

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

3    Elif Vural<sup>†</sup> and Hüseyin Karaca<sup>‡</sup>

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

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

25    **MSC codes.** 68Q32, 68T05, 68T07

26    **1. Introduction.** Domain adaptation is a subfield of machine learning that aims to im-  
27   prove model performance in a target domain by leveraging the greater availability of labeled  
28   samples in a source domain. The main challenge in domain adaptation is to address the  
29   discrepancy between the source and target distributions, which can take various forms such  
30   as covariate shift [27], label shift [2, 40], as well as more challenging heterogeneous settings  
31   with source and target samples originating from different data spaces [37]. Early work in  
32   domain adaptation explored instance reweighting methods for covariate shift [25, 39], fea-  
33   ture augmentation approaches [11, 12, 14], and techniques for learning feature projections  
34   or transformations [3, 35, 54]. More recently, in line with broader advances in data science,  
35   domain adaptation research over the last decade has largely shifted towards deep learning-

---

\*Submitted to the editors February 16, 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.

<sup>†</sup>Department of Electrical and Electronics Engineering, METU, Ankara, Türkiye ([velif@metu.edu.tr](mailto:velif@metu.edu.tr), <http://blog.metu.edu.tr/velif/>).

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

36 based techniques [37, 47]. Metrics such as maximum mean discrepancy (MMD) [29, 44, 22]  
37 lead to efficient solutions for aligning source and target domains across various applications  
38 [55, 50, 52, 53]. Adversarial architectures [21, 43, 41, 58] and reconstruction-based approaches  
39 using encoder-decoder structures [23, 6, 59] are also commonly employed.

40 Despite the variety of models and the diversity of solutions, the basic paradigm in do-  
41 main adaptation - whether using shallow methods or neural networks- often boils down to  
42 first aligning the source and target domains by mapping them to a common space through  
43 feature transformations, followed by learning a hypothesis function, typically a classifier, in  
44 that shared domain. The alignment of the source and target distributions is achieved by  
45 minimizing a suitably defined *distribution distance* (also referred to as *domain discrepancy* or  
46 *distribution divergence*), with common choices including MMD [29], covariance-based metrics  
47 [38], and the Wasserstein distance [7, 9, 16]. Although domain adaptation algorithms have  
48 been successfully applied across a wide range of fields including computer vision, time-series  
49 analysis, and natural language processing [37, 58], surprisingly, the literature still lacks a thor-  
50ough theoretical characterization of their performance. In particular, there is a notable gap  
51 in understanding the behavior of *domain alignment algorithms*, which we define as methods  
52 that explicitly map source and target domains to a common representation through feature  
53 transformations. In this paper, we focus on this important class of algorithms, and aim to  
54 provide a rigorous theoretical analysis of their performance.

55 Most existing theoretical analyses focus on understanding how the discrepancy between  
56 source and target domains affects the target-domain performance of classifiers trained to  
57 perform well on the source domain [36, 5, 31, 56, 13, 49]. While these studies provide useful  
58 insight into how models trained with abundant source labels generalize to a target domain  
59 with limited or no labeled data, they inherently assume that source and target data reside in  
60 the same space. Consequently, their results do not straightforwardly extend to the prevalent  
61 framework where source and target domains are aligned through feature transformations or  
62 mappings -whether shallow or deep- prior to classification. Only a few studies have investigated  
63 the performance of domain alignment algorithms [57, 17, 48]; however, these works rather focus  
64 on specific transformation types, such as linear mappings [57] or location and scale changes  
65 [48]. Some literature has investigated the performance and sample complexity of transfer  
66 learning via deep learning approaches [19, 33, 26]. However, domain adaptation and transfer  
67 learning remain distinct problems: transfer learning deals with differing source and target  
68 tasks, unlike domain adaptation. Notably, the characterization of the sample complexity  
69 of domain-adaptive neural networks remains an important yet largely unexplored subject in  
70 current learning theory. It is well established that the amount of data required to successfully  
71 train a neural network increases with the size of the network to prevent overfitting, and many  
72 studies have addressed this issue in classical single-domain settings [1, 34, 51, 45, 10]. To the  
73 best of our knowledge, however, the scaling of labeled and unlabeled source and target sample  
74 requirements with respect to the width and depth of domain-adaptive networks has not been  
75 extensively studied yet.

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

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

82 • 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  
83 domain. The learning objective minimizes a loss function composed of a weighted (convex)  
84 combination of the source and target classification losses, along with a distribution distance  
85 term that measures the discrepancy between the aligned domains. At this stage, our analysis  
86 remains general and does not assume any specific structure for the learning algorithm. In  
87 [Section 2.2 \(Theorem 2.4\)](#), we present a probabilistic bound on the expected target loss in  
88 terms of the empirical weighted loss and the expected distribution discrepancy.

89 • In [Section 2.3](#) we develop these results for the setting where the distribution distance is  
90 selected as the popular maximum mean discrepancy (MMD) metric. In [Theorem 2.7](#), we show  
91 that the expected target loss can be effectively bounded in terms of the empirical classification  
92 and distribution losses alone. This bound holds provided that the number of labeled source  
93 samples  $M_s$  scales logarithmically with the covering number of the composite hypothesis  
94 class  $\mathcal{H} \circ \mathcal{F}^s$ , while the total number of source and target samples,  $N_s$  and  $N_t$ , must scale  
95 logarithmically with the covering numbers of the feature transformation classes  $\mathcal{F}^s$  and  $\mathcal{F}^t$ .

96 • In [Sections 3.1](#) and [3.2](#) we extend our analysis to domain-adaptive deep learning algo-  
97 rithms and, in particular, investigate their sample complexity. We consider two pioneering  
98 approaches that have inspired a large body of follow-up work: MMD-based domain adaptation  
99 networks [29, 44, 22] and adversarial domain adaptation networks [21, 43, 41]. Our results in  
100 [Theorems 3.6](#) and [3.11](#) show that, in both MMD-based and adversarial domain adaptation  
101 settings, the sample complexities for the number of labeled source samples  $M_s$  and the total  
102 number of source and target samples,  $N_s$  and  $N_t$ , scale quadratically with the width  $d$  and the  
103 depth  $L$  of the network. Our results also offer insight into the optimal choice for the weight  $\alpha$   
104 of the target classification loss, indicating it should decrease at rate  $\alpha = O(\sqrt{M_t})$  to effectively  
105 handle the scarcity of labeled target samples. Our proof technique extends [Theorem 2.7](#) by  
106 thoroughly analyzing the covering numbers of the relevant function classes. To the best of  
107 our knowledge, these are the first results to provide a comprehensive characterization of the  
108 sample complexity of domain-adaptive neural networks.

109 We defer a detailed discussion of closely related literature to [Section SM2](#), where we also  
110 compare and contrast our results with previous findings. [Section 4](#) presents some simulation  
111 results for the experimental validation of our findings, and [Section 5](#) concludes the paper.  
112 A preliminary version of our study was presented in [46], which laid the groundwork for the  
113 results in [Section 2.2](#).

115      **2. General performance bounds for domain alignment.**

116      **2.1. Problem formulation.** Let  $\mathcal{X}^s$  and  $\mathcal{X}^t$  denote two compact metric spaces representing  
 117 respectively a source domain and a target domain, and let  $\mathcal{Y} \subset \mathbb{R}^m$  be a label set. Let  $\mu_s$  be a  
 118 source Borel probability measure and  $\mu_t$  be a target Borel probability measure respectively on  
 119 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  
 120 to learn two mappings (transformations)  $f^s : \mathcal{X}^s \rightarrow \mathcal{X}$  and  $f^t : \mathcal{X}^t \rightarrow \mathcal{X}$  from the source and  
 121 target domains to a common set  $\mathcal{X}$  together with a hypothesis function  $h : \mathcal{X} \rightarrow \mathcal{Y}$  estimating  
 122 class labels on  $\mathcal{X}$ . The expected losses of the transformations  $f^s$ ,  $f^t$ , and the hypothesis  $h$  at  
 123 the source and target are respectively given by

124      
$$\mathcal{L}^s(f^s, h) = \int_{\mathcal{Z}^s} \ell(h \circ f^s(x^s), \mathbf{y}^s) d\mu_s \quad \mathcal{L}^t(f^t, h) = \int_{\mathcal{Z}^t} \ell(h \circ f^t(x^t), \mathbf{y}^t) d\mu_t$$

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

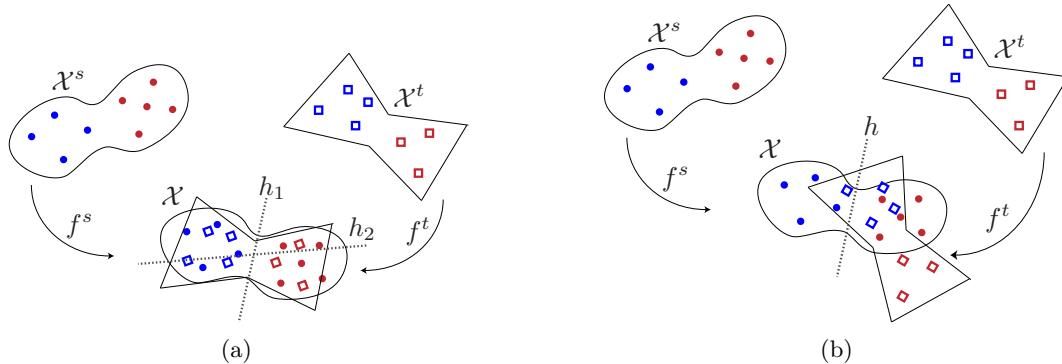
130      Let  $\{x_i^s\}_{i=1}^{N_s}$  be a set of source samples and  $\{x_j^t\}_{j=1}^{N_t}$  be a set of target samples drawn  
 131 independently from the probability measures  $\mu_s$  and  $\mu_t$ , where  $\{x_i^s\}_{i=1}^{M_s}$  are the  $M_s$  labeled  
 132 samples in the source with labels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

136      (2.1)      
$$\min_{f^s \in \mathcal{F}^s, f^t \in \mathcal{F}^t, h \in \mathcal{H}} (1 - \alpha) \hat{\mathcal{L}}^s(f^s, h) + \alpha \hat{\mathcal{L}}^t(f^t, h) + \beta \hat{D}(f^s, f^t).$$

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

141      (2.2)      
$$\hat{\mathcal{L}}^s(f^s, h) = \frac{1}{M_s} \sum_{i=1}^{M_s} \ell(h \circ f^s(x_i^s), \mathbf{y}_i^s) \quad \hat{\mathcal{L}}^t(f^t, h) = \frac{1}{M_t} \sum_{j=1}^{M_t} \ell(h \circ f^t(x_j^t), \mathbf{y}_j^t)$$

142 and the distance  $\hat{D}$  is an estimate of the distribution distance  $D(f^s, f^t)$  computed with all  
 143 (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 distri-  
 144 bution distance  $D(f^s, f^t)$  has been chosen in different ways in previous works such as the  
 145 MMD or Wasserstein distance along with the corresponding estimates  $\hat{D}(f^s, f^t)$  that lead to  
 146 practical learning algorithms. In [Section 2.2](#), we provide generalization bounds for learning  
 147 algorithms with an arbitrary distribution distance function. Then in [Section 2.3](#), we focus on  
 148 the kernel mean matching (KMM) methods in particular, and propose bounds for algorithms  
 149 using a KMM-based distribution distance.



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

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

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

$$155 \quad (2.3) \qquad \qquad \qquad |\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)| \leq R D(f^s, f^t).$$

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

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

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

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

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

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

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

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

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

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

We can now present the following result that bounds the deviation between the expected and empirical weighted losses. <sup>1</sup>

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

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

the expected target loss is bounded as

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

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

<sup>1</sup>Bu kismi tamamen revverite edip kaldırılmış bence. By Lemma SM1.2, the deviation between the expected and empirical weighted losses can be bounded in terms of the covering numbers of the composite hypothesis classes and the number of labeled samples. The proof is provided in Proof 2. We can now combine Lemmas SM1.1 and SM1.2 to bound the expected target loss in terms of the empirical weighted loss and the distribution distance in the following main result.

203 actual expected loss obtained at the target by applying the learnt transformation  $f^t$  and hy-  
204 pothesis  $h$  to target test samples cannot differ from the empirical weighted loss  $\hat{\mathcal{L}}_\alpha(f^s, f^t, h)$   
205 obtained over training samples by more than  $\epsilon$  plus an error term involving the distance  
206  $D(f^s, f^t)$ . This statement holds with probability approaching 1 at an exponential rate with  
207 the increase in number of labeled samples  $M_s$ . Note that in the very typical case where  $M_t$   
208 is limited, the target term in the probability expression (2.5) can be controlled by suitably  
209 scaling down the weight parameter  $\alpha$  proportionally to  $O(\sqrt{M_t})$ . <sup>2</sup>

210 **2.3. Generalization bounds for maximum mean discrepancy measures.** We now extend  
211 the results of Section 2.2 for a setting where the distribution discrepancy in the common  
212 domain of transformation is measured with respect to the maximum mean discrepancy (MMD)  
213 criterion. The MMD criterion is widely used in domain adaptation. In particular, a popular  
214 family of methods called kernel mean matching (KMM) algorithms aim to map the source  
215 and target data to a shared domain via a kernel function such that the distance between the  
216 source and target samples measured with respect to the MMD criterion is minimized.

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

222 Given the source and target probability measures  $\mu_s, \mu_t$  on the sets  $\mathcal{Z}^s = \mathcal{X}^s \times \mathcal{Y}$  and  
223  $\mathcal{Z}^t = \mathcal{X}^t \times \mathcal{Y}$ ; and the probability measures  $\nu_s, \nu_t$  these respectively induce over the domain  
224  $\mathcal{X}$ ; KMM algorithms characterize the distance between  $\nu_s$  and  $\nu_t$  via the MMD given by

225 (2.6) 
$$D(f^s, f^t) = \|E_{x^s}[f^s(x^s)] - E_{x^t}[f^t(x^t)]\|_{\mathcal{X}}$$

226 where  $\|\cdot\|_{\mathcal{X}}$  stands for the inner-product-induced norm in the Hilbert space  $\mathcal{X}$ . <sup>3</sup>Given the  
227 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  
228 given by

229 (2.7) 
$$\hat{D}(f^s, f^t) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} f^s(x_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} f^t(x_j^t) \right\|.$$

---

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

<sup>3</sup>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  $\mu_s$  and  $\mu_t$  in the source and the target domains, respectively. We will simply write  $E[\cdot]$  whenever the meaning is clear.

In order to study the performance of KMM algorithms, we first derive a bound on the deviation between the actual distribution discrepancy  $D(f^s, f^t)$  and its empirical estimate  $\hat{D}(f^s, f^t)$ .<sup>4</sup> We make the following assumption on the data distributions:

**Assumption 2.5.** *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; that is, there exist constants  $\sigma_s^2$  and  $\sigma_t^2$  such that*

$$(2.8) \quad E[\|f^s(x_i^s) - E[f^s(x^s)]\|^2] \leq \sigma_s^2 \quad E[\|f^t(x_j^t) - E[f^t(x^t)]\|^2] \leq \sigma_t^2.$$

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

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

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

By Lemma SM1.3, the deviation between the sample mean and the expectation of the source and target samples transformed to the shared Hilbert space  $\mathcal{X}$  can be upper bounded with probability improving at an exponential rate with  $N_s$  and  $N_t$ .<sup>5</sup> Using this result, Lemma SM1.4 establishes 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 holds for all functions in the transformation function classes  $\mathcal{F}^s$  and  $\mathcal{F}^t$  under the compactness assumption below.<sup>6</sup>

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

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

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

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

<sup>4</sup>Although most KMM methods assume  $\mathcal{X}^s = \mathcal{X}^t$  and  $f^s = f^t = \phi$ , we do not make these assumptions. We only assume that the distribution discrepancy between  $\nu_s$  and  $\nu_t$  is taken as in (2.6) and the empirical estimate is computed as in (2.7).

<sup>5</sup>The proof is provided in Appendix 3.

<sup>6</sup>The proof is provided in Appendix 4.

259 the expected target loss is upper bounded as

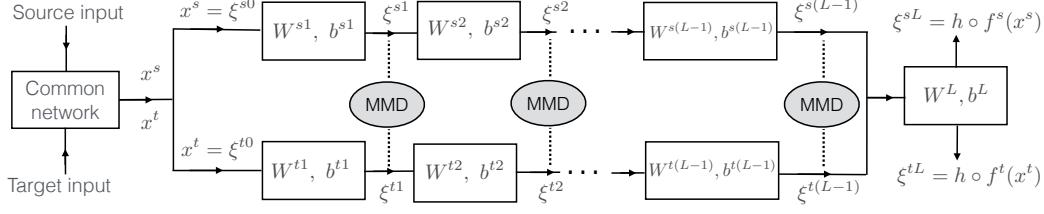
260 (2.12)  $\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1 - \alpha)R\hat{D}(f^s, f^t) + (1 - \alpha)R\epsilon + \epsilon.$

261 <sup>7</sup>

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

---

<sup>7</sup>The proof follows from [Theorem 2.4](#) and [Lemma SM1.4](#) by the union bound.



**Figure 2.** Illustration of MMD-based domain adaptation networks. Source and target samples first pass through a common network (convolutional and fully connected layers), then through domain-specific networks of  $L - 1$  fully connected layers, with the  $L$ -th layer being a shared classifier. The common network parameters are often adopted from pre-trained networks or fine-tuned using source samples [29, 44, 22]; hence we consider feature representations at its output as our domain samples.

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

275     **3.1. MMD-based domain adaptation networks.** We study the implications of Theorem  
 276     2.7 on deep domain adaptation networks that learn domain-invariant features based on  
 277     the MMD distance measure. We consider the network model depicted in Figure 2, a com-  
 278     monly adopted foundation for many MMD-based architectures. Defining  $\xi^{s0} \triangleq x^s \in \mathbb{R}^{d_0}$  and  
 279      $\xi^{t0} \triangleq x^t \in \mathbb{R}^{d_0}$ , the relation between the features of layers  $l$  and  $l - 1$  is given by

280     (3.1)      $\xi^{sl} = \eta^l(\mathbf{W}^{sl}\xi^{s(l-1)} + \mathbf{b}^{sl}) \quad \xi^{tl} = \eta^l(\mathbf{W}^{tl}\xi^{t(l-1)} + \mathbf{b}^{tl})$

281     for  $l = 1, \dots, L$ , where  $\xi^{sl}, \xi^{tl} \in \mathbb{R}^{d_l}$  are  $d_l$ -dimensional source and target features in layer  $l$ ;  
 282     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  
 283      $\mathbf{b}^{sl}, \mathbf{b}^{tl} \in \mathbb{R}^{d_l}$  are source and target bias vectors;  $\eta^l : \mathbb{R}^{d_l} \rightarrow \mathbb{R}^{d_l}$  is a nonlinear activation  
 284     function;  $L$  is the depth of the network; and  $d_l$  is the width of the network at layer  $l$ . We  
 285     assume that the parameters of the output layer  $L$  are common between the source and the  
 286     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  
 287      $m = d_L$  is the number of classes.

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

293     **Assumption 3.1.** The source and target domains are

294     (3.2)      $\mathcal{X}^s = \{x^s \in \mathbb{R}^{d_0} : \|x^s\| \leq A_x\} \quad \mathcal{X}^t = \{x^t \in \mathbb{R}^{d_0} : \|x^t\| \leq A_x\}$

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

297     (3.3)      $|\Theta_{ij}^{sl}|, |\Theta_{ij}^{tl}| \leq A_\Theta$

298 for some magnitude bound parameter  $A_\Theta > 0$ , for  $l = 1, \dots, L$  and  $i = 1, \dots, d_l$ ;  $j =$   
299  $1, \dots, d_{l-1} + 1$ .

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

305 MMD-based deep domain adaptation networks employ a feature mapping  $\phi^l : \mathbb{R}^{d_l} \rightarrow \mathcal{X}^l$   
306 between the hidden layer feature vectors  $\xi^{sl}, \xi^{tl}$  and a Reproducing Kernel Hilbert Space  
307 (RKHS)  $\mathcal{X}^l$  [29, 24]. The RKHS  $\mathcal{X}^l$  of each layer  $l$  has a symmetric, positive definite charac-  
308 teristic kernel  $k^l : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \rightarrow \mathbb{R}$  such that  $k^l(\xi_1^l, \xi_2^l) = \langle \phi^l(\xi_1^l), \phi^l(\xi_2^l) \rangle_{\mathcal{X}^l}$  for any  $\xi_1^l, \xi_2^l \in \mathbb{R}^{d_l}$ ,  
309 where  $\langle \cdot, \cdot \rangle_{\mathcal{X}^l}$  denotes the inner product in the RKHS  $\mathcal{X}^l$  [24]. The feature mapping  $\phi^l$  and the  
310 characteristic kernel  $k^l$  are related as  $\phi^l(\xi^l) = k^l(\xi^l, \cdot) : \mathbb{R}^{d_l} \rightarrow \mathbb{R}$  [24]. The feature mapping  
311  $\phi^l$  has the property that  $\langle \phi^l(\xi^l), \psi \rangle_{\mathcal{X}^l} = \psi(\xi^l)$  for any  $\psi \in \mathcal{X}^l$  and  $\xi^l \in \mathbb{R}^{d_l}$ .

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

314 (3.4) 
$$f^{sl}(x^s) \triangleq \phi^l(\xi^{sl}(x^s)) \in \mathcal{X}^l \quad f^{tl}(x^t) \triangleq \phi^l(\xi^{tl}(x^t)) \in \mathcal{X}^l$$

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

318 (3.5) 
$$\langle (f^1, \dots, f^{L-1}), (g^1, \dots, g^{L-1}) \rangle_{\mathcal{X}} = \sum_{l=1}^{L-1} \langle f^l, g^l \rangle_{\mathcal{X}^l}.$$

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

322 (3.6) 
$$\begin{aligned} \mathcal{F}^s &= \{f^s : \mathcal{X}^s \rightarrow \mathcal{X} \mid f^s(x^s) = (f_{\Theta^s}^{s1}(x^s), \dots, f_{\Theta^s}^{s(L-1)}(x^s)) \in \mathcal{X}, |\Theta_{ij}^{sl}| \leq A_\Theta, \forall i, j\} \\ \mathcal{F}^t &= \{f^t : \mathcal{X}^t \rightarrow \mathcal{X} \mid f^t(x^t) = (f_{\Theta^t}^{t1}(x^t), \dots, f_{\Theta^t}^{t(L-1)}(x^t)) \in \mathcal{X}, |\Theta_{ij}^{tl}| \leq A_\Theta, \forall i, j\} \end{aligned}$$

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

327 (3.7) 
$$g^s(x^s) = (h \circ f^s)(x^s) \triangleq \xi^{sL}(x^s) \quad g^t(x^t) = (h \circ f^t)(x^t) \triangleq \xi^{tL}(x^t).$$

328 Let us also define the corresponding function spaces

329 (3.8) 
$$\begin{aligned} \mathcal{G}^s &= \mathcal{H} \circ \mathcal{F}^s = \{g^s : \mathcal{X}^s \rightarrow \mathcal{Y} \mid g^s(x^s) = \xi_{\Theta^s}^{sL}(x^s) \in \mathcal{Y} \subset \mathbb{R}^m, |\Theta_{ij}^{sl}| \leq A_\Theta, \forall i, j\} \\ \mathcal{G}^t &= \mathcal{H} \circ \mathcal{F}^t = \{g^t : \mathcal{X}^t \rightarrow \mathcal{Y} \mid g^t(x^t) = \xi_{\Theta^t}^{tL}(x^t) \in \mathcal{Y} \subset \mathbb{R}^m, |\Theta_{ij}^{tl}| \leq A_\Theta, \forall i, j\}. \end{aligned}$$

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

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

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

335     We next revisit the distribution discrepancy definition in [Section 2.3](#) for MMD-based  
336     neural networks. Let us define the distribution discrepancy in layer  $l$  as  $D^l(f^{sl}, f^{tl}) \triangleq$   
337      $\|E_{x^s}[f^{sl}(x^s)] - E_{x^t}[f^{tl}(x^t)]\|_{\mathcal{X}^l}$ . MMD-based domain adaptation algorithms typically seek to  
338     minimize the empirical estimate  $\hat{D}^l$  of  $D^l$  at each layer [29, 44, 22]. The empirical distribution  
339     discrepancy  $\hat{D}^l$  is obtained from the source and target sample sets  $\{x_i^s\}_{i=1}^{N_s}$  and  $\{x_j^t\}_{j=1}^{N_t}$  as

$$\begin{aligned} 340 \quad (3.9) \quad (\hat{D}^l)^2(f^{sl}, f^{tl}) &= \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} f^{sl}(x_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} f^{tl}(x_j^t) \right\|_{\mathcal{X}^l}^2 \\ &= \frac{1}{N_s^2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} k^l(\xi_i^{sl}, \xi_j^{sl}) - \frac{2}{N_s N_t} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} k^l(\xi_i^{sl}, \xi_j^{tl}) + \frac{1}{N_t^2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} k^l(\xi_i^{tl}, \xi_j^{tl}) \end{aligned}$$

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

$$345 \quad D(f^s, f^t) = \|E_{x^s}[f^s(x^s)] - E_{x^t}[f^t(x^t)]\|_{\mathcal{X}}$$

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

$$\begin{aligned} 348 \quad (3.10) \quad \hat{D}^2(f^s, f^t) &= \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} f^s(x_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} f^t(x_j^t) \right\|_{\mathcal{X}}^2 \\ &= \frac{1}{N_s^2} \sum_{i,j=1}^{N_s} \langle f^s(x_i^s), f^s(x_j^s) \rangle_{\mathcal{X}} - \frac{2}{N_s N_t} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \langle f^s(x_i^s), f^t(x_j^t) \rangle_{\mathcal{X}} \\ &\quad + \frac{1}{N_t^2} \sum_{i,j=1}^{N_t} \langle f^t(x_i^t), f^t(x_j^t) \rangle_{\mathcal{X}} \\ &= \sum_{l=1}^{L-1} (\hat{D}^l)^2(f^{sl}, f^{tl}). \end{aligned}$$

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

350     Most MMD-based deep domain adaptation networks rely on aligning the source and the  
351     target domains by minimizing the total MMD distance (3.10) summed over all layers [47, 29,  
352     44, 22]. We thus consider a learning algorithm that minimizes the overall loss

$$353 \quad (3.11) \quad \min_{f^s \in \mathcal{F}^s, f^t \in \mathcal{F}^t, h \in \mathcal{H}} (1-\alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha\hat{\mathcal{L}}^t(f^t, h) + \beta \sum_{l=1}^{L-1} (\hat{D}^l)^2(f^{sl}, f^{tl}).$$

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

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

360 (3.12)  $|k^l(\boldsymbol{\xi}_1, \boldsymbol{\xi}) - k^l(\boldsymbol{\xi}_2, \boldsymbol{\xi})| \leq L_K \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|$

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

363 (3.13)  $\|\eta^l(\mathbf{u}) - \eta^l(\mathbf{v})\| \leq L_\eta \|\mathbf{u} - \mathbf{v}\|$

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

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

368 (3.14)  $|\eta_i^l(\mathbf{u})| \leq C_\eta$

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

371 (3.15)  $\|\eta^l(\mathbf{u})\| \leq A_\eta \|\mathbf{u}\|$

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

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

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

389     **Corollary 3.5.** Consider that the feature dimensions  $d_l$  are such that  $d_l = O(d)$  for  $l =$   
 390      $1, \dots, L$ , for some common network width parameter  $d$ . Then, the rate of growth of the  
 391     covering numbers for the function spaces  $\mathcal{N}(\mathcal{F}^s, \epsilon, \mathfrak{d}_{\mathcal{X}}^s)$ ,  $\mathcal{N}(\mathcal{F}^t, \epsilon, \mathfrak{d}_{\mathcal{X}}^t)$ ,  $\mathcal{N}(\mathcal{H} \circ \mathcal{F}^s, \epsilon, \mathfrak{d}^s)$ ,  $\mathcal{N}(\mathcal{H} \circ$   
 392      $\mathcal{F}^t, \epsilon, \mathfrak{d}^t)$  with the width  $d$  and the depth  $L$  of the network is upper bounded by

$$393 \quad O\left(\left(\frac{L}{\epsilon}\right)^{d^2 L} (cd)^{d^2 L^2}\right)$$

394 where  $c$  denotes a constant.

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

399     **Theorem 3.6.** Consider a learning algorithm relying on the minimization of a loss function  
 400 of the form (3.11) via an MMD-based domain adaptation network. Assume that the classifi-  
 401 cation loss function  $\ell$  is bounded by a constant  $A_\ell$  and Lipschitz continuous with respect to the  
 402 first argument with constant  $L_\ell$ . Suppose that the source and target data distributions satisfy  
 403 [Assumptions 2.1](#) and [2.5](#). Assume also that the network parameters, activation functions and  
 404 the kernels satisfy [Assumptions 3.1-3.4](#). Consider that the weight parameter  $\alpha$  in the loss  
 405 function is chosen such that

$$406 \quad \alpha = O\left(\left(\frac{M_t \epsilon^2}{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}\right)^{1/2}\right)$$

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

$$409 \quad (3.16) \quad \mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1 - \alpha)R\hat{D}(f^s, f^t) + (1 - \alpha)R\epsilon + \epsilon,$$

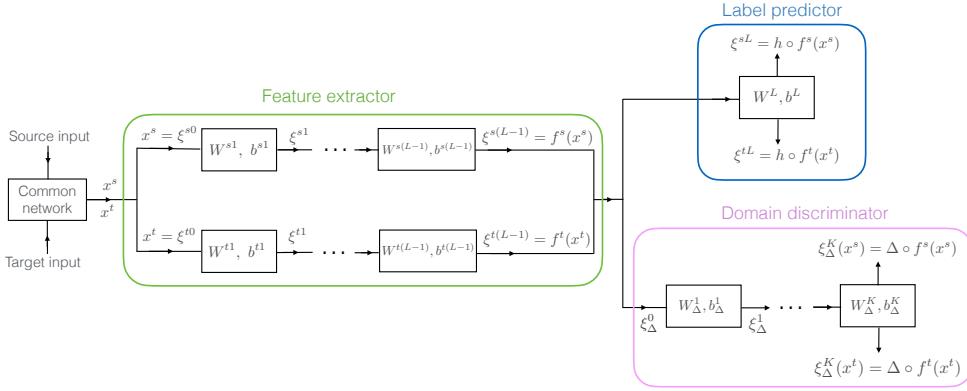
410 the sample complexities in terms of the number  $M_s$  of labeled source samples, the number  $N_s$   
 411 of all (labeled and unlabeled) source samples, and the number  $N_t$  of all target samples are  
 412 upper bounded by

$$413 \quad (3.17) \quad O\left(\frac{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right).$$

414     Theorem 3.6 shows that sample complexities  $M_s$ ,  $N_s$ , and  $N_t$  must increase at rate  $O(d^2 L^2)$   
 415 as network depth  $L$  and width  $d$  increase (ignoring logarithmic terms), indicating quadratic  
 416 growth with network size to prevent overfitting.<sup>8</sup> For limited labeled target samples  $M_t$ ,  
 417 the weight  $\alpha$  must shrink at rate  $\alpha = O(\sqrt{M_t})$  to avoid overfitting, and similarly at rate  
 418  $\alpha = O((dL)^{-1})$  as network size grows. Sample sizes scale as  $O(\epsilon^{-2})$  for an  $O(\epsilon)$  bound on loss  
 419 difference.

---

<sup>8</sup>The assumption of the existence of constants  $A_\ell$  and  $L_\ell$  is satisfied in many settings; we derive these for cross-entropy loss in [Appendix SM7](#).



**Figure 3.** Illustration of adversarial domain adaptation networks

420     **3.2. Adversarial domain adaptation networks.** In this section, we extend our results  
421 to analyze the sample complexity of adversarial domain adaptation networks. Domain-  
422 adversarial neural networks aim to compute domain-invariant representations  $f^s : \mathcal{X}^s \rightarrow \mathcal{X}$ ,  
423  $f^t : \mathcal{X}^t \rightarrow \mathcal{X}$  through a feature extractor network, followed by a label predictor  $h : \mathcal{X} \rightarrow \mathcal{Y}$   
424 (Figure 3). The domain-invariance of the learnt features is ensured by a domain discrimi-  
425 nator network trained to distinguish source from target features. The feature extractor and  
426 discriminator are trained adversarially: the extractor learns representations indistin-  
427 guishable to the discriminator. The domain discriminator  $\Delta : \mathcal{X} \rightarrow \mathbb{R}$  minimizes the domain  
428 discrimination loss  $\mathcal{L}_D(f^s, \Delta) + \mathcal{L}_D(f^t, \Delta)$  where  $\mathcal{L}_D(f^s, \Delta) = E[\ell_D(\Delta \circ f^s(x^s), l^s)]$ , and  
429  $\mathcal{L}_D(f^t, \Delta) = E[\ell_D(\Delta \circ f^t(x^t), l^t)]$  respectively denote the expected domain discrimination  
430 losses in the source and the target domains;  $\ell_D : \mathbb{R} \times \mathbb{R} \rightarrow [0, \infty]$  is a domain discrimination  
431 loss function; and  $l^s, l^t \in \mathbb{R}$  denote the domain labels of the source and the target domains.  
432 It is common practice to set the domain discrimination loss  $\ell_D$  as a logarithmic penalty on  
433 the deviation between the estimated domain labels and the true domain labels  $l^s = 0, l^t = 1$   
434 as [21, 43, 30]

435     (3.18)                  
$$\begin{aligned} \ell_D(\Delta \circ f^s(x^s), l^s) &= -\log(1 - \Delta \circ f^s(x^s)) \\ \ell_D(\Delta \circ f^t(x^t), l^t) &= -\log(\Delta \circ f^t(x^t)). \end{aligned}$$

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

438     (3.19)                  
$$\min_{f^s, f^t, h, \Delta} (1 - \alpha)\hat{\mathcal{L}}^s(f^s, h) + \alpha\hat{\mathcal{L}}^t(f^t, h) - \beta(\hat{\mathcal{L}}_D^s(f^s, \Delta) + \hat{\mathcal{L}}_D^t(f^t, \Delta))$$

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

441                  
$$\hat{\mathcal{L}}_D^s(f^s, \Delta) = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell_D(\Delta \circ f^s(x_i^s), l_i^s) \quad \hat{\mathcal{L}}_D^t(f^t, \Delta) = \frac{1}{N_t} \sum_{j=1}^{N_t} \ell_D(\Delta \circ f^t(x_j^t), l_j^t).$$

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

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

$$447 \quad (3.20) \quad \begin{aligned} \mathcal{F}^s &= \{f^s : \mathcal{X}^s \rightarrow \mathbb{R}^{d_{L-1}} \mid f^s(x^s) = \xi_{\Theta^s}^{s(L-1)}(x^s), |\Theta_{ij}^{sl}| \leq A_\Theta, \forall i, j\} \\ \mathcal{F}^t &= \{f^t : \mathcal{X}^t \rightarrow \mathbb{R}^{d_{L-1}} \mid f^t(x^t) = \xi_{\Theta^t}^{t(L-1)}(x^t), |\Theta_{ij}^{tl}| \leq A_\Theta, \forall i, j\}. \end{aligned}$$

448 Similarly, the hypotheses  $h \circ f^s$  and  $h \circ f^t$  are given by the output of the last layer  $L$   $h \circ f^s(x^s) =$   
 449  $\xi^{sL}(x^s)$ , and  $h \circ f^t(x^t) = \xi^{tL}(x^t)$  with the function spaces  $\mathcal{H} \circ \mathcal{F}^s$  and  $\mathcal{H} \circ \mathcal{F}^t$  defined<sup>9</sup> in (3.8).  
 450 Here, the features between layers  $l - 1$  and  $l$  are related as in (3.1) through the network  
 451 parameters  $\mathbf{W}^{sl}, \mathbf{W}^{tl}, \mathbf{b}^{sl}, \mathbf{b}^{tl}$  and the nonlinear activation functions  $\eta^l$ .<sup>10</sup>

452 The domain discriminator network typically consists of several fully connected layers [21,  
 453 43]. 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  
 454 relation between the responses  $\xi_\Delta^{l-1} \in \mathbb{R}^{d_{l-1}^\Delta}, \xi_\Delta^l \in \mathbb{R}^{d_l^\Delta}$  at layers  $l - 1$  and  $l$  is given by  $\xi_\Delta^l =$   
 455  $\eta_\Delta^l(\mathbf{W}_\Delta^l \xi_\Delta^{l-1} + \mathbf{b}_\Delta^l)$  for  $l = 1, \dots, K$ , where  $K$  denotes the number of layers and  $\eta_\Delta^l : \mathbb{R}^{d_l^\Delta} \rightarrow \mathbb{R}^{d_l^\Delta}$   
 456 denotes the activation function of the domain discriminator network at layer  $l$ . Here, the input  
 457  $\xi_\Delta^0$  to the domain discriminator network corresponds to the outputs  $\xi^{s(L-1)}, \xi^{t(L-1)}$  of the  
 458 feature extractor networks. The domain discriminator output is then given by  $\Delta \circ f^s(x^s) =$   
 459  $\xi_\Delta^K(x^s)$ , and  $\Delta \circ f^t(x^t) = \xi_\Delta^K(x^t)$  for the source and the target domains, where the dimension  
 460 of the output layer of the domain discriminator is  $d_K^\Delta = 1$ . Still using Assumption 3.1 and  
 461 extending it to the domain discriminator network as well, we define the function class of  
 462 domain discriminators with bounded network weights as

$$463 \quad (3.21) \quad \mathcal{D} = \{\Delta : \mathbb{R}^{d_{L-1}} \rightarrow \mathbb{R} \mid \Delta(\xi_\Delta^0) = \xi_\Delta^K, |(\mathbf{W}_\Delta^l)_{ij}| \leq A_\Theta, |(\mathbf{b}_\Delta^l)_i| \leq A_\Theta, \forall i, j\}.$$

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

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

---

<sup>9</sup>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.

<sup>10</sup>Buraya bir ayar verilecek. While feature extractor networks typically consist of several convolutional layers followed by fully connected layers in many common architectures [37]; 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 [43]. 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.

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

468 **Assumption 3.7.** *The domain discriminator output is bounded, i.e., there exists a constant  
469  $C_D > 0$  such that  $|\Delta(\xi_\Delta^0)| = |\xi_\Delta^K| \leq C_D$  for all  $\xi_\Delta^0 \in \mathbb{R}^{d_{L-1}}$ .*

470 Note that **Assumption 3.7** is satisfied for many domain-adversarial networks, as the activation  
471 function  $\eta_\Delta^K$  of the final domain discriminator layer is often selected as a bounded function  
472 such as the sigmoid [21] or the softmax function [42]. Let us denote the composition of the  
473 domain discriminator and the feature extractor as  $v^s(x^s) \triangleq \Delta \circ f^s(x^s)$ ,  $v^t(x^t) \triangleq \Delta \circ f^t(x^t)$ ,  
474 and the corresponding function spaces as  $\mathcal{V}^s = \mathcal{D} \circ \mathcal{F}^s = \{v^s : v^s = \Delta \circ f^s, \Delta \in \mathcal{D}, f^s \in \mathcal{F}^s\}$ ,  
475 and  $\mathcal{V}^t = \mathcal{D} \circ \mathcal{F}^t = \{v^t : v^t = \Delta \circ f^t, \Delta \in \mathcal{D}, f^t \in \mathcal{F}^t\}$ .

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

$$479 \hat{D}_\Delta(f^s, f^t) = \left| \frac{1}{N_s} \sum_{i=1}^{N_s} \Delta \circ f^s(x_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} \Delta \circ f^t(x_j^t) \right|$$

480 in **Lemma SM1.9**, which is the counterpart of **Lemma SM1.4** in the domain-adversarial set-  
481 ting.<sup>11</sup> Before stating the main result of this section, we formalize the following conditions.

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

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

490 Note that **Assumption 3.8** is an adaptation of the conditions in Assumptions 3.2 and  
491 3.3 to the domain-adversarial setting in consideration. Similarly, **Assumption 3.9** simply  
492 adapts the condition in **Assumption 3.4** to the domain discriminator network. We lastly make  
493 the following assumption about the link between the distribution distance and the deviation  
494 between the source and target losses.

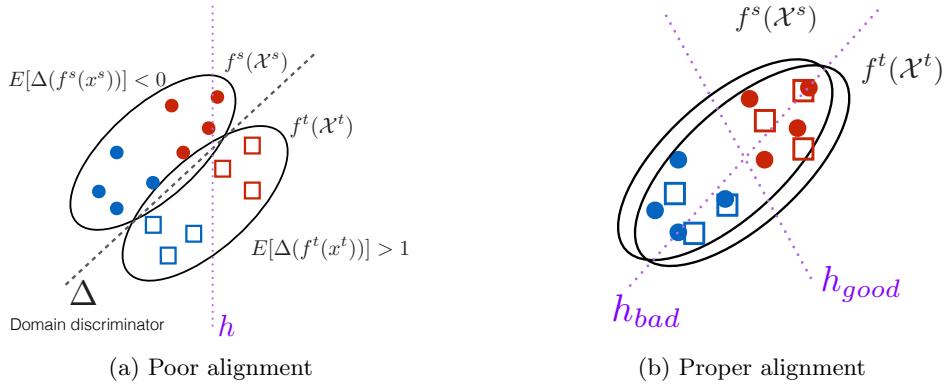
495 **Assumption 3.10.** *There exists a constant  $R_A > 0$  such that, for the domain discriminator  
496  $\Delta \in \mathcal{D}$  learnt by the algorithm, we have*

$$497 |\mathcal{L}^s(f^s, h) - \mathcal{L}^t(f^t, h)| \leq R_A D_\Delta(f^s, f^t)$$

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

---

<sup>11</sup>The proof is presented in **Proof 9**.



**Figure 4.** Illustration of Assumption 3.10. Red and blue colors represent two different classes in the source and target domains. In (a), the two domains are poorly aligned by the mappings  $f^s$  and  $f^t$ , therefore, the algorithm learns a domain discriminator  $\Delta$  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  $h$ .

499 Assumption 3.10 is the counterpart of Assumption 2.1 in the context of adversarial domain  
 500 adaptation networks, which is illustrated in Figure 4. The assumption asserts that the source  
 501 and the target distributions be related in such a way that, when efficiently aligned via the  
 502 feature mappings  $f^s$  and  $f^t$  so as to minimize the domain discrepancy  $D_\Delta(f^s, f^t)$ , the classi-  
 503 fication losses arising in the source and the target domains are also comparable. <sup>12</sup>

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

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

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

$$15 \quad (3.22) \quad \alpha = O \left( \left( \frac{M_t \epsilon^2}{d^2 L \log \left( \frac{L}{\epsilon} \right) + d^2 L^2 \log(d)} \right)^{1/2} \right)$$

<sup>12</sup>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  $\Delta$  is able to distinguish between the source and target domains if the mappings  $f^s$  and  $f^t$  result in poor feature alignment.

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

518 (3.23) 
$$\mathcal{L}^t(f^t, h) \leq \hat{\mathcal{L}}_\alpha(f^s, f^t, h) + (1 - \alpha)R_A\hat{D}_\Delta(f^s, f^t) + (1 - \alpha)R_A\epsilon + \epsilon,$$

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

522 (3.24) 
$$M_s = O\left(\frac{d^2 L \log\left(\frac{L}{\epsilon}\right) + d^2 L^2 \log(d)}{\epsilon^2}\right)$$
  

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

523 The proof of [Theorem 3.11](#) is presented in [Section SM8](#). The findings of [Theorem 3.11](#) on  
524 the sample complexity of domain-adversarial networks are in line with those of [Theorem 3.6](#),  
525 which studied MMD-based networks. The optimal choice for the weight parameter  $\alpha$  scales as  
526  $O(\sqrt{M_t})$  as the number of labeled target samples varies. In order to prevent overfitting,  $M_s$   
527 must increase at rate  $M_s = O(d^2 L^2)$  with  $d$  and  $L$ , which indicates that the number of labeled  
528 source samples must increase quadratically with the width  $d$  and the depth  $L$  of the feature  
529 extractor network, ignoring the logarithmic factors. Likewise, the number of source and target  
530 samples  $N_s$  and  $N_t$  must also increase at a quadratic rate  $O(d^2(L+K)^2)$  with the width  $d$  and  
531 the depth  $L + K$  of the combination of feature extractor and domain discriminator networks,  
532 in order to avoid overfitting to the empirical domain discrimination loss of training samples.  
533 Similarly to the result in [Theorem 3.6](#), for the difference between the expected target loss and  
534 the sum of the empirical losses to be bounded by an amount of  $O(\epsilon)$ , the number of samples  
535  $M_s, N_s, N_t$  must scale at rate  $O(\epsilon^{-2})$ .<sup>13</sup>

---

<sup>13</sup>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  $\mathcal{N}(\mathcal{V}^s, \epsilon, \mathfrak{d}^s)$ ,  $\mathcal{N}(\mathcal{V}^t, \epsilon, \mathfrak{d}^t)$  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  $\alpha$  in [\(3.22\)](#) can similarly be obtained by replacing the number of layers  $L$  with  $L+P$  in this case.

536     **4. Experimental results.** In this section, we present experimental results for the verifi-  
 537     cation of the proposed generalization bounds. We study the generic bounds using shallow  
 538     classifier models, then examine the sample complexity of domain-adaptive neural networks.  
 539     Complete experimental details are provided in [Section SM3](#) of the supplement.

540     **4.1. General domain alignment methods.** We validate our findings on a synthetic data  
 541     set with two classes.<sup>14</sup>

542       Figure 5 confirms our theoretical predictions: target error decreases at rate  $O(\sqrt{1/M_t})$   
 543     with labeled target samples (panel a), larger  $M_t$  values favor larger  $\alpha$  values supporting  $\alpha =$   
 544      $O(\sqrt{M_t})$ , and misclassification rate increases linearly with transformation estimation error  $\tau$   
 545     proportional to  $D(f^s, f^t)$  (panel b).

546       We next experiment on the MIT-CBCL face data set [32] ([Figure 6](#)).<sup>15</sup> [Figure 7](#) shows  
 547     misclassification rates decrease at rates  $O(\sqrt{1/M_t})$  and  $O(\sqrt{1/M_s})$  with increasing labeled  
 548     samples, confirming our theory.

549     **4.2. Domain-adaptive neural networks.** We experimentally verify our sample complexity  
 550     results (Theorems 3.6, 3.11) using MNIST  $\rightarrow$  MNIST-M experiments [28, 20].<sup>16</sup>

551       **4.2.1. MMD-based domain adaptation networks.** We adopt the MMD-based architec-  
 552     ture from [29].<sup>17</sup> Figures 8-9 demonstrate the predicted quadratic growth  $M_s, N_s = O(L^2)$   
 553     and  $M_s = O(d^2)$  with respect to network depth and width.

554       Figure 11 confirms the theoretical prediction that  $\alpha_{opt} = O(\sqrt{M_t})$  from [Theorem 3.6](#).

555       **4.2.2. Adversarial domain adaptation networks.** We adopt the domain-adversarial ar-  
 556     chitecture from [21].<sup>18</sup>

557       Figures 10-13 confirm [Theorem 3.11](#): required sample sizes grow quadratically as  $M_s, N_s =$   
 558      $O(L^2)$  and  $M_s, N_s = O(d^2)$  with respect to depth and width,<sup>19</sup> and the optimal weight scales  
 559     as  $\alpha_{opt} = O(\sqrt{M_t})$ .

<sup>14</sup>Source and target data are generated by applying different geometric transformations to 400 samples from the standard normal distribution in  $\mathbb{R}^2$ . We emulate a setting where the transformations  $f^s$  and  $f^t$  are learnt with some estimation error  $\tau$ . The classifier is a regularized ridge regression trained in the common domain. Target misclassification rates are evaluated over 1000 test samples. Complete setup is provided in [Section SM3.1](#).

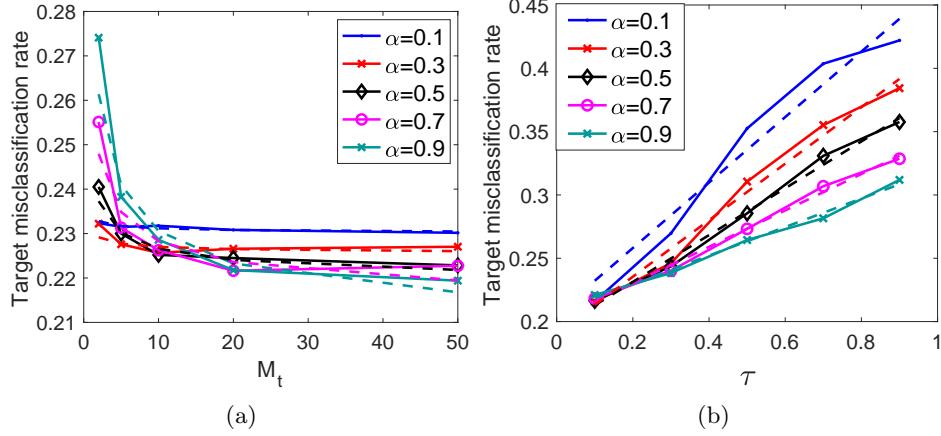
<sup>15</sup>The dataset contains 3240 synthetic face images of 10 subjects under different poses and illumination conditions. We use Pose 1 (frontal) as source and Poses 2, 5, 9 as targets in separate trials. Domain alignment uses the PCA-based method of [18] and an SVM classifier. Complete setup in [Section SM3.1](#).

<sup>16</sup>MNIST (60000 images) serves as source and MNIST-M (59000 colored background images) as target. Networks are trained with varying numbers of labeled and unlabeled samples. Hyperparameters are chosen to maintain overfitting regime for sample complexity characterization. Complete details in [Section SM3.2](#).

<sup>17</sup>The network uses convolutional layers followed by fully connected MMD layers with coupled parameters, minimizing cross-entropy loss plus MMD distance across all layers. In [Figure 8](#), we vary the number  $L$  of MMD layers in overfitting regime. Left panels show accuracy vs  $L$  for different sample sizes; right panels plot minimum samples required to achieve reference accuracy, with fitted quadratic polynomials.

<sup>18</sup>The architecture trains a feature extractor and label predictor adversarially against a domain discriminator. The feature extractor produces domain-invariant representations to fool the discriminator, which tries to distinguish source from target features. We use negative log likelihood for both classification and domain discrimination losses.

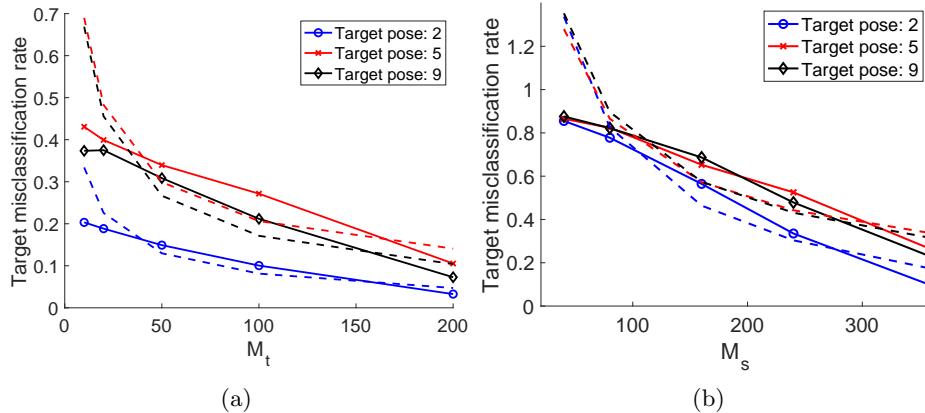
<sup>19</sup>When studying  $M_s$ , we scale feature extractor and label predictor depths together; for  $N_s$ , we scale feature extractor and domain discriminator depths together. Width parameter  $d$  scales all layer widths proportionally.



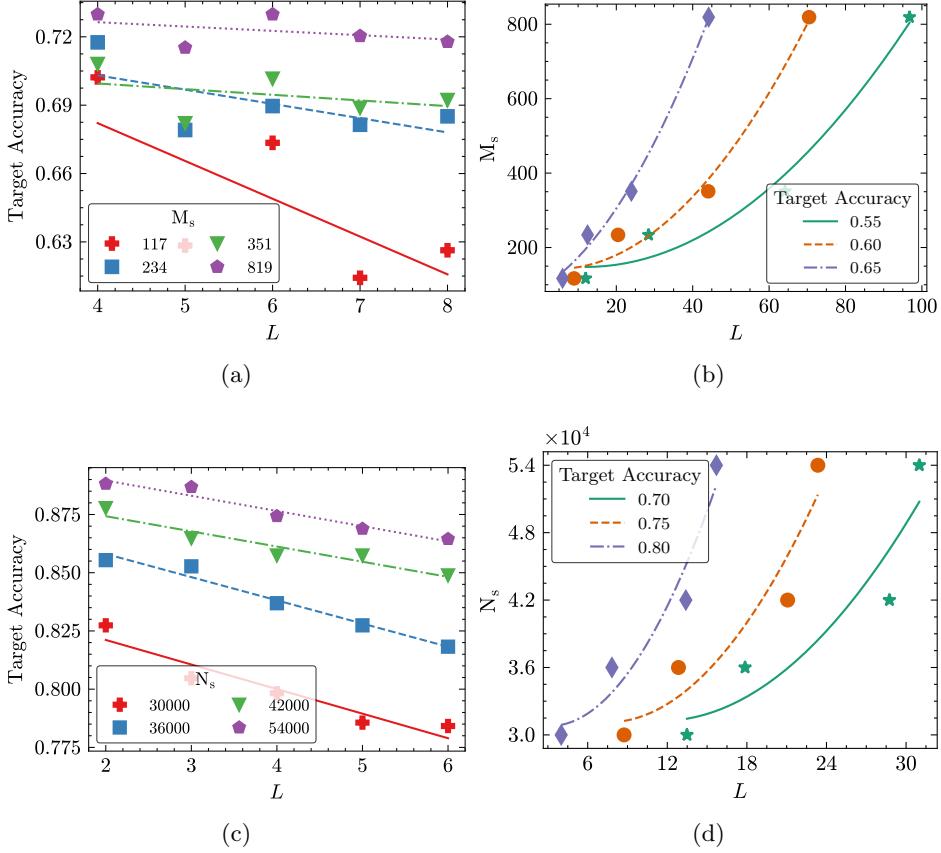
**Figure 5.** Variation of the target error on synthetical data with (a) Number of labeled target samples, (b) Distribution distance after transformation. Solid lines indicate experimental data and dashed lines represent theoretical rates of variation.



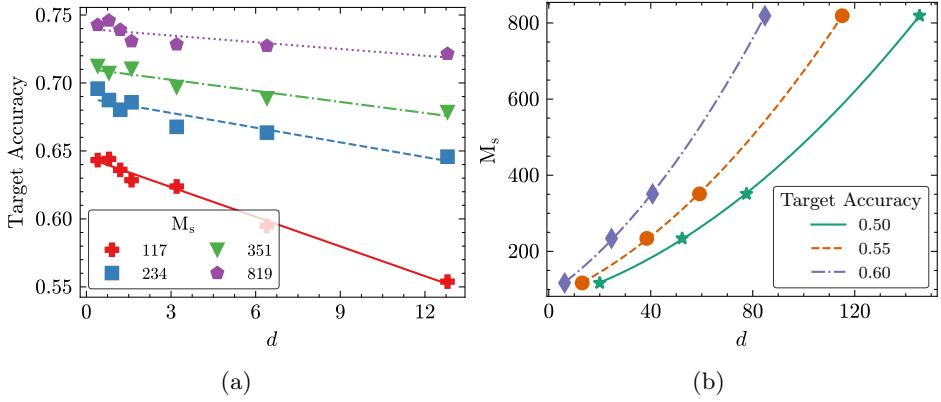
**Figure 6.** Sample images from the MIT-CBCL face data set for four different subjects, rendered respectively under poses 1, 2, 5, and 9 for various illumination conditions.



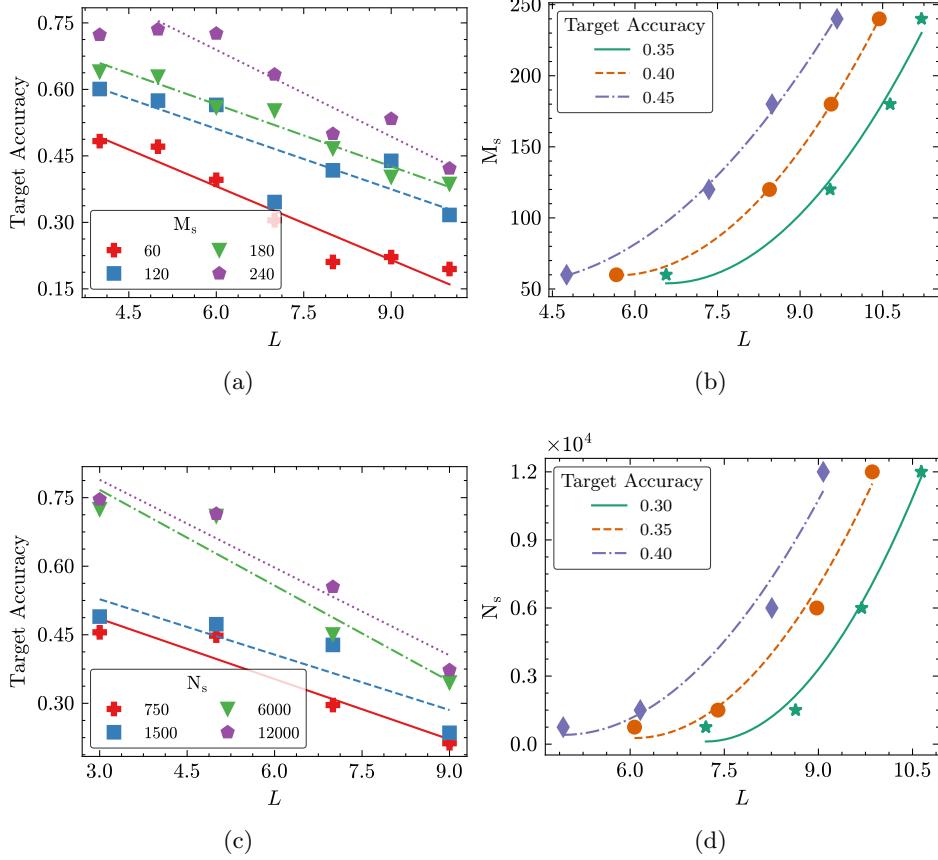
**Figure 7.** Variation of the target error on MIT-CBCL face data with (a) Number of labeled target samples, (b) Number of labeled source samples. Solid lines indicate experimental data and dashed lines represent theoretical rates of variation.



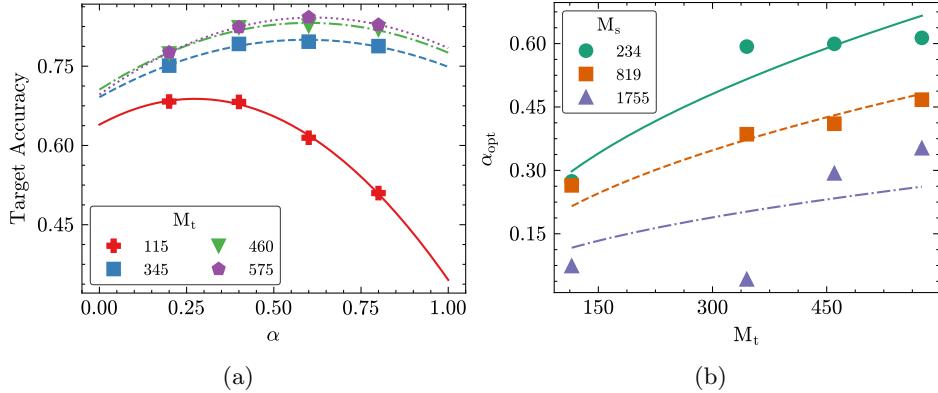
**Figure 8.** Sample complexity with respect to depth  $L$  for MMD-based networks [29]. Left panels show target accuracy variation with  $L$  at different sample sizes. Right panels show quadratic growth  $M_s, N_s = O(L^2)$ .



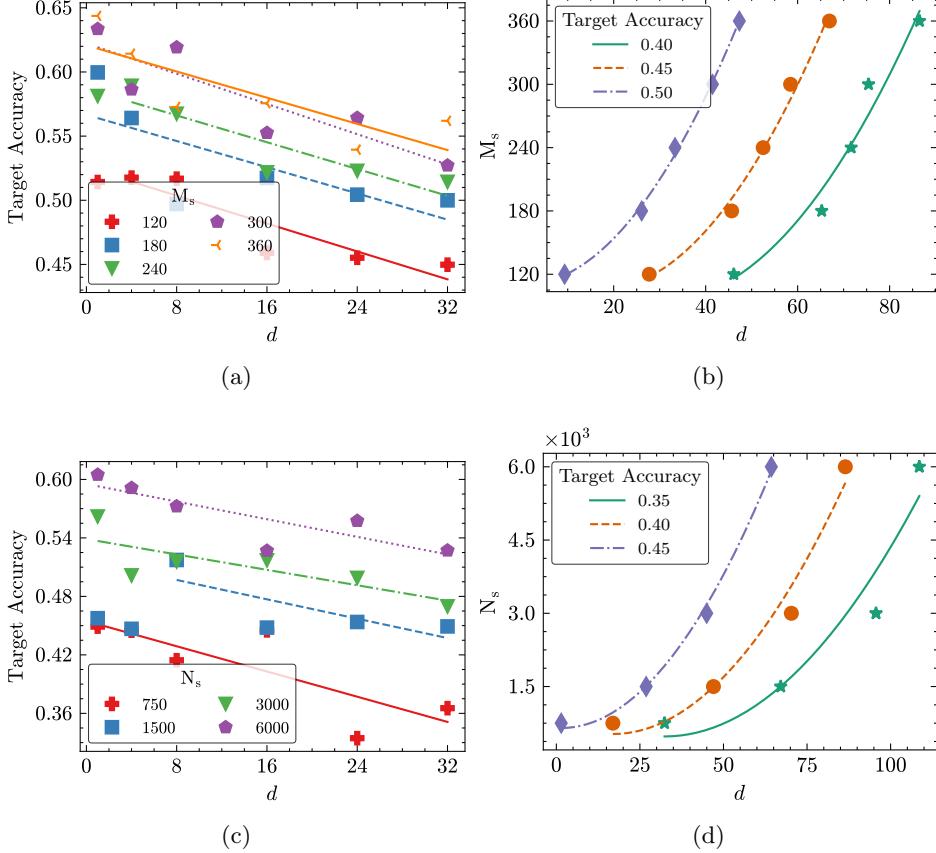
**Figure 9.** Sample complexity with respect to width  $d$  for MMD-based networks. Quadratic growth  $M_s = O(d^2)$  confirmed.



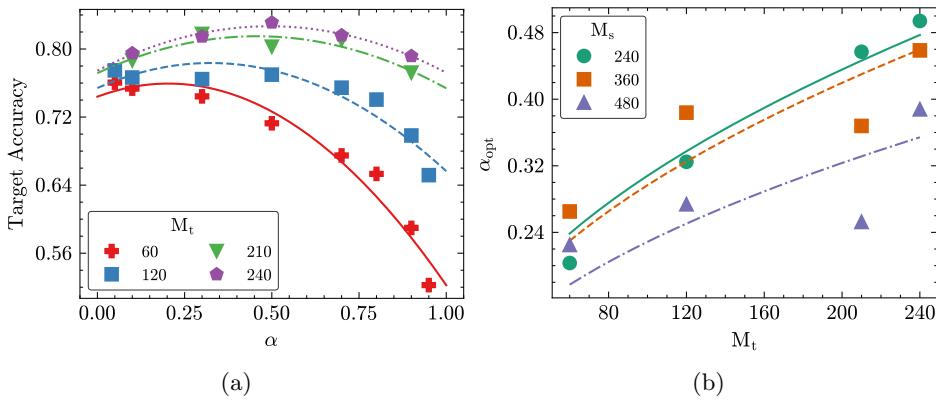
**Figure 10.** Sample complexity with respect to depth  $L$  for adversarial networks. Left: accuracy vs depth. Right: quadratic growth  $M_s, N_s = O(L^2)$ .



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



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



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

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

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

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

581     **Relation to prior work.**<sup>20</sup> Previous theoretical analyses of domain adaptation have  
582     primarily focused on how domain discrepancy affects generalization when learning a classifier  
583     in the original source and target domains, without considering domain-aligning transfor-  
584     mations [5, 4, 31]. These works establish generalization bounds in terms of VC-dimensions or  
585     Rademacher complexities of hypothesis classes, combined with various distribution divergence  
586     measures. While theoretically insightful, many of these divergence measures are difficult to  
587     estimate in practice. In contrast, our results in Theorems 2.7-3.11 provide practical gener-  
588     alization bounds based on empirical losses and distribution distances computed directly on  
589     aligned training data. Research on neural network sample complexity in single-domain set-  
590     tings [34, 51] has shown dependencies on network size; our work extends these insights to  
591     the domain adaptation setting, demonstrating quadratic scaling with both network depth and  
592     width.

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

---

<sup>20</sup>A detailed discussion is provided in the supplement.

## REFERENCES

- [1] P. L. ANTHONY, M. BARTLETT, *Neural Network Learning - Theoretical Foundations*, Cambridge University Press, Cambridge, UK, 2002.
- [2] K. AZIZZADENESHELI, A. LIU, F. YANG, AND A. ANANDKUMAR, *Regularized learning for domain adaptation under label shifts*, in Int. Conf. Learning Representations, 2019.
- [3] M. BAKTASHMOTLAGH, M. T. HARANDI, B. C. LOVELL, AND M. SALZMANN, *Unsupervised domain adaptation by domain invariant projection*, in IEEE International Conference on Computer Vision, 2013, pp. 769–776.
- [4] S. BEN-DAVID, J. BLITZER, K. CRAMMER, A. KULESZA, F. PEREIRA, AND J. WORTMAN, *A theory of learning from different domains*, Machine Learning, 79 (2010), pp. 151–175.
- [5] S. BEN-DAVID, J. BLITZER, K. CRAMMER, AND F. PEREIRA, *Analysis of representations for domain adaptation*, in Proc. Advances in Neural Information Processing Systems 19, 2006, pp. 137–144.
- [6] K. BOUSMALIS ET AL., *Domain separation networks*, in Adv. Neural Information Processing Systems, 2016, pp. 343–351.
- [7] N. COURTY ET AL., *Optimal transport for domain adaptation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (2017), pp. 1853–1865.
- [8] F. CUCKER AND S. SMALE, *On the Mathematical Foundations of Learning*, Bulletin of the American Mathematical Society, 39 (2002), pp. 1–49.
- [9] B. B. DAMODARAN ET AL., *Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation*, in European Conf. Comp. Vision, vol. 11208, 2018, pp. 467–483.
- [10] A. DANIELY AND E. GRANOT, *On the sample complexity of two-layer networks: Lipschitz vs. element-wise Lipschitz activation*, in International Conference on Algorithmic Learning Theory, vol. 237, 2024, pp. 505–517.
- [11] H. DAUMÉ III, *Frustratingly easy domain adaptation*, in Annual Meeting–Association for Computational Linguistics, 2007.
- [12] H. DAUMÉ III, A. KUMAR, AND A. SAHA, *Co-regularization based semi-supervised domain adaptation*, in Proc. Advances in Neural Information Processing Systems 23, 2010, pp. 478–486.
- [13] S. DHOUB, I. REDKO, AND C. LARTIZIEN, *Margin-aware adversarial domain adaptation with optimal transport*, in Proc. Int. Conf. Machine Learning, vol. 119, 2020, pp. 2514–2524.
- [14] L. DUAN, D. XU, AND I. W. TSANG, *Learning with augmented features for heterogeneous domain adaptation*, in Proc. 29th International Conference on Machine Learning, 2012.
- [15] N. DUNFORD AND J. SCHWARTZ, *Linear Operators, Part 1: General Theory*, Wiley Classics Library, Interscience Publishers Inc., New York, 1988.
- [16] M. EL HAMRI, Y. BENNANI, AND I. FALIH, *Theoretical guarantees for domain adaptation with hierarchical optimal transport*, Mach. Learn., 114 (2025), p. 119.
- [17] Z. FANG, J. LU, F. LIU, AND G. ZHANG, *Semi-supervised heterogeneous domain adaptation: Theory and algorithms*, IEEE Trans. Pattern Anal. Mach. Intell., 45 (2023), pp. 1087–1105.
- [18] B. FERNANDO, A. HABRARD, M. SEBBAN, AND T. TUATELAARS, *Unsupervised visual domain adaptation using subspace alignment*, in IEEE International Conference on Computer Vision, 2013, pp. 2960–2967.
- [19] T. GALANTI, L. WOLF, AND T. HAZAN, *A theoretical framework for deep transfer learning*, Information and Inference: A Journal of the IMA, 5 (2016), pp. 159–209.
- [20] Y. GANIN AND V. LEMPITSKY, *Unsupervised domain adaptation by backpropagation*, in Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015, pp. 1180–1189.
- [21] Y. GANIN ET AL., *Domain-adversarial training of neural networks*, J. Mach. Learn. Res., 17 (2016), pp. 59:1–59:35.
- [22] M. GHIFARY, W. B. KLEIJN, AND M. ZHANG, *Domain adaptive neural networks for object recognition*, in Int. Conf. Artificial Intelligence, vol. 8862, 2014, pp. 898–904.
- [23] M. GHIFARY ET AL., *Deep reconstruction-classification networks for unsupervised domain adaptation*, in European Conf. Comp. Vision, vol. 9908, 2016, pp. 597–613.
- [24] 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.
- [25] J. HUANG, A. J. SMOLA, A. GRETTON, K. M. BORGWARDT, AND B. SCHÖLKOPF, *Correcting sample*

- 649 selection bias by unlabeled data, in Proc. Advances in Neural Information Processing Systems 19,  
650 2006, pp. 601–608.
- [26] 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).
- [27] 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.
- [28] Y. LE CUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.
- [29] 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.
- [30] 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.
- [31] Y. MANSOUR, M. MOHRI, AND A. ROSTAMIZADEH, *Domain adaptation: Learning bounds and algorithms*, in The 22nd Conference on Learning Theory, 2009.
- [32] MASSACHUSETTS INSTITUTE OF TECHNOLOGY, *MIT-CBCL face recognition database*. Available: <http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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).
- [37] P. SINGHAL, R. WALAMBE, S. RAMANNA, AND K. KOTECHA, *Domain adaptation: Challenges, methods, datasets, and applications*, IEEE Access, 11 (2023), pp. 6973–7020.
- [38] B. SUN AND K. SAENKO, *Deep CORAL: correlation alignment for deep domain adaptation*, in European Conf. Comp. Vision, vol. 9915, 2016, pp. 443–450.
- [39] 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.
- [40] R. TACHET DES COMBES ET AL., *Domain adaptation with conditional distribution matching and generalized label shift*, in Neural Inf. Proc. Systems, 2020.
- [41] H. TANG AND K. JIA, *Discriminative adversarial domain adaptation*, in AAAI Conference on Artificial Intelligence, 2020, pp. 5940–5947.
- [42] 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.
- [43] 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.
- [44] 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).
- [45] 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.
- [46] E. VURAL, *Generalization bounds for domain adaptation via domain transformations*, in IEEE Int. Workshop Machine Learning for Signal Processing, 2018, pp. 1–6.
- [47] M. WANG AND W. DENG, *Deep visual domain adaptation: A survey*, Neurocomputing, 312 (2018), pp. 135–153.
- [48] X. WANG AND J. SCHNEIDER, *Generalization bounds for transfer learning under model shift*, in Proc. Conf. Uncertainty in Artificial Intelligence, 2015, pp. 922–931.
- [49] Z. WANG AND Y. MAO, *On f-divergence principled domain adaptation: An improved framework*, in Advances in Neural Information Processing Systems, 2024.
- [50] 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.
- [51] C. WEI AND T. MA, *Data-dependent sample complexity of deep neural networks via Lipschitz augmentation*.

- tion, in Advances in Neural Information Processing Systems 32, 2019, pp. 9722–9733.
- [52] Z. XIA ET AL., *Meta domain adaptation approach for multi-domain ranking*, IEEE Access, 13 (2025), pp. 92921–92931.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] 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.
- [58] M. H. ZONOOZI AND V. SEYDI, *A survey on adversarial domain adaptation*, Neural Process. Lett., 55 (2023), pp. 2429–2469.
- [59] 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.