x^s}[f^{sl}(x^s)(\boldsymbol{\xi})].$$

Next, we would like to show that $E_{x^s}[f^{sl}(x^s)] \in \mathcal{X}^l$. Consider the linear functional $T_{\mu_s} : \mathcal{X}^l \to \mathbb{R}$ on the RKHS $\mathcal{X}^l$ defined by

$$T_{\mu_s}(\psi) E_{x^s}[\psi(\boldsymbol{\xi}^{sl})]$$

for $\psi \in \mathcal{X}^l$. Following the steps as in the proof of [32, Lemma 3], the linear functional T_{μ_s} is observed to be bounded since

$$\begin{aligned} |T_{\mu_s}(\psi)| &= \left| E_{x^s}[\psi(\boldsymbol{\xi}^{sl})] \right| \leq E_{x^s}\left[|\psi(\boldsymbol{\xi}^{sl})| \right] = E_{x^s}\left[\left| \langle k^l(\boldsymbol{\xi}^{sl}, \cdot), \psi(\cdot) \rangle_{\mathcal{X}^l} \right| \right] \\ &\leq E_{x^s}\left[\|k^l(\boldsymbol{\xi}^{sl}, \cdot)\|_{\mathcal{X}^l} \|\psi\|_{\mathcal{X}^l} \right] = E_{x^s}\left[\sqrt{k^l(\boldsymbol{\xi}^{sl}, \boldsymbol{\xi}^{sl})} \right] \|\psi\|_{\mathcal{X}^l}. \end{aligned}$$

Hence, by the Riesz Representation Theorem [3, Theorem 12.5],[32, Lemma 3], there exists an element $\psi^{sl} \in \mathcal{X}^l$ in the RKHS $\mathcal{X}^l$ (called the mean embedding), such that

$$T_{\mu_s}(\psi) = \langle \psi, \psi^{sl} \rangle_{\mathcal{X}^l}$$

for all $\psi \in \mathcal{X}^l$. In particular, setting $\psi = \phi^l(\boldsymbol{\xi})$ for an arbitrary $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$, we have

$$T_{\mu_s}(\phi^l(\boldsymbol{\xi})) = \langle \phi^l(\boldsymbol{\xi}), \psi^{sl} \rangle_{\mathcal{X}^l} = \psi^{sl}(\boldsymbol{\xi}). \tag{D.1}$$

But it also holds that

$$\begin{aligned} T_{\mu_s}(\phi^l(\boldsymbol{\xi})) &= E_{x^s}[\phi^l(\boldsymbol{\xi})(\boldsymbol{\xi}^{sl})] = E_{x^s}[k^l(\boldsymbol{\xi}, \boldsymbol{\xi}^{sl})] = E_{x^s}[k^l(\boldsymbol{\xi}^{sl}, \boldsymbol{\xi})] \\ &= E_{x^s}[\phi^l(\boldsymbol{\xi}^{sl})(\boldsymbol{\xi})] = E_{x^s}[f^{sl}(x^s)(\boldsymbol{\xi})] = E_{x^s}[f^{sl}(x^s)](\boldsymbol{\xi}). \end{aligned} \tag{D.2}$$

From the equality of the expressions in (D.1) and (D.2), we observe that

$$E_{x^s}[f^{sl}(x^s)] = \psi^{sl} \in \mathcal{X}^l.$$

It then simply follows from the construction of $\mathcal{X}$ that

$$E_{x^s}[f^s(x^s)](E_{x^s}[f^{s1}(x^s)], \ldots, E_{x^s}[f^{s(L-1)}(x^s)])$$

is in the Hilbert space $\mathcal{X}$. ∎

Appendix E. Derivation of Lipschitz constants for common nonlinear activation functions.

Here we derive Lipschitz constants for some widely used nonlinear activation functions. Let $\eta : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ represent an activation function in layer l giving the output $\zeta = \eta(\boldsymbol{\xi})$ for the input $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$.

E.1. ReLU activation.

We begin with the rectified linear unit (ReLU) function $\eta_R : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ given by

$$\boldsymbol{\zeta}(k) = \max\{0, \boldsymbol{\xi}(k)\} \tag{E.1}$$

where $\boldsymbol{\zeta} = \eta_R(\boldsymbol{\xi})$, and the notation $(\cdot)(k)$ denotes the k-th entry of a vector. For two vectors $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \mathbb{R}^{d_l}$, we have

$$\begin{aligned} \|\eta_R(\boldsymbol{\xi}_1) - \eta_R(\boldsymbol{\xi}_2)\|^2 &= \sum_{k=1}^{d_l} (\max\{0, \boldsymbol{\xi}_1(k)\} - \max\{0, \boldsymbol{\xi}_2(k)\})^2 \\ &\leq \sum_{k=1}^{d_l} (\boldsymbol{\xi}_1(k) - \boldsymbol{\xi}_2(k))^2 = \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|^2 \end{aligned} \tag{E.2}$$

where $\max\{\cdot, \cdot\}$ denotes the maximum of two scalar values. We thus get

$$\|\eta_R(\boldsymbol{\xi}_1) - \eta_R(\boldsymbol{\xi}_2)\| \leq \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|$$

which gives the Lipschitz constant of the ReLU function as $L_R = 1$.

E.2. Softplus activation.

Next, we consider the softplus function $\eta_{SP} : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ given by

$$\boldsymbol{\zeta}(k) = \log\left(1 + e^{\boldsymbol{\xi}(k)}\right) \tag{E.3}$$

where $\boldsymbol{\zeta} = \eta_{SP}(\boldsymbol{\xi})$. The derivative of the components of the softplus function can be upper bounded as

$$\left|\frac{d}{dt}\log(1 + e^t)\right| = \left|\frac{e^t}{1 + e^t}\right| < 1 \tag{E.4}$$

for all $t \in \mathbb{R}$. Then for $\boldsymbol{\zeta}_1 = \eta_{SP}(\boldsymbol{\xi}_1)$ and $\boldsymbol{\zeta}_2 = \eta_{SP}(\boldsymbol{\xi}_2)$ with $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \mathbb{R}^{d_l}$, from the mean value theorem we get

$$|\boldsymbol{\zeta}_1(k) - \boldsymbol{\zeta}_2(k)| \leq |\boldsymbol{\xi}_1(k) - \boldsymbol{\xi}_2(k)| \tag{E.5}$$

which implies

$$\|\eta_{SP}(\boldsymbol{\xi}_1) - \eta_{SP}(\boldsymbol{\xi}_2)\| \leq \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|. \tag{E.6}$$

Hence, we obtain the Lipschitz constant of the softplus function as $L_{SP} = 1$.

E.3. Softmax activation. Lastly, we consider the softmax function $\eta_{SM} : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ given by

$$\eta_{SM}(\boldsymbol{\xi}) = [\eta^1_{SM}(\boldsymbol{\xi})\ \eta^2_{SM}(\boldsymbol{\xi})\ \cdots\ \eta^{d_l}_{SM}(\boldsymbol{\xi})]^T$$

where $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$ and each k-th component $\eta^k_{SM}(\boldsymbol{\xi}) : \mathbb{R}^{d_l} \to \mathbb{R}$ of the softmax activation is defined as

$$\eta^k_{SM}(\boldsymbol{\xi}) = \frac{e^{\boldsymbol{\xi}(k)}}{\sum_{n=1}^{d_l} e^{\boldsymbol{\xi}(n)}}. \tag{E.7}$$

Since the functions $\eta^k_{SM}(\boldsymbol{\xi})$ are differentiable for all k, for any two $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \mathbb{R}^{d_l}$, it follows from the multivariable mean value theorem that there exists some $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$ lying in the line segment between $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ such that

$$\eta^k_{SM}(\boldsymbol{\xi}_1) - \eta^k_{SM}(\boldsymbol{\xi}_2) = (\nabla \eta^k_{SM}(\boldsymbol{\xi}))^T(\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2)$$

where $\nabla \eta^k_{SM}(\boldsymbol{\xi}) \in \mathbb{R}^{d_l}$ denotes the gradient of η^k_{SM} at $\boldsymbol{\xi}$. The following inequality is then obtained

$$|\eta^k_{SM}(\boldsymbol{\xi}_1) - \eta^k_{SM}(\boldsymbol{\xi}_2)| \leq \sup_{\boldsymbol{\xi} \in \mathbb{R}^{d_l}} \|\nabla \eta^k_{SM}(\boldsymbol{\xi})\|\, \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|. \tag{E.8}$$

In the sequel, in order to find a Lipschitz constant for the softmax function, we derive a bound on the norm $\|\nabla \eta^k_{SM}(\boldsymbol{\xi})\|$ of its gradient.

For the case $k \neq n$, the derivative of $\eta^k_{SM}(\boldsymbol{\xi})$ with respect to the n-th entry $\boldsymbol{\xi}(n)$ of $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$ is obtained as

$$\frac{\partial \eta^k_{SM}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}(n)} = \frac{\partial}{\partial \boldsymbol{\xi}(n)} \left(\frac{e^{\boldsymbol{\xi}(k)}}{\sum_{r=1}^{d_l} e^{\boldsymbol{\xi}(r)}} \right) = -\frac{e^{\boldsymbol{\xi}(k)} e^{\boldsymbol{\xi}(n)}}{\left(\sum_{r=1}^{d_l} e^{\boldsymbol{\xi}(r)} \right)^2}.$$

Since all $e^{\boldsymbol{\xi}(1)}, \ldots, e^{\boldsymbol{\xi}(d_l)}$ are positive, it is easy to show that $(e^{\boldsymbol{\xi}(1)} + \ldots, + e^{\boldsymbol{\xi}(d_l)})^2 \geq 4 e^{\boldsymbol{\xi}(k)} e^{\boldsymbol{\xi}(n)}$. Using this in the above expression, we get the bound

$$\left| \frac{\partial \eta^k_{SM}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}(n)} \right| \leq \frac{1}{4}. \tag{E.9}$$

Next, for the case $k = n$, we have

$$\frac{\partial \eta^k_{SM}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}(k)} = \frac{\partial}{\partial \boldsymbol{\xi}(k)} \left(\frac{e^{\boldsymbol{\xi}(k)}}{\sum_{r=1}^{d_l} e^{\boldsymbol{\xi}(r)}} \right) = \left(\frac{e^{\boldsymbol{\xi}(k)}}{\sum_{r=1}^{d_l} e^{\boldsymbol{\xi}(r)}} \right) \left(1 - \frac{e^{\boldsymbol{\xi}(k)}}{\sum_{r=1}^{d_l} e^{\boldsymbol{\xi}(r)}} \right).$$

Letting $\alpha = e^{\boldsymbol{\xi}(k)} / \sum_{r=1}^{d_l} e^{\boldsymbol{\xi}(r)}$ in the above expression and observing that the maximum value of the function $\alpha(1-\alpha)$ in the interval $\alpha \in [0, 1]$ is $1/4$, we get

$$\left| \frac{\partial \eta^k_{SM}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}(k)} \right| \leq \frac{1}{4}. \tag{E.10}$$

Combining the results (E.9) and (E.10), the gradient of $\eta_{SM}^k(\boldsymbol{\xi})$ can be bounded as

$$\|\nabla \eta_{SM}^k(\boldsymbol{\xi})\| \leq \frac{\sqrt{d_l}}{4}$$

for any $\boldsymbol{\xi} \in \mathbb{R}^{d_l}$. Using this in (E.8) gives

$$|\eta_{SM}^k(\boldsymbol{\xi}_1) - \eta_{SM}^k(\boldsymbol{\xi}_2)| \leq \frac{\sqrt{d_l}}{4}\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|$$

for any $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \mathbb{R}^{d_l}$, which implies

$$\|\eta_{SM}(\boldsymbol{\xi}_1) - \eta_{SM}(\boldsymbol{\xi}_2)\| \leq \frac{d_l}{4}\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|.$$

Defining

$$d_{\max} = \max_{l=1,\dots,L} d_l$$

we thus get the Lipschitz constant of the softmax function as $L_{SM} = d_{\max}/4$.

Appendix F. Proof of Lemma 3.2.

Proof. We prove the statements only for $\mathcal{F}^s$ and $\mathcal{G}^s$ as the proofs for the target domain are similar. We first show that $\mathcal{F}^s$ is compact with respect to the metric $\mathfrak{d}_{\mathcal{X}}^s$. Let

$$\boldsymbol{\Phi}^s = \{\boldsymbol{\Theta}^s = (\boldsymbol{\Theta}^{s1}, \dots, \boldsymbol{\Theta}^{sL}) : \ |\boldsymbol{\Theta}_{ij}^{sl}| \leq A_\Theta, \forall i, j, l\}$$

denote the parameter space over which the source network parameters are defined. Regarding $\boldsymbol{\Phi}^s$ as the Cartesian product of the corresponding matrix spaces at layers $l = 1, \dots, L$, it follows from the bound $|\boldsymbol{\Theta}_{ij}^{sl}| \leq A_\Theta$ on the network parameters that the finite dimensional set $\boldsymbol{\Phi}^s$ is closed and bounded, hence compact.

We next define a mapping $\mathcal{M}_{\mathcal{F}^s} : \boldsymbol{\Phi}^s \to \mathcal{F}^s$ such that

$$\mathcal{M}_{\mathcal{F}^s}(\boldsymbol{\Theta}^s) = f_{\boldsymbol{\Theta}^s}^s = \left(f_{\boldsymbol{\Theta}^s}^{s1}, \dots, f_{\boldsymbol{\Theta}^s}^{s(L-1)}\right) \tag{F.1}$$

where the notation $f_{\boldsymbol{\Theta}^s}^s(x^s)$ stands for the function $f^s(x^s)$ defined in (3.6) by explicitly referring to its dependence on the network parameters $\boldsymbol{\Theta}^s$. In the following, we show that the mapping $\mathcal{M}_{\mathcal{F}^s}$ is continuous. Let us consider a sequence $\{\boldsymbol{\Theta}_n^s\} \subset \boldsymbol{\Phi}^s$ converging to an element $\boldsymbol{\Theta}_*^s \in \boldsymbol{\Phi}^s$. Since the relation (3.1) between the features of adjacent layers is given by a linear mapping followed by a continuous activation function η^l, the mapping $\boldsymbol{\xi}_{\boldsymbol{\Theta}^s}^{sl}(x^s)$ is a continuous function of $\boldsymbol{\Theta}^s$, i.e.

$$\lim_{n\to\infty} \boldsymbol{\xi}_{\boldsymbol{\Theta}_n^s}^{sl}(x^s) = \boldsymbol{\xi}_{\boldsymbol{\Theta}_*^s}^{sl}(x^s). \tag{F.2}$$

In fact, due to the assumptions on the boundedness (3.2) of the source samples, the boundedness (3.3) of the network parameters, and the Lipschitz continuity (3.12) of the activation

functions η^l, it is easy to show that the convergence in (F.2) is uniform on $\mathcal{X}^s$. Hence, for any given $\epsilon > 0$, one can find some n_0 such that for $n \geq n_0$, we have

$$\|\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_n}(x^s) - \boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_*}(x^s)\| < \epsilon$$

for all $x^s \in \mathcal{X}^s$, for $l = 1, \ldots, L-1$. Then we have

$$\begin{aligned}
&\|f^{sl}_{\boldsymbol{\Theta}^s_n}(x^s) - f^{sl}_{\boldsymbol{\Theta}^s_*}(x^s)\|^2_{\mathcal{X}^l} = \|\phi^l(\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_n}(x^s)) - \phi^l(\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_*}(x^s))\|^2_{\mathcal{X}^l} \\
&= k^l\big(\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_n}(x^s), \boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_n}(x^s)\big) - 2k^l\big(\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_n}(x^s), \boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_*}(x^s)\big) + k^l\big(\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_*}(x^s), \boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_*}(x^s)\big) \\
&\leq 2L_K\|\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_n}(x^s) - \boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s_*}(x^s)\| < 2L_K\epsilon
\end{aligned}$$

for all $x^s \in \mathcal{X}^s$ due to the Lipschitz continuity of the kernels k^l. This gives

$$\|f^s_{\boldsymbol{\Theta}^s_n}(x^s) - f^s_{\boldsymbol{\Theta}^s_*}(x^s)\|^2_{\mathcal{X}} = \sum_{l=1}^{L-1} \|f^{sl}_{\boldsymbol{\Theta}^s_n}(x^s) - f^{sl}_{\boldsymbol{\Theta}^s_*}(x^s)\|^2_{\mathcal{X}^l} < 2(L-1)L_K\epsilon.$$

We have thus obtained

$$\|f^s_{\boldsymbol{\Theta}^s_n}(x^s) - f^s_{\boldsymbol{\Theta}^s_*}(x^s)\|_{\mathcal{X}} < \sqrt{2(L-1)L_K}\sqrt{\epsilon}$$

for all $n \geq n_0$ and for all $x^s \in \mathcal{X}^s$, which shows that $f^s_{\boldsymbol{\Theta}^s_n}(x^s)$ converges to $f^s_{\boldsymbol{\Theta}^s_*}(x^s)$ uniformly on $\mathcal{X}^s$. Then we have

$$\begin{aligned}
\lim_{n\to\infty} \mathfrak{d}^s_{\mathcal{X}}(f^s_{\boldsymbol{\Theta}^s_n}, f^s_{\boldsymbol{\Theta}^s_*}) &= \lim_{n\to\infty} \sup_{x^s\in\mathcal{X}^s} \|f^s_{\boldsymbol{\Theta}^s_n}(x^s) - f^s_{\boldsymbol{\Theta}^s_*}(x^s)\|_{\mathcal{X}} \\
&= \sup_{x^s\in\mathcal{X}^s} \lim_{n\to\infty} \|f^s_{\boldsymbol{\Theta}^s_n}(x^s) - f^s_{\boldsymbol{\Theta}^s_*}(x^s)\|_{\mathcal{X}} = 0
\end{aligned}$$

where the second equality follows from the uniform convergence of $f^s_{\boldsymbol{\Theta}^s_n}(x^s)$. We have thus shown that the mapping $\mathcal{M}_{\mathcal{F}^s} : \boldsymbol{\Phi}^s \to \mathcal{F}^s$ defined in (F.1) is continuous. Since the set $\boldsymbol{\Phi}^s$ is compact, we conclude that the function space $\mathcal{F}^s$ is a compact metric space.

Next, in order to show the compactness of $\mathcal{G}^s$, we proceed in a similar fashion. Let us define a mapping $\mathcal{M}_{\mathcal{G}^s} : \boldsymbol{\Phi}^s \to \mathcal{G}^s$ with $\mathcal{M}_{\mathcal{G}^s}(\boldsymbol{\Theta}^s) = g^s_{\boldsymbol{\Theta}^s}$, where the notation $g^s_{\boldsymbol{\Theta}^s}(x^s) = \boldsymbol{\xi}^{sL}_{\boldsymbol{\Theta}^s}(x^s)$ refers to the network output function defined in (3.7) by clarifying its dependence on the network parameters. Similarly to (F.2), it is easy to observe that $\boldsymbol{\xi}^{sL}_{\boldsymbol{\Theta}^s}(x^s)$ is a continuous function of $\boldsymbol{\Theta}^s$ and for any sequence $\{\boldsymbol{\Theta}^s_n\}$ converging to an element $\boldsymbol{\Theta}^s_* \in \boldsymbol{\Phi}^s$

$$\lim_{n\to\infty} g^s_{\boldsymbol{\Theta}^s_n}(x^s) = \lim_{n\to\infty} \boldsymbol{\xi}^{sL}_{\boldsymbol{\Theta}^s_n}(x^s) = \boldsymbol{\xi}^{sL}_{\boldsymbol{\Theta}^s_*}(x^s) = g^s_{\boldsymbol{\Theta}^s_*}(x^s)$$

uniformly. Hence,

$$\begin{aligned}
\lim_{n\to\infty} \mathfrak{d}^s(g^s_{\boldsymbol{\Theta}^s_n}, g^s_{\boldsymbol{\Theta}^s_*}) &= \lim_{n\to\infty} \sup_{x^s\in\mathcal{X}^s} \|g^s_{\boldsymbol{\Theta}^s_n}(x^s) - g^s_{\boldsymbol{\Theta}^s_*}(x^s)\| \\
&= \sup_{x^s\in\mathcal{X}^s} \lim_{n\to\infty} \|g^s_{\boldsymbol{\Theta}^s_n}(x^s) - g^s_{\boldsymbol{\Theta}^s_*}(x^s)\| = 0.
\end{aligned}$$

Hence, the mapping $\mathcal{M}_{\mathcal{G}^s} : \boldsymbol{\Phi}^s \to \mathcal{G}^s$ is continuous. Then, from the compactness of $\boldsymbol{\Phi}^s$, it follows that the function space $\mathcal{G}^s$ is compact as well. ∎

Appendix G. Proof of Lemma 3.3.

Proof. We obtain the bound only for the source domain, as the derivation for the target domain is identical. Our proof is based on constructing an ϵ-cover for the compact metric space $\mathcal{F}^s$. For two mappings $f_1^s, f_2^s \in \mathcal{F}^s$ defined respectively by the parameter vectors $\mathbf{\Theta}_1^s, \mathbf{\Theta}_2^s$ we have

$$
\begin{aligned}
(\mathfrak{d}_{\mathcal{X}}^s(f_1^s, f_2^s))^2 &= \sup_{x^s \in \mathcal{X}^s} \|f_1^s(x^s) - f_2^s(x^s)\|_{\mathcal{X}}^2 \\
&= \sup_{x^s \in \mathcal{X}^s} \sum_{l=1}^{L-1} \|\phi^l(\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s)) - \phi^l(\boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s))\|_{\mathcal{X}^l}^2 \\
&= \sup_{x^s \in \mathcal{X}^s} \sum_{l=1}^{L-1} k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s)\right) - 2k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s)\right) \\
&\qquad + k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s)\right) \qquad \text{(G.1)} \\
&\leq \sup_{x^s \in \mathcal{X}^s} \sum_{l=1}^{L-1} \left|k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s)\right) - k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s)\right)\right| \\
&\qquad + \left|k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s)\right) - k^l\left(\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s), \boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s)\right)\right| \\
&\leq \sup_{x^s \in \mathcal{X}^s} \sum_{l=1}^{L-1} 2L_K \|\boldsymbol{\xi}_{\mathbf{\Theta}_1^s}^{sl}(x^s) - \boldsymbol{\xi}_{\mathbf{\Theta}_2^s}^{sl}(x^s)\|
\end{aligned}
$$

where the last inequality is due to the Lipschitz continuity of the kernels k^l. We next construct a cover for the set of parameter vectors $\mathbf{\Theta}^s$, which will define a cover for $\mathcal{F}^s$ using the relation in (G.1). From (3.3) the network parameter vectors of layer l are in the compact set

$$
\mathbf{\Theta}^l = \{\mathbf{\Theta}^l = [\mathbf{W}^l \ \mathbf{b}^l] \in \mathbb{R}^{d_l \times (d_{l-1}+1)} : |\mathbf{W}_{ij}^l| \leq A_\Theta, |\mathbf{b}_i^l| \leq A_\Theta, \forall i, j, l\}. \qquad \text{(G.2)}
$$

Then there exists a cover of $\mathbf{\Theta}^l$ consisting of open balls around a set $\mathfrak{G}^l = \{\mathbf{\Theta}_m^l\}_{m=1}^{\kappa^l}$ of regularly sampled grid points, with a distance of δ between adjacent grid centers in each dimension. The maximal overall distance between two adjacent grid centers is then $\delta\sqrt{d_l(d_{l-1}+1)}$. Hence, the distance between any parameter vector $\mathbf{\Theta}^l \in \mathbf{\Theta}^l$ and the nearest grid center $\mathbf{\Theta}_m^l$ is at most

$$
\frac{\delta\sqrt{d_l(d_{l-1}+1)}}{2}
$$

with the number of balls in the cover being

$$
\kappa^l = \left(\frac{2A_\Theta}{\delta} + 1\right)^{d_l(d_{l-1}+1)}.
$$

From the Cartesian product of the grid centers at layers $l = 1, \ldots, L-1$, we then obtain a product grid

$$
\mathfrak{G} = \mathfrak{G}^1 \times \cdots \times \mathfrak{G}^{L-1} = \{\mathbf{\Theta}_k\}_{k=1}^{\kappa^1 \ldots \ \kappa^{L-1}} \qquad \text{(G.3)}
$$

which defines a cover for the overall parameter space

$$\mathbf{\Phi} = \{\mathbf{\Theta} = (\mathbf{\Theta}^1, \ldots, \mathbf{\Theta}^{L-1}) : \ |\mathbf{\Theta}^l_{ij}| \leq A_\Theta, \forall i, j, l\}$$

consisting of

$$\kappa_{\mathfrak{G}} = \prod_{l=1}^{L-1} \kappa^l = \prod_{l=1}^{L-1} \left(\frac{2A_\Theta}{\delta} + 1\right)^{d_l(d_{l-1}+1)}$$

balls. Then for any $f^s \in \mathcal{F}^s$ with parameters $\mathbf{\Theta}^s$, there exists some $f^s_k \in \mathcal{F}^s$ with parameters $\mathbf{\Theta}_k = (\mathbf{\Theta}^1_k, \mathbf{\Theta}^2_k, \ldots, \mathbf{\Theta}^{L-1}_k) \in \mathfrak{G}$ in the product grid such that

$$\|\mathbf{\Theta}^{sl} - \mathbf{\Theta}^l_k\| < \delta\sqrt{d_l(d_{l-1}+1)}. \tag{G.4}$$

For any $x^s \in \mathcal{X}^s$, the distance between the l-th layer features of these parameters can be bounded as

$$\begin{aligned}
\|\boldsymbol{\xi}^{sl}_{\mathbf{\Theta}^s}(x^s) - \boldsymbol{\xi}^l_{\mathbf{\Theta}_k}(x^s)\| &= \left\|\eta^l\left(\mathbf{W}^{sl}\boldsymbol{\xi}^{s(l-1)}_{\mathbf{\Theta}^s}(x^s) + \mathbf{b}^{sl}\right) - \eta^l\left(\mathbf{W}^l_k\,\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) + \mathbf{b}^l_k\right)\right\| \\
&\leq L_\eta\left\|\mathbf{W}^{sl}\boldsymbol{\xi}^{s(l-1)}_{\mathbf{\Theta}^s}(x^s) + \mathbf{b}^{sl} - \mathbf{W}^l_k\,\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) - \mathbf{b}^l_k\right\| \\
&= L_\eta\left\|\mathbf{W}^{sl}\boldsymbol{\xi}^{s(l-1)}_{\mathbf{\Theta}^s}(x^s) - \mathbf{W}^{sl}\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) + \mathbf{W}^{sl}\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) - \mathbf{W}^l_k\,\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) + \mathbf{b}^{sl} - \mathbf{b}^l_k\right\| \\
&\leq L_\eta\|\mathbf{W}^{sl}\|\,\|\boldsymbol{\xi}^{s(l-1)}_{\mathbf{\Theta}^s}(x^s) - \boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s)\| + L_\eta\|\mathbf{W}^{sl} - \mathbf{W}^l_k\|\,\|\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s)\| + L_\eta\|\mathbf{b}^{sl} - \mathbf{b}^l_k\|
\end{aligned} \tag{G.5}$$

where $\mathbf{W}^l_k$, $\mathbf{b}^l_k$, and $\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}$ denote the l-th layer network parameters and features generated by the parameter vector $\mathbf{\Theta}_k$; and $\|\cdot\|$ and $\|\cdot\|_F$ respectively denote the operator norm and the Frobenius norm of a matrix. From (G.2) and (G.4), we have

$$\begin{aligned}
\|\mathbf{W}^{sl}\| \leq \|\mathbf{W}^{sl}\|_F &\leq A_\Theta\sqrt{d_l d_{l-1}} \\
\|\mathbf{W}^{sl} - \mathbf{W}^l_k\| \leq \|\mathbf{W}^{sl} - \mathbf{W}^l_k\|_F &< \delta\sqrt{d_l d_{l-1}} \\
\|\mathbf{b}^{sl} - \mathbf{b}^l_k\| &< \delta\sqrt{d_l}.
\end{aligned}$$

These bounds together with the inequality in (G.5) yield

$$\begin{aligned}
\|\boldsymbol{\xi}^{sl}_{\mathbf{\Theta}^s}(x^s) - \boldsymbol{\xi}^l_{\mathbf{\Theta}_k}(x^s)\| < L_\eta A_\Theta\sqrt{d_l d_{l-1}}\,\|\boldsymbol{\xi}^{s(l-1)}_{\mathbf{\Theta}^s}(x^s) - \boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s)\| \\
+ L_\eta\delta\sqrt{d_l d_{l-1}}\,\|\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s)\| + L_\eta\delta\sqrt{d_l}.
\end{aligned} \tag{G.6}$$

In order to study (G.6), we first obtain an upper bound on the term $\|\boldsymbol{\xi}^l_{\mathbf{\Theta}_k}(x^s)\|$. Notice that for the condition (3.13), we simply have

$$\begin{aligned}
\|\boldsymbol{\xi}^l_{\mathbf{\Theta}_k}(x^s)\| = \|\eta^l\left(\mathbf{W}^l\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) + \mathbf{b}^l\right)\| &= \left(\sum_{i=1}^{d_l}\left(\eta^l_i(\mathbf{W}^l\boldsymbol{\xi}^{l-1}_{\mathbf{\Theta}_k}(x^s) + \mathbf{b}^l)\right)^2\right)^{1/2} \\
&\leq C_\eta\sqrt{d_l}.
\end{aligned} \tag{G.7}$$

Next, for the condition (3.14) we have

$$\begin{aligned}
&\|\boldsymbol{\xi}^0_{\boldsymbol{\Theta}_k}(x^s)\| = \|x^s\| \le A_x \\
&\|\boldsymbol{\xi}^1_{\boldsymbol{\Theta}_k}(x^s)\| = \|\eta^1\left(\mathbf{W}^1\boldsymbol{\xi}^0_{\boldsymbol{\Theta}_k}(x^s)+\mathbf{b}^1\right)\| \le A_\eta\,\|\mathbf{W}^1\boldsymbol{\xi}^0_{\boldsymbol{\Theta}_k}(x^s)+\mathbf{b}^1\| \\
&\qquad\le A_\eta\left(\|\mathbf{W}^1\|\|\boldsymbol{\xi}^0_{\boldsymbol{\Theta}_k}(x^s)\|+\|\mathbf{b}^1\|\right) \le A_\eta A_\Theta\sqrt{d_1d_0}A_x + A_\eta A_\Theta\sqrt{d_1}
\end{aligned}$$

for layers $l=0$ and $l=1$. For $l\ge 2$, one can similarly establish a recursive relation between the parameter vectors of layers l and $l-1$, which yields

$$\begin{aligned}
\|\boldsymbol{\xi}^l_{\boldsymbol{\Theta}_k}(x^s)\| &\le A_\eta\left(\|\mathbf{W}^l\|\|\boldsymbol{\xi}^{l-1}_{\boldsymbol{\Theta}_k}(x^s)\|+\|\mathbf{b}^l\|\right) \\
&\le A_\eta A_\Theta\sqrt{d_ld_{l-1}}\|\boldsymbol{\xi}^{l-1}_{\boldsymbol{\Theta}_k}(x^s)\| + A_\eta A_\Theta\sqrt{d_l} \\
&\le (A_\eta A_\Theta)^l(A_x\sqrt{d_0}+1)\sqrt{d_1}\prod_{k=1}^{l-1}\sqrt{d_{k+1}d_k} \\
&\quad+\sum_{i=2}^{l-1}(A_\eta A_\Theta)^{l+1-i}\sqrt{d_i}\prod_{k=1}^{l-1}\sqrt{d_{k+1}d_k} + A_\eta A_\Theta\sqrt{d_l}.
\end{aligned}$$

Hence, combining this with (G.7), we get

$$\|\boldsymbol{\xi}^l_{\boldsymbol{\Theta}_k}(x^s)\| \le R_l \tag{G.8}$$

for $l=2,\dots,L-1$, where R_l is the constant defined in Lemma 3.3. Using this in (G.6), we obtain

$$\begin{aligned}
\|\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s}(x^s) - \boldsymbol{\xi}^l_{\boldsymbol{\Theta}_k}(x^s)\| &< L_\eta A_\Theta\sqrt{d_ld_{l-1}}\,\|\boldsymbol{\xi}^{s(l-1)}_{\boldsymbol{\Theta}^s}(x^s) - \boldsymbol{\xi}^{l-1}_{\boldsymbol{\Theta}_k}(x^s)\| \\
&\quad + L_\eta\delta\sqrt{d_ld_{l-1}}\,R_{l-1} + L_\eta\delta\sqrt{d_l}.
\end{aligned} \tag{G.9}$$

For layer $l=1$, we have

$$\begin{aligned}
\|\boldsymbol{\xi}^{s1}_{\boldsymbol{\Theta}^s}(x^s) - \boldsymbol{\xi}^1_{\boldsymbol{\Theta}_k}(x^s)\| &< L_\eta A_\Theta\sqrt{d_1d_0}\,\|\boldsymbol{\xi}^{s0}_{\boldsymbol{\Theta}^s}(x^s) - \boldsymbol{\xi}^0_{\boldsymbol{\Theta}_k}(x^s)\| \\
&\quad + L_\eta\delta\sqrt{d_1d_0}\,R_0 + L_\eta\delta\sqrt{d_1} \\
&= L_\eta\delta\sqrt{d_1d_0}\,R_0 + L_\eta\delta\sqrt{d_1}
\end{aligned}$$

since $\boldsymbol{\xi}^{s0}_{\boldsymbol{\Theta}^s}(x^s) = \boldsymbol{\xi}^0_{\boldsymbol{\Theta}_k}(x^s) = x^s$. This relation together with the recursive inequality in (G.9) yields

$$\begin{aligned}
\|\boldsymbol{\xi}^{sl}_{\boldsymbol{\Theta}^s}(x^s) - \boldsymbol{\xi}^l_{\boldsymbol{\Theta}_k}(x^s)\| &< \delta\Bigg((L_\eta R_{l-1}\sqrt{d_ld_{l-1}}+L_\eta\sqrt{d_l}) \\
&\quad+\sum_{i=1}^{l-1}(L_\eta R_{i-1}\sqrt{d_id_{i-1}}+L_\eta\sqrt{d_i})\prod_{k=i+1}^{l}L_\eta A_\Theta\sqrt{d_kd_{k-1}}\Bigg) \\
&= Q_l\delta
\end{aligned} \tag{G.10}$$

for $l = 1, \dots, L-1$. Hence, we have shown that for any $f^s \in \mathcal{F}^s$ with parameters $\mathbf{\Theta}^s$, there exists some $f_k^s \in \mathcal{F}^s$ with parameters $\mathbf{\Theta}_k \in \mathfrak{G}$ in the product grid such that

$$\|\boldsymbol{\xi}^{sl}_{\mathbf{\Theta}^s}(x^s) - \boldsymbol{\xi}^{l}_{\mathbf{\Theta}_k}(x^s)\| < Q_l \delta$$

for any $x^s \in \mathcal{X}^s$. We can now use this in (G.1) to bound the distance $\mathfrak{d}^s_{\mathcal{X}}(f^s, f^s_k)$ as

$$(\mathfrak{d}^s_{\mathcal{X}}(f^s, f^s_k))^2 \leq \sup_{x^s \in \mathcal{X}^s} \sum_{l=1}^{L-1} 2L_K \|\boldsymbol{\xi}^{sl}_{\mathbf{\Theta}^s}(x^s) - \boldsymbol{\xi}^{l}_{\mathbf{\Theta}_k}(x^s)\| < 2L_K \delta \sum_{l=1}^{L-1} Q_l = 2L_K \delta Q. \tag{G.11}$$

Therefore, the set $\{f^s_k\}_{k=1}^{\kappa_{\mathfrak{G}}} \subset \mathcal{F}^s$ provides a cover for $\mathcal{F}^s$ with covering radius $\sqrt{2L_K\delta Q}$. In order to obtain a covering radius of $\epsilon = \sqrt{2L_K\delta Q}$, we set

$$\delta = \frac{\epsilon^2}{2L_K Q}$$

which provides a grid consisting of

$$\prod_{l=1}^{L-1} \kappa^l = \prod_{l=1}^{L-1} \left(\frac{4A_\Theta L_K Q}{\epsilon^2} + 1 \right)^{d_l(d_{l-1}+1)}$$

balls that covers $\mathcal{F}^s$. Hence, we obtain the upper bound

$$\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}^s_{\mathcal{X}}) \leq \prod_{l=1}^{L-1} \left(\frac{4A_\Theta L_K Q}{\epsilon^2} + 1 \right)^{d_l(d_{l-1}+1)}$$

for the covering number stated in the lemma. ∎

Appendix H. Proof of Lemma 3.4.

Proof. We prove the statement of the lemma only for the source function space $\mathcal{H} \circ \mathcal{F}^s$, as the derivations for the target domain are identical. In order to bound the covering number for $\mathcal{H} \circ \mathcal{F}^s$, we proceed as in the proof of Lemma 3.3 and extend the grid construction in (G.3) to include layer L as well. This defines a grid

$$\mathfrak{G}_{\mathcal{H}\circ\mathcal{F}} = \mathfrak{G}^1 \times \cdots \times \mathfrak{G}^L = \{\mathbf{\Theta}_k\}_{k=1}^{\kappa^1 \dots \kappa^L} \tag{H.1}$$

providing a cover for the parameter space

$$\mathbf{\Phi}_{\mathcal{H}\circ\mathcal{F}} = \{\mathbf{\Theta} = (\mathbf{\Theta}^1, \dots, \mathbf{\Theta}^L) : \ |\mathbf{\Theta}^l_{ij}| \leq A_\Theta, \forall i, j, l\}$$

consisting of

$$\prod_{l=1}^{L} \kappa^l = \prod_{l=1}^{L} \left(\frac{2A_\Theta}{\delta} + 1 \right)^{d_l(d_{l-1}+1)} \tag{H.2}$$

balls. Then for any $g^s \in \mathcal{H} \circ \mathcal{F}^s$ with network parameters $\mathbf{\Theta}^s$, there exists some $g_k^s \in \mathcal{H} \circ \mathcal{F}^s$ with network parameters $\mathbf{\Theta}_k = (\mathbf{\Theta}_k^1, \mathbf{\Theta}_k^2, \dots, \mathbf{\Theta}_k^L) \in \mathfrak{G}_{\mathcal{H}\circ\mathcal{F}}$ in the grid such that

$$\|\mathbf{\Theta}^{sl} - \mathbf{\Theta}_k^l\| < \delta\sqrt{d_l(d_{l-1}+1)}$$

for $l = 1, \dots, L$. Proceeding in a similar fashion to the derivations in (G.5) and (G.6), we obtain

$$\begin{aligned}\|\boldsymbol{\xi}_{\mathbf{\Theta}^s}^{sL}(x^s) - \boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L}(x^s)\| &\le L_\eta \|\mathbf{W}^{sL}\| \, \|\boldsymbol{\xi}_{\mathbf{\Theta}^s}^{s(L-1)}(x^s) - \boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L-1}(x^s)\| \\ &\quad + L_\eta \|\mathbf{W}^{sL} - \mathbf{W}_k^L\| \, \|\boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L-1}(x^s)\| + L_\eta \|\mathbf{b}^{sL} - \mathbf{b}_k^L\| \\ &< L_\eta A_\Theta \sqrt{d_L d_{L-1}} \, \|\boldsymbol{\xi}_{\mathbf{\Theta}^s}^{s(L-1)}(x^s) - \boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L-1}(x^s)\| \\ &\quad + L_\eta \delta \sqrt{d_L d_{L-1}} \, \|\boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L-1}(x^s)\| + L_\eta \delta \sqrt{d_L}\end{aligned} \tag{H.3}$$

for any $x^s \in \mathcal{X}^s$. Combining this inequality with the bounds in (G.8) and (G.10) gives

$$\begin{aligned}\|\boldsymbol{\xi}_{\mathbf{\Theta}^s}^{sL}(x^s) - \boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L}(x^s)\| &< L_\eta A_\Theta \sqrt{d_L d_{L-1}} \, Q_{L-1}\delta \\ &\quad + L_\eta \delta \sqrt{d_L d_{L-1}} \, R_{L-1} + L_\eta \delta \sqrt{d_L} \\ &= Q_L \delta.\end{aligned}$$

Recalling the definition of the distance $\mathfrak{d}^s$ in (2.4), we then have

$$\mathfrak{d}^s(g^s, g_k^s) = \sup_{x^s \in \mathcal{X}^s} \|g^s(x^s) - g_k^s(x^s)\| = \sup_{x^s \in \mathcal{X}^s} \|\boldsymbol{\xi}_{\mathbf{\Theta}^s}^{sL}(x^s) - \boldsymbol{\xi}_{\mathbf{\Theta}_k}^{L}(x^s)\| < Q_L \delta.$$

Hence, the grid $\mathfrak{G}_{\mathcal{H}\circ\mathcal{F}}$ in (H.1) provides a cover for $\mathcal{H} \circ \mathcal{F}^s$ with covering radius $Q_L\delta$. For a covering radius of ϵ, we set $\epsilon = Q_L\delta$, which results in a cover with

$$\prod_{l=1}^{L} \left(\frac{2A_\Theta Q_L}{\epsilon} + 1 \right)^{d_l(d_{l-1}+1)} \tag{H.4}$$

balls due to (H.2). We thus get the covering number upper bound

$$\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s) \le \prod_{l=1}^{L} \left(\frac{2A_\Theta Q_L}{\epsilon} + 1 \right)^{d_l(d_{l-1}+1)}$$

stated in the lemma. ∎

Appendix I. Proof of Corollary 3.5.

Proof. In order to analyze the dependence of $\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}_{\mathcal{X}}^s)$ on d and L, we first study how the term R_l in Lemma 3.3 grows with the dimension d and the number of layers L. For condition (3.13), we have

$$R_l = C_\eta \sqrt{d_l} = O(d^{1/2}).$$

For condition (3.14), representing the relevant constant terms as c for simplicity, we have

$$R_l = O\big((cd)^l\big).$$

We next study the term Q_l in (3.15). For condition (3.13), we obtain

$$Q_l = O(c^{l-1}\, d^{l+\frac{1}{2}})$$

which results in

$$Q = O(c^{L-2} d^{L-\frac{1}{2}}). \tag{I.1}$$

Meanwhile, condition (3.14) yields

$$Q_l = O\big((l-1)\, c^{l-1}\, d^l\big)$$

resulting in

$$Q = O\big((L-2)\, c^{L-2}\, d^{L-1}\big). \tag{I.2}$$

For simplicity, we may combine the results in (I.1) and (I.2) through a slightly more pessimistic but brief common upper bound as

$$Q = O\big(L\, c^{L-2} d^L\big)$$

which is valid for both of the conditions in (3.13) and (3.14). Then, from the expressions of the covering numbers $\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}^s_{\mathcal{X}})$ and $\mathcal{N}(\mathcal{F}^t, \epsilon, \mathfrak{d}^t_{\mathcal{X}})$ in Lemma 3.3, we conclude

$$\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}^s_{\mathcal{X}}) = O\left(\left(\frac{cQ}{\epsilon^2}\right)^{d^2 L}\right) = O\left(\left(\frac{L}{\epsilon}\right)^{d^2 L} (cd)^{d^2 L^2}\right)$$

where we have taken the liberty to replace the ϵ^2 term in the denominator with ϵ for simplicity, as they will lead to equivalent bounds. Similarly,

$$\mathcal{N}(\mathcal{F}^t, \epsilon, \mathfrak{d}^t_{\mathcal{X}}) = O\left(\left(\frac{L}{\epsilon}\right)^{d^2 L} (cd)^{d^2 L^2}\right).$$

We next analyze the covering number $\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s)$ for the hypothesis space $\mathcal{H} \circ \mathcal{F}^s$. For condition (3.13), we have

$$Q_L = O(c^{L-1}\, d^{L+\frac{1}{2}})$$

which gives from Lemma 3.4

$$\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s) = O\left(\left(\frac{cQ_L}{\epsilon}\right)^{d^2 L}\right) = O\left(\frac{(cd)^{d^2 L^2}}{\epsilon^{d^2 L}}\right) \tag{I.3}$$

if the $d^2 L/2$ term added to the $d^2 L^2$ term in the exponent is ignored for simplicity. Next, for condition (3.14) we obtain

$$Q_L = O\big((L-1)\, c^{L-1}\, d^L\big)$$

resulting in

$$\mathcal{N}(\mathcal{H}\circ\mathcal{F}^s,\epsilon,\mathfrak{d}^s)=O\left(\left(\frac{cQ_L}{\epsilon}\right)^{d^2L}\right)=O\left(\left(\frac{L}{\epsilon}\right)^{d^2L}(cd)^{d^2L^2}\right). \tag{I.4}$$

Combining the bounds in (I.3) and (I.4), we arrive at the common upper bound

$$\mathcal{N}(\mathcal{H}\circ\mathcal{F}^s,\epsilon,\mathfrak{d}^s)=O\left(\left(\frac{L}{\epsilon}\right)^{d^2L}(cd)^{d^2L^2}\right)$$

which covers both conditions. Identical derivations for the target domain yield

$$\mathcal{N}(\mathcal{H}\circ\mathcal{F}^t,\epsilon,\mathfrak{d}^t)=O\left(\left(\frac{L}{\epsilon}\right)^{d^2L}(cd)^{d^2L^2}\right). \qquad \blacksquare$$

Appendix J. Proof of Theorem 3.6.

Proof. We first notice that, owing to Lemma 3.1, we can analyze MMD-based domain adaptation networks within the setting of Theorem 2.9. The compactness of the function spaces $\mathcal{F}^s$, $\mathcal{F}^t$, $\mathcal{H}\circ\mathcal{F}^s$, and $\mathcal{H}\circ\mathcal{F}^t$ follow from Assumptions 3.1-3.1 due to Lemma 3.2. Assumptions 2.2 and 2.3 are thereby satisfied; hence, the statement of Theorem 2.9 applies to the current setting in consideration.

We recall from Theorem 2.9 that the expected target loss in (3.16) is attained with probability at least

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

Our proof is then based on identifying the rate at which the number of samples should grow with L and d so that each one of the terms subtracted from 1 in the expression (J.1) remains fixed. This will in return guarantee that the generalization gap of $O(\epsilon)$ in (3.16) be attained with high probability.

We begin with the term $\mathcal{N}(\mathcal{F}^s,\frac{\epsilon}{8},\mathfrak{d}^s_{\mathcal{X}})\exp(-a_s(N_s,\epsilon))$. Recalling the definition of $a_s(N_s,\epsilon)$ from Lemma 2.8, we have

$$a_s(N_s,\epsilon)=\theta(N_s\epsilon^2)$$

where we use the notation $\theta(\cdot)$ to refer to asymptotic tight bounds. Combining this with Corollary 3.5, we obtain

$$\begin{aligned}\mathcal{N}(\mathcal{F}^s,\frac{\epsilon}{8},\mathfrak{d}^s_{\mathcal{X}})\exp(-a_s(N_s,\epsilon))&=O\left(\left(\frac{L}{\epsilon}\right)^{d^2L}(cd)^{d^2L^2}\exp(-N_s\epsilon^2)\right)\\&=O\left(\exp\left(d^2L\log\left(\frac{L}{\epsilon}\right)+d^2L^2\log(cd)-N_s\epsilon^2\right)\right).\end{aligned}$$

We conclude that the total number N_s of source samples required to ensure a lower bound on the probability expression (J.1) scales as

$$N_s = O\left(\frac{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right),$$

yielding the sample complexity stated in the theorem. An identical derivation based on bounding the term $\mathcal{N}(\mathcal{F}^t, \frac{\epsilon}{8}, \mathfrak{d}^t_{\mathcal{X}}) \exp(-a_t(N_t, \epsilon))$ shows that N_t has the same sample complexity.

Next, we examine the terms involving the number of labeled samples. Proceeding similarly, we get

$$\begin{aligned}\mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t) e^{-\frac{M_t \epsilon^2}{8\alpha^2 A_\ell^2}} &= O\left(\left(\frac{L\alpha}{\epsilon}\right)^{d^2 L} (cd)^{d^2 L^2} \exp\left(-\frac{M_t \epsilon^2}{\alpha^2}\right)\right) \\ &= O\left(\exp\left(d^2 L \log\left(\frac{L\alpha}{\epsilon}\right) + d^2 L^2 \log(cd) - \frac{M_t \epsilon^2}{\alpha^2}\right)\right).\end{aligned}$$

Recalling that $0 \leq \alpha \leq 1$, we conclude that upper bounding the choice of the weight parameter α by the rate

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

ensures that the probability term $\mathcal{N}(\mathcal{H} \circ \mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t) e^{-\frac{M_t \epsilon^2}{8\alpha^2 A_\ell^2}}$ remain bounded.

Finally, for the number of labeled samples in the source domain, we have

$$\begin{aligned}&\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \frac{\epsilon}{8(1-\alpha) L_\ell}, \mathfrak{d}^s) e^{-\frac{M_s \epsilon^2}{8(1-\alpha)^2 A_\ell^2}} \\ &\quad = O\left(\left(\frac{L(1-\alpha)}{\epsilon}\right)^{d^2 L} (cd)^{d^2 L^2} \exp\left(-\frac{M_s \epsilon^2}{(1-\alpha)^2}\right)\right) \\ &\quad = O\left(\exp\left(d^2 L \log\left(\frac{L(1-\alpha)}{\epsilon}\right) + d^2 L^2 \log(cd) - \frac{M_s \epsilon^2}{(1-\alpha)^2}\right)\right).\end{aligned}$$

Recalling again the bound $0 \leq 1 - \alpha \leq 1$, we observe that the sample complexity

$$M_s = O\left(\frac{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right)$$

ensures a lower bound on the probability expression (J.1), which concludes the proof of the theorem. ∎

Appendix K. Derivation of the bound and the Lipschitz constant for the cross-entropy loss.

We first discuss the magnitude bound A_ℓ for the widely used cross-entropy loss function. Let $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y} \subset \mathbb{R}^m$ be two nonnegative label vectors in the label set $\mathcal{Y} = [0,1] \times \cdots \times [0,1] \subset \mathbb{R}^m$. In its naïve form, the cross-entropy loss between $\mathbf{y}_1$ and $\mathbf{y}_2$ is given by

$$\ell(\mathbf{y}_1, \mathbf{y}_2) = -\sum_{k=1}^{m} \log(\mathbf{y}_1(k))\, \mathbf{y}_2(k) \tag{K.1}$$

where $\mathbf{y}(k)$ denotes the k-th entry of the vector $\mathbf{y}$. While the original form (K.1) of the cross-entropy loss is not bounded, often the following modification is made in order to avoid numerical issues in practical implementations

$$\ell(\mathbf{y}_1, \mathbf{y}_2) = -\sum_{k=1}^{m} \log(\mathbf{y}_1(k) + \delta)\, \mathbf{y}_2(k)$$

where $0 < \delta < 1$ is a positive constant. We then have

$$|\ell(\mathbf{y}_1, \mathbf{y}_2)| \leq \sum_{k=1}^{m} |-\log(\mathbf{y}_1(k) + \delta)\mathbf{y}_2(k)| \leq m \max\{|\log(\delta)|,\ \log(1+\delta)\}.$$

Assuming that δ is very small, we get the following bound on the loss magnitude

$$|\ell(\mathbf{y}_1, \mathbf{y}_2)| \leq A_\ell m\, |\log(\delta)|.$$

We next derive the Lipschitz constant L_ℓ of the cross-entropy loss function. For any $\mathbf{y}, \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}$ we have

$$\begin{aligned} |\ell(\mathbf{y}_1, \mathbf{y}) - \ell(\mathbf{y}_2, \mathbf{y})| &= \left| -\sum_{k=1}^{m} \log(\mathbf{y}_1(k) + \delta)\mathbf{y}(k) + \sum_{k=1}^{m} \log(\mathbf{y}_2(k) + \delta)\mathbf{y}(k) \right| \\ &\leq \sum_{k=1}^{m} |\log(\mathbf{y}_2(k) + \delta) - \log(\mathbf{y}_1(k) + \delta)|\,. \end{aligned} \tag{K.2}$$

For any $t \geq \delta$, we have

$$\left| \frac{d}{dt} \log(t) \right| = \left| \frac{1}{t} \right| \leq \frac{1}{\delta}$$

which gives

$$\left| \frac{\log(\mathbf{y}_2(k) + \delta) - \log(\mathbf{y}_1(k) + \delta)}{\mathbf{y}_2(k) - \mathbf{y}_1(k)} \right| \leq \frac{1}{\delta}$$

due to the mean value theorem. Using this in (K.2), we get

$$|\ell(\mathbf{y}_1, \mathbf{y}) - \ell(\mathbf{y}_2, \mathbf{y})| \leq \sum_{k=1}^{m} \delta^{-1} |\mathbf{y}_2(k) - \mathbf{y}_1(k)| \leq \delta^{-1} \sqrt{m}\, \|\mathbf{y}_2 - \mathbf{y}_1\|$$

which shows that the cross-entropy loss is Lipschitz continuous with respect to the first argument with constant

$$L_\ell \delta^{-1} \sqrt{m}.$$

Appendix L. Proof of Lemma 3.7.

Proof. Due to the assumption of compactness of the function classes $\mathcal{V}^s$ and $\mathcal{V}^t$, there exists an ϵ-cover of each function space. Let us denote the cover numbers of $\mathcal{V}^s$ and $\mathcal{V}^t$ as

$$\kappa^s = \mathcal{N}(\mathcal{V}^s, \epsilon, \mathfrak{d}^s_{\mathcal{V}}), \qquad \kappa^t = \mathcal{N}(\mathcal{V}^t, \epsilon, \mathfrak{d}^t_{\mathcal{V}})$$

respectively, and the corresponding sets of ball centers as $\{v^s_k\}_{k=1}^{\kappa^s}$ and $\{v^t_l\}_{l=1}^{\kappa^t}$. Then, for any $v^s \in \mathcal{V}^s$ and any $v^t \in \mathcal{V}^t$ there exist some $v^s_k \in \mathcal{V}^s$ and $v^t_l \in \mathcal{V}^t$ such that

$$\begin{aligned} \mathfrak{d}^s_{\mathcal{V}}(v^s, v^s_k) &= \sup_{x^s \in \mathcal{X}^s} |v^s(x^s) - v^s_k(x^s)| < \epsilon \\ \mathfrak{d}^t_{\mathcal{V}}(v^t, v^t_l) &= \sup_{x^t \in \mathcal{X}^t} |v^t(x^t) - v^t_l(x^t)| < \epsilon. \end{aligned} \tag{L.1}$$

Let us denote

$$\begin{aligned} & D(v^s_k, v^t_l) \left| E[v^s_k(x^s)] - E[v^t_l(x^t)] \right| \\ & \hat{D}(v^s_k, v^t_l) \left| \frac{1}{N_s} \sum_{i=1}^{N_s} v^s_k(x^s_i) - \frac{1}{N_t} \sum_{j=1}^{N_t} v^t_l(x^t_j) \right|. \end{aligned}$$

Take any $f^s \in \mathcal{F}^s$, $f^t \in \mathcal{F}^t$ and $\Delta \in \mathcal{D}$. We have

$$\begin{aligned} & |D_\Delta(f^s, f^t) - \hat{D}_\Delta(f^s, f^t)| \\ & = |D_\Delta(f^s, f^t) - D(v^s_k, v^t_l) + D(v^s_k, v^t_l) - \hat{D}(v^s_k, v^t_l) + \hat{D}(v^s_k, v^t_l) - \hat{D}_\Delta(f^s, f^t)| \\ & \leq |D_\Delta(f^s, f^t) - D(v^s_k, v^t_l)| + |D(v^s_k, v^t_l) - \hat{D}(v^s_k, v^t_l)| + |\hat{D}(v^s_k, v^t_l) - \hat{D}_\Delta(f^s, f^t)|. \end{aligned} \tag{L.2}$$

We proceed by bounding each one of the three terms at the right hand side of the inequality in (L.2). The first term can be upper bounded as

$$\begin{aligned} |D_\Delta(f^s, f^t) - D(v^s_k, v^t_l)| &= \left| |E[v^s(x^s)] - E[v^t(x^t)]| - |E[v^s_k(x^s)] - E[v^t_l(x^t)]| \right| \\ &\leq \left| E[v^s(x^s)] - E[v^t(x^t)] - E[v^s_k(x^s)] + E[v^t_l(x^t)] \right| \\ &\leq |E[v^s(x^s)] - E[v^s_k(x^s)]| + |E[v^t(x^t)] - E[v^t_l(x^t)]| < 2\epsilon \end{aligned} \tag{L.3}$$

where the last inequality follows from (L.1). For the third term in (L.2), one can similarly show that

$$|\hat{D}(v^s_k, v^t_l) - \hat{D}_\Delta(f^s, f^t)| < 2\epsilon. \tag{L.4}$$

We lastly study the second term in (L.2). We have

$$\begin{aligned} & |D(v^s_k, v^t_l) - \hat{D}(v^s_k, v^t_l)| \\ & = \left| \left| E[v^s_k(x^s)] - E[v^t_l(x^t)] \right| - \left| \frac{1}{N_s} \sum_{i=1}^{N_s} v^s_k(x^s_i) - \frac{1}{N_t} \sum_{j=1}^{N_t} v^t_l(x^t_j) \right| \right| \\ & \leq \left| E[v^s_k(x^s)] - E[v^t_l(x^t)] - \frac{1}{N_s} \sum_{i=1}^{N_s} v^s_k(x^s_i) + \frac{1}{N_t} \sum_{j=1}^{N_t} v^t_l(x^t_j) \right| \\ & \leq \left| \frac{1}{N_s} \sum_{i=1}^{N_s} v^s_k(x^s_i) - E[v^s_k(x^s)] \right| + \left| \frac{1}{N_t} \sum_{j=1}^{N_t} v^t_l(x^t_j) - E[v^t_l(x^t)] \right|. \end{aligned} \tag{L.5}$$

As the domain discriminator is bounded due to Assumption 3.2, from Hoeffding's inequality we have

$$P\left(\left|\frac{1}{N_s}\sum_{i=1}^{N_s} v_k^s(x_i^s) - E[v_k^s(x^s)]\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{N_s\epsilon^2}{2C_{\mathcal{D}}^2}\right)$$

for a fixed $v_k^s \in \mathcal{V}^s$, and a similar inequality can be obtained for a fixed $v_l^t \in \mathcal{V}^t$. Applying the union bound over all ball centers $\{v_k^s\}_{k=1}^{\kappa^s}$ and $\{v_l^t\}_{l=1}^{\kappa^t}$, we get that with probability at least

$$1 - 2\kappa^s \exp\left(-\frac{N_s\epsilon^2}{2C_{\mathcal{D}}^2}\right) - 2\kappa^t \exp\left(-\frac{N_t\epsilon^2}{2C_{\mathcal{D}}^2}\right)$$

we have

$$\left|\frac{1}{N_s}\sum_{i=1}^{N_s} v_k^s(x_i^s) - E[v_k^s(x^s)]\right| < \epsilon \quad \text{and} \quad \left|\frac{1}{N_t}\sum_{j=1}^{N_t} v_l^t(x_j^t) - E[v_l^t(x^t)]\right| < \epsilon$$

for all ball centers, which implies from (L.5)

$$|D(v_k^s, v_l^t) - \hat{D}(v_k^s, v_l^t)| < 2\epsilon.$$

Combining this result with the bounds in (L.2)-(L.4), we get

$$\begin{aligned} &P\left(\sup_{f^s\in\mathcal{F}^s, f^t\in\mathcal{F}^t, \Delta\in\mathcal{D}} |D_\Delta(f^s, f^t) - \hat{D}_\Delta(f^s, f^t)| \leq 6\epsilon\right) \\ &\geq 1 - 2\kappa^s \exp\left(-\frac{N_s\epsilon^2}{2C_{\mathcal{D}}^2}\right) - 2\kappa^t \exp\left(-\frac{N_t\epsilon^2}{2C_{\mathcal{D}}^2}\right). \end{aligned}$$

Replacing ϵ with $\epsilon/6$, we get the statement of the lemma. ∎

Appendix M. Proof of Theorem 3.8.

Proof. We begin by bounding the expected target loss as

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

using Assumption 3.2. It follows that

$$\begin{aligned} \mathcal{L}^t(f^t, h) &= \alpha\mathcal{L}^t(f^t, h) + (1-\alpha)\mathcal{L}^t(f^t, h) \\ &\leq \alpha\mathcal{L}^t(f^t, h) + (1-\alpha)\left(\mathcal{L}^s(f^s, h) + R_A\, D_\Delta(f^s, f^t)\right) \\ &= \mathcal{L}_\alpha(f^s, f^t, h) + (1-\alpha)R_A\, D_\Delta(f^s, f^t). \end{aligned} \tag{M.1}$$

We next aim to upper bound the expected loss $\mathcal{L}_\alpha(f^s, f^t, h)$ and the expected distribution distance $D_\Delta(f^s, f^t)$ in terms of their empirical counterparts. It follows from Assumptions 3.1 and 3.2 that the source hypothesis space $\mathcal{G}^s = \mathcal{H}\circ\mathcal{F}^s$, the target hypothesis space $\mathcal{G}^t = \mathcal{H}\circ\mathcal{F}^t$, the source domain discriminator space $\mathcal{V}^s = \mathcal{D}\circ\mathcal{F}^s$ and the target domain discriminator space

$\mathcal{V}^t = \mathcal{D} \circ \mathcal{F}^t$ are compact with respect to the metrics $\mathfrak{d}^s, \mathfrak{d}^t, \mathfrak{d}^s_{\mathcal{V}}, \mathfrak{d}^t_{\mathcal{V}}$ respectively, which can be shown by following similar steps as in the proof of Lemma 3.2 in Appendix F.

Due to the compactness of $\mathcal{G}^s, \mathcal{G}^t$ and the assumptions on the classification loss function ℓ, we have

$$\begin{aligned} &P\left(\sup_{f^s\in\mathcal{F}^s, f^t\in\mathcal{F}^t, h\in\mathcal{H}} |\mathcal{L}_\alpha(f^s,f^t,h) - \hat{\mathcal{L}}_\alpha(f^s,f^t,h)| \le \epsilon\right) \\ &\ge 1 - 2\mathcal{N}(\mathcal{H}\circ\mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t) e^{-\frac{M_t\epsilon^2}{8\alpha^2 A_\ell^2}} - 2\mathcal{N}(\mathcal{H}\circ\mathcal{F}^s, \frac{\epsilon}{8(1-\alpha)L_\ell}, \mathfrak{d}^s) e^{-\frac{M_s\epsilon^2}{8(1-\alpha)^2 A_\ell^2}} \end{aligned} \tag{M.2}$$

from Lemma 2.3. Similarly, the compactness of $\mathcal{V}^s, \mathcal{V}^t$ together with Assumption 3.2 implies that

$$\begin{aligned} &P\left(\sup_{f^s\in\mathcal{F}^s, f^t\in\mathcal{F}^t, \Delta\in\mathcal{D}} |D_\Delta(f^s,f^t) - \hat{D}_\Delta(f^s,f^t)| \le \epsilon\right) \\ &\ge 1 - 2\mathcal{N}(\mathcal{V}^s, \frac{\epsilon}{6}, \mathfrak{d}^s_{\mathcal{V}}) \exp\left(-\frac{N_s\epsilon^2}{72C_{\mathcal{D}}^2}\right) - 2\mathcal{N}(\mathcal{V}^t, \frac{\epsilon}{6}, \mathfrak{d}^t_{\mathcal{V}}) \exp\left(-\frac{N_t\epsilon^2}{72C_{\mathcal{D}}^2}\right) \end{aligned} \tag{M.3}$$

due to Lemma 3.7.

Combining the results in (M.1), (M.2), and (M.3), we get that with probability at least

$$\begin{aligned} &1 - 2\mathcal{N}(\mathcal{H}\circ\mathcal{F}^s, \frac{\epsilon}{8(1-\alpha)L_\ell}, \mathfrak{d}^s) e^{-\frac{M_s\epsilon^2}{8(1-\alpha)^2 A_\ell^2}} - 2\mathcal{N}(\mathcal{H}\circ\mathcal{F}^t, \frac{\epsilon}{8\alpha L_\ell}, \mathfrak{d}^t) e^{-\frac{M_t\epsilon^2}{8\alpha^2 A_\ell^2}} \\ &- 2\mathcal{N}(\mathcal{V}^s, \frac{\epsilon}{6}, \mathfrak{d}^s_{\mathcal{V}}) \exp\left(-\frac{N_s\epsilon^2}{72C_{\mathcal{D}}^2}\right) - 2\mathcal{N}(\mathcal{V}^t, \frac{\epsilon}{6}, \mathfrak{d}^t_{\mathcal{V}}) \exp\left(-\frac{N_t\epsilon^2}{72C_{\mathcal{D}}^2}\right) \end{aligned} \tag{M.4}$$

the expected target loss is bounded as

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

In the sequel, we examine each one of the terms in the probability expression in (M.4). As for the covering numbers of $\mathcal{H}\circ\mathcal{F}^s$ and $\mathcal{H}\circ\mathcal{F}^t$, Assumptions 3.1, 3.1, and 3.2 ensure that the result in Lemma 3.1 applies to this setting as well, which implies that the rate of growth of $\mathcal{N}(\mathcal{H}\circ\mathcal{F}^s, \epsilon, \mathfrak{d}^s)$ and $\mathcal{N}(\mathcal{H}\circ\mathcal{F}^t, \epsilon, \mathfrak{d}^t)$ with L and d is upper bounded by

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

due to Corollary 3.5. Then, following the very same steps as in the proof of Theorem 3.6, we get that upper bounding the weight parameter α by

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

together with scaling M_s at rate

$$M_s = O\left(\frac{d^2 L \, \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right)$$

ensures an upper bound on the terms

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

and

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

in the probability expression in (M.4).

Then, in order to analyze the covering numbers of $\mathcal{V}^s$ and $\mathcal{V}^t$, we proceed with the following reasoning: Noting the paralel between the structures of the domain discriminator and the feature extractor network parameters considered in Assumptions 3.2, 3.1 and 3.2, we observe that the function space $\mathcal{V}^s = \mathcal{D} \circ \mathcal{F}^s$ has an identical construction to the function space $\mathcal{G}^s = \mathcal{H} \circ \mathcal{F}^s$, if the metric

$$\mathfrak{d}^s(g_1^s, g_2^s) = \sup_{x^s \in \mathcal{X}^s} \|g_1^s(x^s) - g_2^s(x^s)\|$$

based on the Euclidean distance in $\mathbb{R}^m$ is replaced by its counterpart

$$\mathfrak{d}_{\mathcal{V}}^s(v_1^s, v_2^s) = \sup_{x^s \in \mathcal{X}^s} |v_1^s(x^s) - v_2^s(x^s)|$$

which uses the Euclidean distance in $\mathbb{R}$ instead. Hence, the latter is a special case of the former that can be obtained by setting $m = 1$. Consequently, the analysis of the covering number $\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s)$ in Corollary 3.5 immediately applies to $\mathcal{N}(\mathcal{D} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}_{\mathcal{V}}^s)$ as well, only by replacing the number of layers L with the total number of layers $L + K - 1$ in the cascade network formed by the combination of the feature extractor and the domain discriminator networks. We thus get

$$\mathcal{N}(\mathcal{V}^s, \epsilon, \mathfrak{d}_{\mathcal{V}}^s) = O\left(\left(\frac{L+K}{\epsilon}\right)^{d^2(L+K)} (cd)^{d^2(L+K)^2}\right)$$

which yields

$$\begin{aligned} &\mathcal{N}(\mathcal{V}^s, \frac{\epsilon}{6}, \mathfrak{d}_{\mathcal{V}}^s) \exp\left(-\frac{N_s \epsilon^2}{72 C_{\mathcal{D}}^2}\right) \\ &= O\left(\left(\frac{L+K}{\epsilon}\right)^{d^2(L+K)} (cd)^{d^2(L+K)^2} \exp\left(-\frac{N_s \epsilon^2}{72 C_{\mathcal{D}}^2}\right)\right) \\ &= O\left(\exp\left(d^2(L+K)\log\left(\frac{L+K}{\epsilon}\right) + d^2(L+K)^2 \log(cd) - \frac{N_s \epsilon^2}{72 C_{\mathcal{D}}^2}\right)\right). \end{aligned} \tag{M.5}$$

We thus conclude that the sample complexity

$$N_s = O\left(\frac{d^2(L+K)\log\left(\frac{L+K}{\epsilon}\right) + d^2(L+K)^2\log(d)}{\epsilon^2}\right)$$

ensures an upper bound on the term (M.5). The same arguments also hold for the target domain, resulting in the sample complexity

$$N_t = O\left(\frac{d^2(L+K)\log\left(\frac{L+K}{\epsilon}\right) + d^2(L+K)^2\log(d)}{\epsilon^2}\right)$$

for the number of target samples, which concludes the proof of the theorem. ∎

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] G. Bachman and L. Narici, *Functional Analysis*, Academic Press, New York and London, 1966.

[4] 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.

[5] P. L. Bartlett, D. J. Foster, and M. Telgarsky, *Spectrally-normalized margin bounds for neural networks*, in Advances in Neural Information Processing Systems 30, 2017, pp. 6240–6249.

[6] P. L. Bartlett, A. Montanari, and A. Rakhlin, *Deep learning: a statistical viewpoint*, Acta Numerica, 30 (2021), pp. 87–201.

[7] 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.

[8] 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.

[9] V. I. Bogachev, *Measure Theory*, Springer, Berlin Heidelberg, 2007.

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

[11] C. Cai, *Deep adaptation networks (DAN) in PyTorch*, 2020. [Online]. Available: https://github.com/CuthbertCai/pytorch_DAN. Accessed: 2024-11-13.

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

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

[14] 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.

[15] 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.

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

[17] 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.

[18] Y. Deng et al., *On the hardness of robustness transfer: A perspective from Rademacher complexity over symmetric difference hypothesis space*, arXiv preprint: http://arxiv.org/abs/2302.12351, (2023).

[19] 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.
[20] 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.
[21] N. Dunford and J. Schwartz, *Linear Operators, Part 1: General Theory*, Wiley Classics Library, Interscience Publishers Inc., New York, 1988.
[22] M. El Hamri, Y. Bennani, and I. Falih, *Theoretical guarantees for domain adaptation with hierarchical optimal transport*, Mach. Learn., 114 (2025), p. 119.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] Y. Ganin et al., *Domain-adversarial training of neural networks*, J. Mach. Learn. Res., 17 (2016), pp. 59:1–59:35.
[28] 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.
[29] M. Ghifary et al., *Deep reconstruction-classification networks for unsupervised domain adaptation*, in European Conf. Comp. Vision,, vol. 9908, 2016, pp. 597–613.
[30] GitHub repository, *Dann_py3*, 2023. [Online]. Available: https://github.com/fungtion/DANN_py3.git.
[31] N. Golowich, A. Rakhlin, and O. Shamir, *Size-independent sample complexity of neural networks*, in Conference On Learning Theory, vol. 75, 2018, pp. 297–299.
[32] 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.
[33] N. Harvey, C. Liaw, and A. Mehrabian, *Nearly-tight VC-dimension bounds for piecewise linear neural networks*, in Proc. Conf. Learning Theory, vol. 65, 2017, pp. 1064–1068.
[34] 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.
[35] 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).
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] 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.
[41] Y. Mansour, M. Mohri, and A. Rostamizadeh, *Domain adaptation: Learning bounds and algorithms*, in The 22nd Conference on Learning Theory, 2009.
[42] Massachusetts Institute of Technology, *MIT-CBCL face recognition database.* Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html.
[43] 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.
[44] M. Mohri and A. M. Medina, *New analysis and algorithm for learning with drifting distributions*, in Int. Conf. Algorithmic Learning Theory, vol. 7568, 2012, pp. 124–138.
[45] B. Neyshabur, S. Bhojanapalli, and N. Srebro, *A PAC-bayesian approach to spectrally-normalized*

margin bounds for neural networks, in Int. Conf. Learning Representations, 2018.
[46] 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.
[47] 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.
[48] 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).
[49] A. Sicilia, K. Atwell, M. Alikhani, and S. J. Hwang, *PAC-Bayesian domain adaptation bounds for multiclass learners*, in Proc. Conf. Uncertainty in Artificial Intelligence, vol. 180, 2022, pp. 1824–1834.
[50] P. Singhal, R. Walambe, S. Ramanna, and K. Kotecha, *Domain adaptation: Challenges, methods, datasets, and applications*, IEEE Access, 11 (2023), pp. 6973–7020.
[51] M. Subedi and J. Cortez, *Reproducing Kernel Hilbert Spaces - Part III*. https://www.math.uh.edu/~dlabate/LectureNote_06.pdf. Accessed: 2022-03-22.
[52] B. Sun and K. Saenko, *Deep CORAL: correlation alignment for deep domain adaptation*, in European Conf. Comp. Vision, vol. 9915, 2016, pp. 443–450.
[53] 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.
[54] R. Tachet des Combes et al., *Domain adaptation with conditional distribution matching and generalized label shift*, in Neural Inf. Proc. Systems, 2020.
[55] H. Tang and K. Jia, *Discriminative adversarial domain adaptation*, in AAAI Conference on Artificial Intelligence, 2020, pp. 5940–5947.
[56] N. Tripuraneni, M. I. Jordan, and C. Jin, *On the theory of transfer learning: The importance of task diversity*, in Advances in Neural Information Processing Systems, 2020.
[57] 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.
[58] 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.
[59] 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).
[60] 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.
[61] E. Vural, *Generalization bounds for domain adaptation via domain transformations*, in IEEE Int. Workshop Machine Learning for Signal Processing, 2018, pp. 1–6.
[62] M. Wang and W. Deng, *Deep visual domain adaptation: A survey*, Neurocomputing, 312 (2018), pp. 135–153.
[63] X. Wang and J. Schneider, *Generalization bounds for transfer learning under model shift*, in Proc. Conf. Uncertainty in Artificial Intelligence, 2015, pp. 922–931.
[64] Z. Wang and Y. Mao, *Information-theoretic analysis of unsupervised domain adaptation*, in Int. Conf. Learning Representations, 2023.
[65] Z. Wang and Y. Mao, *On f-divergence principled domain adaptation: An improved framework*, in Advances in Neural Information Processing Systems, 2024.
[66] B. Wang et al., *Gap minimization for knowledge sharing and transfer*, J. Mach. Learn. Res., 24 (2023), pp. 33:1–33:57.
[67] 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.
[68] 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.
[69] X. Wu, J. H. Manton, U. Aickelin, and J. Zhu, *On the generalization for transfer learning: An information-theoretic analysis*, IEEE Trans. Inf. Theory, 70 (2024), pp. 7089–7124.
[70] Z. Xia et al., *Meta domain adaptation approach for multi-domain ranking*, IEEE Access, 13 (2025), pp. 92921–92931.
[71] 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.

[72] 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.
[73] V. V. Yurinskii, *Exponential inequalities for sums of random vectors*, Journal of Multivariate Analysis, 6 (1976), pp. 473–499.
[74] W. Zellinger, B. A. Moser, and S. Saminger-Platz, *On generalization in moment-based domain adaptation*, Ann. Math. Artif. Intell., 89 (2021), pp. 333–369.
[75] 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.
[76] 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.
[77] 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.
[78] M. H. Zonoozi and V. Seydi, *A survey on adversarial domain adaptation*, Neural Process. Lett., 55 (2023), pp. 2429–2469.
[79] 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.