MULTIPLE SOURCE DOMAIN ADAPTATION WITH ADVERSARIAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

While domain adaptation has been actively researched in recent years, most theoretical results and algorithms focus on the single-source-single-target adaptation setting. Naive application of such algorithms on multiple source domain adaptation problem may lead to suboptimal solutions. We propose a new generalization bound for domain adaptation when there are multiple source domains with labeled instances and one target domain with unlabeled instances. Compared with existing bounds, the new bound does not require expert knowledge about the target distribution, nor the optimal combination rule for multisource domains. Interestingly, our theory also leads to an efficient learning strategy using adversarial neural networks: we show how to interpret it as learning feature representations that are invariant to the multiple domain shifts while still being discriminative for the learning task. To this end, we propose two models, both of which we call multisource domain adversarial networks (MDANs): the first model optimizes directly our bound, while the second model is a smoothed approximation of the first one, leading to a more data-efficient and task-adaptive model. The optimization tasks of both models are minimax saddle point problems that can be optimized by adversarial training. To demonstrate the effectiveness of MDANs, we conduct extensive experiments showing superior adaptation performance on three real-world datasets: sentiment analysis, digit classification, and vehicle counting.

1 INTRODUCTION

The success of machine learning algorithms has been partially attributed to rich datasets with abundant annotations (Krizhevsky et al., 2012; Hinton et al., 2012; Russakovsky et al., 2015). Unfortunately, collecting and annotating such large-scale training data is prohibitively expensive and time-consuming. To solve these limitations, different labeled datasets can be combined to build a larger one, or synthetic training data can be generated with explicit yet inexpensive annotations (Shrivastava et al., 2016). However, due to the possible shift between training and test samples, learning algorithms based on these cheaper datasets still suffer from high generalization error. Domain adaptation (DA) focuses on such problems by establishing knowledge transfer from a labeled source domain to an unlabeled target domain, and by exploring domain-invariant structures and representations to bridge the gap (Pan & Yang, 2010). Both theoretical results (Ben-David et al., 2010; Mansour et al., 2009a; Mansour & Schain, 2012; Xu & Mannor, 2012) and algorithms (Becker et al., 2013; Hoffman et al., 2012; Ajakan et al., 2014) for DA have been proposed. Recently, DA algorithms based on deep neural networks produce breakthrough performance by learning more transferable features (Glorot et al., 2011; Donahue et al., 2014; Yosinski et al., 2014; Bousmalis et al., 2016; Long et al., 2015). Most theoretical results and algorithms with respect to DA focus on the single-source-single-target adaptation setting (Ganin et al., 2016). However, in many application scenarios, the labeled data available may come from multiple domains with different distributions. As a result, naive application of the single-source-single-target DA algorithms may lead to suboptimal solutions. Such problem calls for an efficient technique for multiple source domain adaptation.

In this paper, we theoretically analyze the multiple source domain adaptation problem and propose an adversarial learning strategy based on our theoretical results. Specifically, we prove a new generalization bound for domain adaptation when there are multiple source domains with labeled instances and one target domain with unlabeled instances. Our theoretical results build on the seminal theoretical model for domain adaptation introduced by Ben-David et al. (2010), where a divergence

measure, known as the $\mathcal{H}$-divergence, was proposed to measure the distance between two distributions based on a given hypothesis space $\mathcal{H}$. Our new result generalizes the bound (Ben-David et al., 2010, Thm. 2) to the case when there are multiple source domains. The new bound has an interesting interpretation and reduces to (Ben-David et al., 2010, Thm. 2) when there is only one source domain. Technically, we derive our bound by first proposing a generalized $\mathcal{H}$-divergence measure between two sets of distributions from multi-domains. We then prove a PAC bound (Valiant, 1984) for the target risk by bounding it from empirical source risks, using tools from concentration inequalities and the VC theory (Vapnik, 1998). Compared with existing bounds, the new bound does not require expert knowledge about the target domain distribution (Mansour et al., 2009b), nor the optimal combination rule for multiple source domains (Ben-David et al., 2010). Our results also imply that it is not always beneficial to naively incorporate more source domains into training, which we verify to be true in our experiments.

Interestingly, our bound also leads to an efficient implementation using adversarial neural networks. This implementation learns both domain invariant and task discriminative feature representations under multiple domains. Specifically, we propose two models (both named MDANs) by using neural networks as rich function approximators to instantiate the generalization bound we derive (Fig. 1). After proper transformations, both models can be viewed as computationally efficient approximations of our generalization bound, so that the goal is to optimize the parameters of the networks in order to minimize the bound. The first model optimizes directly our generalization bound, while the second is a smoothed approximation of the first, leading to a more data-efficient and task-adaptive model. The optimization problem for each model is a minimax saddle point problem, which can be interpreted as a zero-sum game with two participants competing against each other to learn invariant features. Both models combine feature extraction, domain classification, and task learning in one training process. MDANs is generalization of the popular domain adversarial neural network (DANN) (Ganin et al., 2016) and reduce to it when there is only one source domain. We propose to use stochastic optimization with simultaneous updates to optimize the parameters in each iteration. To demonstrate the effectiveness of MDANs as well as the relevance of our theoretical results, we conduct extensive experiments on real-world datasets, including both natural language and vision tasks. We achieve superior adaptation performances on all the tasks, validating the effectiveness of our models.

2 PRELIMINARY

We first introduce the notation used in this paper and review a theoretical model for domain adaptation when there is only one source and one target domain (Kifer et al., 2004; Ben-David et al., 2007; Blitzer et al., 2008; Ben-David et al., 2010). The key idea is the $\mathcal{H}$-divergence to measure the discrepancy between two distributions. Other theoretical models for DA exist (Cortes et al., 2008; Mansour et al., 2009a;c; Cortes & Mohri, 2014); we choose to work with the above model because this distance measure has a particularly natural interpretation and can be well approximated using samples from both domains.

Notations We use *domain* to represent a distribution $\mathcal{D}$ on input space $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to [0, 1]$. In the setting of one source one target domain adaptation, we use $\langle \mathcal{D}_S, f_S \rangle$ and $\langle \mathcal{D}_T, f_T \rangle$ to denote the source and target domain, respectively. A *hypothesis* is a binary classification function $h : \mathcal{X} \to \{0, 1\}$. The *error* of a hypothesis h w.r.t. a labeling function f under distribution $\mathcal{D}_S$ is defined as: $\varepsilon_S(h, f) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[|h(\mathbf{x}) - f(\mathbf{x})|]$. When f is also a hypothesis, then this definition reduces to the probability that h disagrees with h under $\mathcal{D}_S$: $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[|h(\mathbf{x}) - f(\mathbf{x})|] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x}))] = \Pr_{\mathbf{x}\sim\mathcal{D}_S}(f(\mathbf{x}) \neq h(\mathbf{x}))$.

We define the *risk* of hypothesis h as the error of h w.r.t. a true labeling function under domain $\mathcal{D}_S$, i.e., $\varepsilon_S(h) := \varepsilon_S(h, f_S)$. As common notation in computational learning theory, we use $\widehat{\varepsilon}_S(h)$ to denote the empirical risk of h on the source domain. Similarly, we use $\varepsilon_T(h)$ and $\widehat{\varepsilon}_T(h)$ to mean the true risk and the empirical risk on the target domain. $\mathcal{H}$-divergence is defined as follows:

Definition 2.1. Let $\mathcal{H}$ be a hypothesis class for instance space $\mathcal{X}$, and $\mathcal{A}_\mathcal{H}$ be the collection of subsets of $\mathcal{X}$ that are the support of some hypothesis in $\mathcal{H}$, i.e., $\mathcal{A}_\mathcal{H} := \{h^{-1}(\{1\}) \mid h \in \mathcal{H}\}$. The distance between two distributions $\mathcal{D}$ and $\mathcal{D}'$ based on $\mathcal{H}$ is:

$$d_\mathcal{H}(\mathcal{D}, \mathcal{D}') := 2 \sup_{A \in \mathcal{A}_\mathcal{H}} |\Pr_{\mathcal{D}}(A) - \Pr_{\mathcal{D}'}(A)|$$

When the hypothesis class $\mathcal{H}$ contains all the possible measurable functions over $\mathcal{X}$, $d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}')$ reduces to the familiar total variation. Given a hypothesis class $\mathcal{H}$, we define its symmetric difference w.r.t. itself as: $\mathcal{H}\Delta\mathcal{H} = \{h(\mathbf{x}) \oplus h'(\mathbf{x}) \mid h, h' \in \mathcal{H}\}$, where $\oplus$ is the xor operation. Let h^* be the optimal hypothesis that achieves the minimum combined risk on both the source and the target domains:

$$h^* := \arg\min_{h \in \mathcal{H}} \varepsilon_S(h) + \varepsilon_T(h)$$

and use λ to denote the combined risk of the optimal hypothesis h^*:

$$\lambda := \varepsilon_S(h^*) + \varepsilon_T(h^*)$$

Ben-David et al. (2007) and Blitzer et al. (2008) proved the following generalization bound on the target risk in terms of the source risk and the discrepancy between the source domain and the target domain:

Theorem 2.1 ((Blitzer et al., 2008))**.** Let $\mathcal{H}$ be a hypothesis space of VC-dimension d and $\mathcal{U}_S, \mathcal{U}_T$ be unlabeled samples of size m each, drawn from $\mathcal{D}_S$ and $\mathcal{D}_T$, respectively. Let $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}$ be the empirical distance on $\mathcal{U}_S$ and $\mathcal{U}_T$; then with probability at least $1-\delta$ over the choice of samples, for each $h \in \mathcal{H}$,

$$\varepsilon_T(h) \leq \varepsilon_S(h) + \frac{1}{2}\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m) + \log(4/\delta)}{m}} + \lambda \tag{1}$$

The generalization bound depends on λ, the optimal combined risk that can be achieved by hypothesis in $\mathcal{H}$. The intuition is that if λ is large, then we cannot hope for a successful domain adaptation. One notable feature of this bound is that the empirical discrepancy distance between two samples $\mathcal{U}_S$ and $\mathcal{U}_T$ can usually be approximated by a discriminator to distinguish instances from these two domains.

3 A New Generalization Bound for Multiple Source Domain Adaptation

In this section we first generalize the definition of the discrepancy function $d_{\mathcal{H}}(\cdot,\cdot)$ that is only appropriate when we have two domains. We will then use the generalized discrepancy function to derive a generalization bound for multisource domain adaptation. We conclude this section with a discussion and comparison of our bound and existing generalization bounds for multisource domain adaptation (Mansour et al., 2009c; Ben-David et al., 2010). We refer readers to appendix for proof details and we mainly focus on discussing the interpretations and implications of the theorems.

Let $\{\mathcal{D}_{S_i}\}_{i=1}^k$ and $\mathcal{D}_T$ be k source domains and the target domain, respectively. We define the discrepancy function $d_{\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$ induced by $\mathcal{H}$ to measure the distance between $\mathcal{D}_T$ and a set of domains $\{\mathcal{D}_{S_i}\}_{i=1}^k$ as follows:

Definition 3.1.

$$d_{\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) := \max_{i\in[k]} d_{\mathcal{H}}(\mathcal{D}_T; \mathcal{D}_{S_i}) = 2\max_{i\in[k]} \sup_{A\in\mathcal{A}_{\mathcal{H}}} |\Pr_{D_T}(A) - \Pr_{D_{S_i}}(A)|$$

Again, let h^* be the optimal hypothesis that achieves the minimum combined risk:

$$h^* := \arg\min_{h\in\mathcal{H}} \left(\varepsilon_T(h) + \max_{i\in[k]} \varepsilon_{S_i}(h)\right)$$

and define

$$\lambda := \varepsilon_T(h^*) + \max_{i\in[k]} \varepsilon_{S_i}(h^*)$$

i.e., the minimum risk that is achieved by h^*. The following lemma holds for $\forall h \in \mathcal{H}$:

Theorem 3.1. $\varepsilon_T(h) \leq \max_{i\in[k]} \varepsilon_{S_i}(h) + \lambda + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$.

Remark. Let us take a closer look at the generalization bound: to make it small, the discrepancy measure between the target domain and the multiple source domains need to be small. Otherwise we cannot hope for successful adaptation by only using labeled instances from the source domains. In this case there will be no hypothesis that performs well on both the source domains and the target

domain. It is worth pointing out here that the second term and the third term together introduce a tradeoff (regularization) on the complexity of our hypothesis class $\mathcal{H}$. Namely, if $\mathcal{H}$ is too restricted, then the second term λ can be large while the discrepancy term can be small. On the other hand, if $\mathcal{H}$ is very rich, then we expect the optimal error, λ, to be small, while the discrepancy measure $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$ to be large. The first term is a standard source risk term that usually appears in generalization bounds under the PAC-learning framework (Valiant, 1984; Vapnik, 1998). Later we shall upper bound this term by its corresponding empirical risk.

The discrepancy distance $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$ is usually unknown. However, we can bound $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$ from its empirical estimation using $i.i.d.$ samples from $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$:

Theorem 3.2. Let $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ be the target distribution and k source distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class where $VC\dim(\mathcal{H}) = d$. If $\widehat{\mathcal{D}}_T$ and $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ are the empirical distributions of $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ generated with m $i.i.d.$ samples from each domain, then for $\epsilon > 0$, we have:

$$\Pr\left(\left|d_{\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) - d_{\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k)\right| \geq \epsilon\right) \leq 4k\left(\frac{em}{d}\right)^d \exp\left(-m\epsilon^2/8\right)$$

The main idea of the proof is to use VC theory (Vapnik, 1998) to reduce the infinite hypothesis space to a finite space when acting on finite samples. The theorem then follows from standard union bound and concentration inequalities. Equivalently, the following corollary holds:

Corollary 3.1. Let $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ be the target distribution and k source distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class where $VC\dim(\mathcal{H}) = d$. If $\widehat{\mathcal{D}}_T$ and $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ are the empirical distributions of $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ generated with m $i.i.d.$ samples from each domain, then, for $0 < \delta < 1$, with probability at least $1 - \delta$ (over the choice of samples), we have:

$$\left|d_{\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) - d_{\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k)\right| \leq 2\sqrt{\frac{2}{m}\left(\log\frac{4k}{\delta} + d\log\frac{em}{d}\right)}$$

Note that multiple source domains do not increase the sample complexity too drastically: it is only the square root of a log term in Corollary. 3.1 where k appears.

Similarly, we do not usually have access to the true error $\max_{i\in[k]} \varepsilon_{S_i}(h)$ on the source domains, but we can often have an estimate ($\max_{i\in[k]} \widehat{\varepsilon}_{S_i}(h)$) from training samples. We now provide a probabilistic guarantee to bound the difference between $\max_{i\in[k]} \varepsilon_{S_i}(h)$ and $\max_{i\in[k]} \widehat{\varepsilon}_{S_i}(h)$ uniformly for all $h \in \mathcal{H}$:

Theorem 3.3. Let $\{\mathcal{D}_{S_i}\}_{i=1}^k$ be k source distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class where $VC\dim(\mathcal{H}) = d$. If $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ are the empirical distributions of $\{\mathcal{D}_{S_i}\}_{i=1}^k$ generated with m $i.i.d.$ samples from each domain, then, for $\epsilon > 0$, we have:

$$\Pr\left(\sup_{h\in\mathcal{H}}\left|\max_{i\in[k]} \varepsilon_{S_i}(h) - \max_{i\in[k]} \widehat{\varepsilon}_{S_i}(h)\right| \geq \epsilon\right) \leq 2k\left(\frac{me}{d}\right)^d \exp(-2m\epsilon^2)$$

Again, Thm. 3.3 can be proved by a combination of concentration inequalities and a reduction from infinite space to finite space, along with the subadditivity of the $\max$ function. Equivalently, we have the following corollary hold:

Corollary 3.2. Let $\{\mathcal{D}_{S_i}\}_{i=1}^k$ be k source distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class where $VC\dim(\mathcal{H}) = d$. If $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ are the empirical distributions of $\{\mathcal{D}_{S_i}\}_{i=1}^k$ generated with m $i.i.d.$ samples from each domain, then, for $0 < \delta < 1$, with probability at least $1 - \delta$ (over the choice of samples), we have:

$$\sup_{h\in\mathcal{H}}\left|\max_{i\in[k]} \varepsilon_{S_i}(h) - \max_{i\in[k]} \widehat{\varepsilon}_{S_i}(h)\right| \leq \sqrt{\frac{1}{2m}\left(\log\frac{2k}{\delta} + d\log\frac{me}{d}\right)}$$

Combining Thm. 3.1 and Corollaries. 3.1, 3.2 and realizing that $VC\dim(\mathcal{H}\Delta\mathcal{H}) \leq 2VC\dim(\mathcal{H})$ (Anthony & Bartlett, 2009), we have the following theorem:

Theorem 3.4. Let $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ be the target distribution and k source distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class where $VC\dim(\mathcal{H}) = d$. If $\widehat{\mathcal{D}}_T$ and $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ are the empirical distributions of $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ generated with m $i.i.d.$ samples from each domain, then, for $0 < \delta < 1$, with probability at least $1 - \delta$ (over the choice of samples), we have:

$$\begin{aligned}\varepsilon_T(h) &\leq \max_{i\in[k]} \widehat{\varepsilon}_{S_i}(h) + \sqrt{\frac{1}{2m}\left(\log\frac{4k}{\delta} + d\log\frac{me}{d}\right)} + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k) + \sqrt{\frac{2}{m}\left(\log\frac{8k}{\delta} + 2d\log\frac{me}{2d}\right)} + \lambda \\ &= \max_{i\in[k]} \widehat{\varepsilon}_{S_i}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k) + \lambda + O\left(\sqrt{\frac{1}{m}\left(\log\frac{k}{\delta} + d\log\frac{me}{d}\right)}\right) \qquad (2)\end{aligned}$$

Remark. Thm. 3.4 has a nice interpretation for each term: the first term measures the worst case accuracy of hypothesis h on the k source domains, and the second term measures the discrepancy between the target domain and the k source domains. For domain adaptation to succeed in the multiple sources setting, we have to expect these two terms to be small: we pick our hypothesis h based on its source training errors, and it will generalize only if the discrepancy between sources and target is small. The third term λ is the optimal error we can hope to achieve. Hence, if λ is large, one should not hope the generalization error to be small by training on the source domains. [1] The last term bounds the additional error we may incur because of the possible bias from finite samples. It is also worth pointing out that these four terms appearing in the generalization bound also capture the tradeoff between using a rich hypothesis class $\mathcal{H}$ and a limited one as we discussed above: when using a richer hypothesis class, the first and the third terms in the bound will decrease, while the value of the second term will increase; on the other hand, choosing a limited hypothesis class can decrease the value of the second term, but we may incur additional source training errors and a large λ due to the simplicity of $\mathcal{H}$. One interesting prediction implied by Thm. 3.4 is that the performance on the target domain depends on the worst empirical error among multiple source domains, i.e., it is not always beneficial to naively incorporate more source domains into training. As we will see in the experiment, this is indeed the case in many real-world problems.

Comparison with Existing Bounds First, it is easy to see that, upto a multiplicative constant, our bound in (2) reduces to the one in Thm. 2.1 when there is only one source domain ($k = 1$). Hence Thm. 3.4 can be treated as a generalization of Thm. 2.1. Blitzer et al. (2008) give a generalization bound for semi-supervised multisource domain adaptation where, besides labeled instances from multiple source domains, the algorithm also has access to a fraction of labeled instances from the target domain. Although in general our bound and the one in (Blitzer et al., 2008, Thm. 3) are incomparable, it is instructive to see the connections and differences between them: on one hand, the multiplicative constants of the discrepancy measure and the optimal error in our bound are half of those in Blitzer et al. (2008)'s bound, leading to a tighter bound; on the other hand, because of the access to labeled instances from the target domain, their bound is expressed relative to the optimal error rate on the target domain, while ours is in terms of the empirical error on the source domain. Finally, thanks to our generalized definition of $d_{\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$, we do not need to manually specify the optimal combination vector α in (Blitzer et al., 2008, Thm. 3), which is unknown in practice. Mansour et al. (2009b) also give a generalization bound for multisource domain adaptation under the assumption that the target distribution is a mixture of the k sources and the target hypothesis can be represented as a convex combination of the source hypotheses. While the distance measure we use assumes 0-1 loss function, their generalized discrepancy measure can also be applied for other losses functions (Mansour et al., 2009a;c;b).

4 Multisource Domain Adaptation with Adversarial Neural Networks

In this section we shall describe a neural network based implementation to minimize the generalization bound we derive in Thm. 3.4. The key idea is to reformulate the generalization bound by a minimax saddle point problem and optimize it via adversarial training.

[1] Of course it is still possible that $\varepsilon_T(h)$ is small while λ is large, but in domain adaptation we do not have access to labeled samples from $\mathcal{D}_T$.

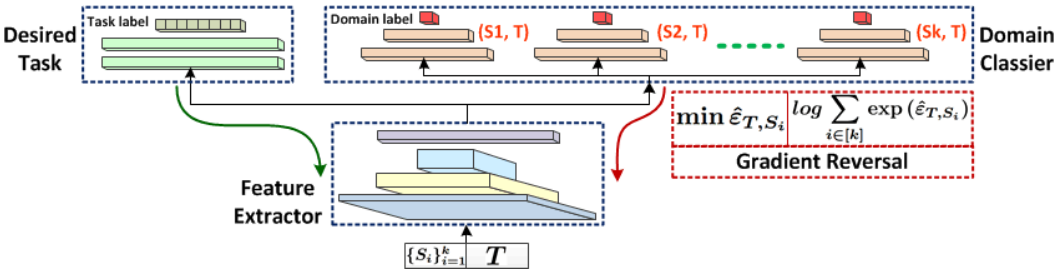

Figure 1: MDANs Network architecture. Feature extractor, domain classifier, and task learning are combined in one training process. Hard version: the source that achieves the minimum domain classification error is backpropagated with gradient reversal; Smooth version: all the domain classification risks over k source domains are combined and backpropagated adaptively with gradient reversal.

Suppose we are given samples drawn from k source domains $\{\mathcal{D}_{S_i}\}$, each of which contains m instance-label pairs. Additionally, we also have access to unlabeled instances sampled from the target domain $\mathcal{D}_T$. Once we fix our hypothesis class $\mathcal{H}$, the last two terms in the generalization bound (2) will be fixed; hence we can only hope to minimize the bound by minimizing the first two terms, i.e., the maximum source training error and the discrepancy between source domains and target domain. The idea is to train a neural network to learn a representation with the following two properties: 1). indistinguishable between the k source domains and the target domain; 2). informative enough for our desired task to succeed. Note that both requirements are necessary: without the second property, a neural network can learn trivial random noise representations for all the domains, and such representations cannot be distinguished by any discriminator; without the first property, the learned representation does not necessarily generalize to the unseen target domain. Taking these two properties into consideration, we propose the following optimization problem:

$$\text{minimize} \quad \max_{i\in[k]} \left(\widehat{\varepsilon}_{S_i}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k) \right) \tag{3}$$

One key observation that leads to a practical approximation of $d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k)$ from Ben-David et al. (2007) is that computing the discrepancy measure is closely related to learning a classifier that is able to disintuish samples from different domains:

$$d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k) = \max_{i\in[k]} \left(1 - 2 \min_{h\in\mathcal{H}\Delta\mathcal{H}} \left(\frac{1}{2m} \sum_{\mathbf{x}\sim\widehat{\mathcal{D}}_T} \mathbb{I}(h(\mathbf{x}) = 1) + \frac{1}{2m} \sum_{\mathbf{x}\sim\widehat{\mathcal{D}}_{S_i}} \mathbb{I}(h(\mathbf{x} = 0)) \right) \right)$$

Let $\widehat{\varepsilon}_{T,S_i}(h)$ be the empirical risk of hypothesis h in the domain discriminating task. Ignoring the constant terms that do not affect the optimization formulation, moving the $\max$ operator out, we can reformulate (3) as:

$$\text{minimize} \quad \max_{i\in[k]} \left(\widehat{\varepsilon}_{S_i}(h) - \min_{h'\in\mathcal{H}\Delta\mathcal{H}} \widehat{\varepsilon}_{T,S_i}(h') \right) \tag{4}$$

The two terms in (4) exactly correspond to the two criteria we just proposed: the first term asks for an informative feature representation for our desired task to succeed, while the second term captures the notion of invariant feature representations between different domains.

Inspired by Ganin et al. (2016), we use the gradient reversal layer to effectively implement (4) by backpropagation. The network architecture is shown in Figure. 1. The pseudo-code is listed in Alg. 1 (the hard version). One notable drawback of the hard version in Alg. 1 is that in each iteration the algorithm only updates its parameter based on the gradient from one of the k domains. This is data inefficient and can waste our computational resources in the forward process. To improve this, we approximate the $\max$ function in (4) by the log-sum-exp function, which is a frequently used smooth approximation of the $\max$ function. Define $\widehat{\varepsilon}_i(h) := \widehat{\varepsilon}_{S_i}(h) - \min_{h'\in\mathcal{H}\Delta\mathcal{H}} \widehat{\varepsilon}_{T,S_i}(h')$:

$$\max_{i\in[k]} \widehat{\varepsilon}_i(h) \approx \frac{1}{\gamma} \log \sum_{i\in[k]} \exp(\gamma \widehat{\varepsilon}_i(h))$$

Algorithm 1 Multiple Source Domain Adaptation via Adversarial Training

for $t = 1$ to ∞ **do**
  Sample $\{S_i^{(t)}\}_{i=1}^k$ and $T^{(t)}$ from $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ and $\widehat{\mathcal{D}}_T$, each of size m
  for $i = 1$ to k **do**
    Compute $\widehat{\varepsilon}_i^{(t)} := \widehat{\varepsilon}_{S_i^{(t)}}(h) - \min_{h' \in \mathcal{H}\Delta\mathcal{H}} \widehat{\varepsilon}_{T^{(t)}, S_i^{(t)}}(h')$
    Compute $w_i^{(t)} := \exp(\widehat{\varepsilon}_i^{(t)})$
  end for
  # Hard version
  Select $i^{(t)} := \arg\max_{i \in [k]} \widehat{\varepsilon}_i^{(t)}$
  Update parameters via backpropagating gradient of $\widehat{\varepsilon}_{i^{(t)}}^{(t)}$
  # Smoothed version
  for $i = 1$ to k **do**
    Normalize $w_i^{(t)} \leftarrow w_i^{(t)} / \sum_{i' \in [k]} w_{i'}^{(t)}$
  end for
  Update parameters via backpropagating gradient of $\sum_{i \in [k]} w_i^{(t)} \widehat{\varepsilon}_i^{(t)}$
end for

where $\gamma > 0$ is a parameter that controls the accuracy of this approximation. As $\gamma \to \infty$, $\frac{1}{\gamma} \log \sum_{i \in [k]} \exp(\gamma \widehat{\varepsilon}_i(h)) \to \max_{i \in [k]} \widehat{\varepsilon}_i(h)$. Correspondingly, we can formulate a smoothed version of (4) as:

$$\text{minimize} \quad \frac{1}{\gamma} \log \sum_{i \in [k]} \exp\left(\gamma(\widehat{\varepsilon}_{S_i}(h) - \min_{h' \in \mathcal{H}\Delta\mathcal{H}} \widehat{\varepsilon}_{T,S_i}(h'))\right) \tag{5}$$

During the optimization, (5) naturally provides an adaptive weighting scheme for the k source domains depending on their relative error. Use θ to denote all the model parameters, then:

$$\frac{\partial}{\partial \theta} \frac{1}{\gamma} \log \sum_{i \in [k]} \exp\left(\gamma(\widehat{\varepsilon}_{S_i}(h) - \min_{h' \in \mathcal{H}\Delta\mathcal{H}} \widehat{\varepsilon}_{T,S_i}(h'))\right) = \sum_{i \in [k]} \frac{\exp \gamma \widehat{\varepsilon}_i(h)}{\sum_{i' \in [k]} \exp \gamma \widehat{\varepsilon}_{i'}(h)} \frac{\partial \widehat{\varepsilon}_i(h)}{\partial \theta} \tag{6}$$

The approximation trick not only smooths the objective, but also provides a principled and adaptive way to combine all the gradients from the k source domains. In words, (6) says that the gradient of MDAN is a convex combination of the gradients from all the domains. The larger the error from one domain, the larger the combination weight in the ensemble. We summarize this algorithm in the smoothed version of Alg. 1. Note that both algorithms, including the hard version and the smoothed version, reduce to the DANN algorithm (Ganin et al., 2016) when there is only one source domain.

5 Experiments

We evaluate both hard and soft MDANs and compare them with state-of-the-art methods on three real-world datasets: the Amazon benchmark dataset (Chen et al., 2012) for sentiment analysis, a digit classification task that includes 4 datasets: MNIST (LeCun et al., 1998), MNIST-M (Ganin et al., 2016), SVHN (Netzer et al., 2011), and SynthDigits (Ganin et al., 2016), and a public, large-scale image dataset on vehicle counting from city cameras (Zhang et al., 2017). Details about network architecture and training parameters of proposed and baseline methods, and detailed dataset description will be introduced in the appendix.

5.1 Amazon Reviews

Domains within the dataset consist of reviews on a specific kind of product (Books, DVDs, Electronics, and Kitchen appliances). Reviews are encoded as 5000 dimensional feature vectors of unigrams and bigrams, with binary labels indicating sentiment. We conduct 4 experiments: for each of them, we pick one product as target domain and the rest as source domains. Each source domain has 2000 labeled examples, and the target test set has 3000 to 6000 examples. During training, we randomly sample the same number of unlabeled target examples as the source examples in each mini-batch. We

implement the Hard-Max and Soft-Max methods according to Alg. 1, and compare them with three baselines: MLPNet, marginalized stacked denoising autoencoders (mSDA) (Chen et al., 2012), and DANN (Ganin et al., 2016). DANN cannot be directly applied in multiple source domains setting. In order to make a comparison, we use two protocols. The first one is to combine all the source domains into a single one and train it using DANN, which we denote as (cDANN). The second protocol is to train multiple DANNs separately, where each one corresponds to a source-target pair. Among all the DANNs, we report the one achieving the best performance on the target domain. We denote this experiment as (sDANN). For fair comparison, all these models are built on the same basic network structure with one input layer (5000 units) and three hidden layers (1000, 500, 100 units).

Results and Analysis We show the accuracy of different methods in Table 1. Clearly, Soft-Max significantly outperforms all other methods in most settings. When Kitchen is the target domain, cDANN performs slightly better than Soft-Max, and all the methods perform close to each other. Hard-Max is typically slightly worse than Soft-Max. This is mainly due to the low data-efficiency of the Hard-Max model (Section 4, Eq. 4, Eq. 5). We argue that with more training iterations, the performance of Hard-Max can be further improved. These results verify the effectiveness of MDANs for multisource domain adaptation. To validate the statistical significance of the results, we run a non-parametric Wilcoxon signed-ranked test for each task to compare Soft-Max with the other competitors, as shown in Table 2. Each cell corresponds to the p-value of a Wilcoxon test between Soft-Max and one of the other methods, under the null hypothesis that the two paired samples have the same mean. From these p-values, we see Soft-Max is convincingly better than other methods.

Table 1: Sentiment classification accuracy.

Train/Test	**MLPNet**	**mSDA**	**sDANN**	**cDANN**	**MDANs**	
					H-Max	S-Max
D+E+K/B	0.7655	0.7698	0.7650	0.7789	0.7845	**0.7863**
B+E+K/D	0.7588	0.7861	0.7732	0.7886	0.7797	**0.8065**
B+D+K/E	0.8460	0.8198	0.8381	0.8491	0.8483	**0.8534**
B+D+E/K	0.8545	0.8426	0.8433	**0.8639**	0.8580	0.8626

Table 2: p-values under Wilcoxon test.

	MLPNet	**mSDA**	**sDANN**	**cDANN**	**H-Max**
	S-Max	**S-Max**	**S-Max**	**S-Max**	**S-Max**
B	0.550	0.101	0.521	0.013	0.946
D	0.000	0.072	0.000	0.051	0.000
E	0.066	0.000	0.097	0.150	0.022
K	0.306	0.001	0.001	0.239	0.008

5.2 Digits Datasets

Following the setting in (Ganin et al., 2016), we combine four popular digits datasets (MNIST, MNIST-M, SVHN, and SynthDigits) to build the multisource domain dataset. We take each of MNIST-M, SVHN, and MNIST as target domain in turn, and the rest as sources. Each source domain has $20,000$ labeled images and the target test set has $9,000$ examples. We compare Hard-Max and Soft-Max of MDANs with five baselines: i). *best-Single-Source*. A basic network trained on each source domain ($20,000$ images) without domain adaptation and tested on the target domain. Among the three models, we report the one achieves the best performance on the test set. ii). *Combine-Source*. A basic network trained on a combination of three source domains ($20,000$ images for each) without domain adaptation and tested on the target domain. iii). *best-Single-DANN*. We train DANNs (Ganin et al., 2016) on each source-target domain pair ($20,000$ images) and test it on target. Again, we report the best score among the three. iv). *Combine-DANN*. We train a single DANN on a combination of three source domains ($20,000$ images for each). v). *Target-only*. It is the basic network trained and tested on the target data. It serves as an upper bound of DA algorithms. All the MDANs and baseline methods are built on the same basic network structure to put them on a equal footing.

Results and Analysis The classification accuracy is shown in Table 3. The results show that a naive combination of different training datasets can sometimes even decrease the performance. Furthermore, we observe that adaptation to the SVHN dataset (the third experiment) is hard. In this case, increasing the number of source domains does not help. We conjecture this is due to the large dissimilarity between the SVHN data to the others. For the combined sources, MDANs always perform better than the source-only baseline (MDANs vs. Combine-Source). However, directly training DANN on a combination of multiple sources leads to worse performance when compared with our approach (Combine-DANN vs. MDANs). In fact, this strategy may even lead to worse results than the source-only baseline (Combine-DANN vs. Combine-Source). Surprisingly, using a single domain (best-Single DANN) can sometimes achieve the best result. This means that in domain adaptation the quality of data (how close to the target data) is much more important than the quantity (how many source domains). As a conclusion, this experiment further demonstrates the effectiveness of MDANs when there are multiple source domains available, where a naive combination of multiple sources using DANN may hurt generalization.

Table 3: Accuracy on digit classification. Mt: MNIST; Mm: MNIST-M, Sv: SVHN, Sy: SynthDigits.

Train/Test	best-Single Source	best-Single DANN	Combine Source	Combine DANN	MDAN		Target Only
					Hard-Max	Soft-Max	
Sv+Mm+Sy/Mt	0.964	0.967	0.938	0.925	0.976	**0.979**	0.987
Mt+Sv+Sy/Mm	0.519	0.591	0.561	0.651	0.663	**0.687**	0.901
Mm+Mt+Sy/Sv	0.814	**0.818**	0.771	0.776	0.802	0.816	0.898

Table 4: Counting error statistics. S is the number of source cameras; T is the target camera id.

S	T	MDANs		DANN	FCN	T	MDANs		DANN	FCN
		Hard-Max	Soft-Max				Hard-Max	Soft-Max		
2	A	1.8101	**1.7140**	1.9490	1.9094	B	2.5059	**2.3438**	2.5218	2.6528
3	A	1.3276	**1.2363**	1.3683	1.5545	B	1.9092	**1.8680**	2.0122	2.4319
4	A	1.3868	**1.1965**	1.5520	1.5499	B	**1.7375**	1.8487	2.1856	2.2351
5	A	1.4021	**1.1942**	1.4156	1.7925	B	1.7758	**1.6016**	1.7228	2.0504
6	A	1.4359	**1.2877**	2.0298	1.7505	B	1.5912	**1.4644**	1.5484	2.2832
7	A	1.4381	**1.2984**	1.5426	1.7646	B	1.5989	**1.5126**	1.5397	1.7324

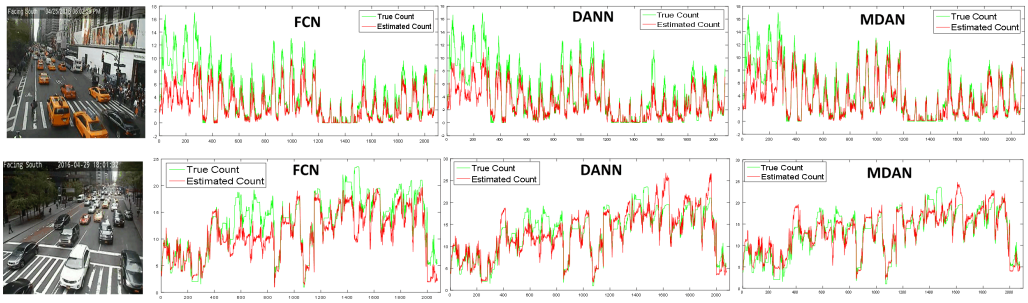

Figure 2: Counting results for target camera A (first row) and B (second row). X-frames; Y-Counts.

5.3 WEBCAMT VEHICLE COUNTING DATASET

WebCamT is a public dataset for vehicle counting from large-scale city camera videos, which has low resolution (352×240), low frame rate (1 frame/second), and high occlusion. It has $60,000$ frames annotated with vehicle bounding box and count, divided into training and testing sets, with $42,200$ and $17,800$ frames, respectively. Here we demonstrate the effectiveness of MDANs to count vehicles from an unlabeled target camera by adapting from multiple labeled source cameras: we select 8 cameras that each has more than $2,000$ labeled images for our evaluations. As shown in Fig. 3, they are located in different intersections of the city with different scenes. Among these 8 cameras, we randomly pick two cameras and take each camera as the target camera, with the other 7 cameras as sources. We compute the proxy $\mathcal{A}$-distance (PAD) (Ben-David et al., 2007) between each

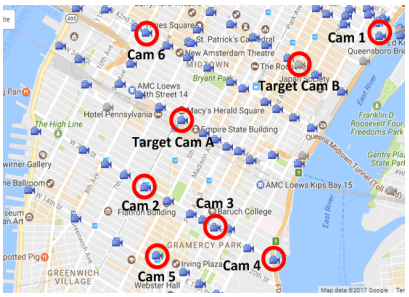

Figure 3: Source&target camera map.

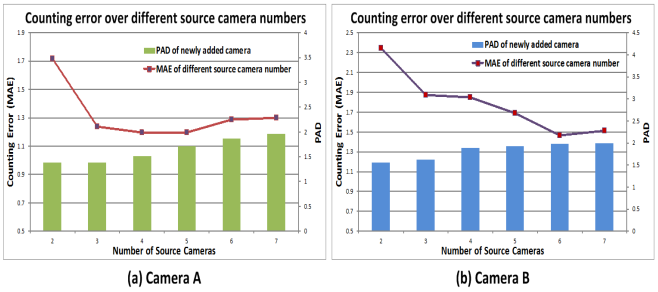

Figure 4: Counting error over different source numbers.

source camera and the target camera to approximate the divergence between them. We then rank the source cameras by the PAD from low to high and choose the first k cameras to form the k source domains. Thus the proposed methods and baselines can be evaluated on different numbers of sources (from 2 to 7). We implement the Hard-Max and Soft-Max MDANs according to Alg. 1, based on the basic vehicle counting network FCN (Zhang et al., 2017). We compare our method with two baselines: FCN (Zhang et al., 2017), a basic network without domain adaptation, and DANN (Ganin et al., 2016), implemented on top of the same basic network. We record mean absolute error (MAE) between true count and estimated count.

Results and Analysis The counting error of different methods is compared in Table 4. The Hard-Max version achieves lower error than DANN and FCN in most settings for both target cameras. The Soft-Max approximation outperforms all the baselines and the Hard-Max in most settings, demonstrating the effectiveness of the smooth and adaptative approximation. The lowest MAE achieved by Soft-Max is 1.1942. Such MAE means that there is only around one vehicle miscount for each frame (the average number of vehicles in one frame is around 20). Fig. 2 shows the counting results of Soft-Max for the two target cameras under the 5 source cameras setting. We can see that the proposed method accurately counts the vehicles of each target camera for long time sequences. Does adding more source cameras always help improve the performance on the target camera? To answer this question, we analyze the counting error when we vary the number of source cameras as shown in Fig. 4. From the curves, we see the counting error goes down with more source cameras at the beginning, while it goes up when more sources are added at the end. This phenomenon corresponds to the prediction implied by Thm. 3.4 (the last remark in Section 3): the performance on the target domain depends on the worst empirical error among multiple source domains, i.e., it is not always beneficial to naively incorporate more source domains into training. To illustrate this prediction better, we show the PAD of the newly added camera (when the source number increases by one) in Fig. 4. By observing the PAD and the counting error, we see the performance on the target can degrade when the newly added source camera has large divergence from the target camera.

6 Conclusion

We derive a new generalization bound for DA under the setting of multiple source domains with labeled instances and one target domain with unlabeled instances. The new bound has interesting interpretation and reduces to an existing bound when there is only one source domain. Following our theoretical results, we propose MDANs to learn feature representations that are invariant under multiple domain shifts while at the same time being discriminative for the learning task. Both hard and soft versions of MDANs are generalizations of the popular DANN to the case when multiple source domains are available. Empirically, MDANs outperform the state-of-the-art DA methods on three real-world datasets, including a sentiment analysis task, a digit classification task, and a visual vehicle counting task, demonstrating its effectiveness for multisource domain adaptation.

References

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 769–776, 2013.

Carlos J Becker, Christos M Christoudias, and Pascal Fua. Non-linear domain adaptation with boosting. In *Advances in Neural Information Processing Systems*, pp. 485–493, 2013.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 120–128. Association for Computational Linguistics, 2006.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pp. 129–136, 2008.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pp. 343–351, 2016.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In *International Conference on Algorithmic Learning Theory*, pp. 38–53. Springer, 2008.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Icml*, volume 32, pp. 647–655, 2014.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117. ACM, 2004.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In *ICML (3)*, pp. 738–746, 2013.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 513–520, 2011.

Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML (1)*, pp. 222–230, 2013.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation. In *Computer Vision–ECCV 2012*, pp. 702–715. Springer, 2012.

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pp. 601–608, 2006.

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 180–191. VLDB Endowment, 2004.

Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pp. 97–105, 2015.

Yishay Mansour and Mariano Schain. Robust domain adaptation. In *ISAIM*, 2012.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009a.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pp. 1041–1048, 2009b.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 367–374. AUAI Press, 2009c.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2016.

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17: 138–155, 2009.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Shanghang Zhang, Guanhang Wu, Joao P Costeira, and Jose MF Moura. Understanding traffic density from large-scale web camera data. *arXiv preprint arXiv:1703.05868*, 2017.

A OUTLINE

Organization of the appendix: 1). For the convenience of exposition in showing our technical proofs, we first introduce the technical tools that will be used during our proofs in Sec. B. 2). We provide detailed proofs for all the claims, lemmas and theorems presented in the main paper in Sec. C. 3). We describe more experiment details in Sec. D, including dataset description, network architecture and training parameters of the proposed and baseline methods, and more analysis of the experimental results. 4). We introduce and discuss more related work about domain adaptation in Sec. E.

B TECHNICAL TOOLS

Definition B.1 (Growth function)**.** The *growth function* $\Pi_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ for a hypothesis class $\mathcal{H}$ is defined by:

$$\forall m \in \mathbb{N}, \quad \Pi_{\mathcal{H}}(m) = \max_{X_m \subseteq \mathcal{X}} |\{(h(x_1), \ldots, h(x_m)) \mid h \in \mathcal{H}\}|$$

where $X_m = \{x_1, \ldots, x_m\}$ is a subset of $\mathcal{X}$ with size m.

Roughly, the growth function $\Pi_{\mathcal{H}}(m)$ computes the maximum number of distinct ways in which m points can be classified using hypothesis in $\mathcal{H}$. A closely related concept is the *Vapnik–Chervonenkis dimension* (VC dimension) (Vapnik, 1998):

Definition B.2 (VC dimension)**.** The VC-dimension of a hypothesis class $\mathcal{H}$ is defined as:

$$VC\text{dim}(\mathcal{H}) = \max\{m : \Pi_{\mathcal{H}}(m) = 2^m\}$$

A well-known result relating $VC\text{dim}(\mathcal{H})$ and the growth function $\Pi_{\mathcal{H}}(m)$ is the Sauer's lemma:

Lemma B.1 (Sauer's lemma)**.** Let $\mathcal{H}$ be a hypothesis class with $VC\text{dim}(\mathcal{H}) = d$. Then, for $m \geq d$, the following inequality holds:

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d$$

The following concentration inequality will be used:

Theorem B.1 (Hoeffding's inequality)**.** Let $X_1, \ldots, X_n$ be independent random variables where each X_i is bounded by the interval $[a_i, b_i]$. Define the empirical mean of these random variables by $\bar{X} := \frac{1}{n}\sum_{i=1}^{n} X_i$, then $\forall \varepsilon > 0$:

$$\Pr\left(\left|\bar{X} - \mathbb{E}[\bar{X}]\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2n^2\varepsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

The VC inequality allows us to give a uniform bound on the binary classification error of a hypothesis class $\mathcal{H}$ using growth function:

Theorem B.2 (VC inequality)**.** Let $\Pi_{\mathcal{H}}$ be the growth function of hypothesis class $\mathcal{H}$. For $h \in \mathcal{H}$, let $\varepsilon(h)$ be the true risk of h w.r.t. the generation distribution $\mathcal{D}$ and the true labeling function h^*. Similarly, let $\hat{\varepsilon}_n(h)$ be the empirical risk on a random $i.i.d.$ sample containing n instances from $\mathcal{D}$, then, for $\forall \varepsilon > 0$, the following inequality hold:

$$\Pr\left(\sup_{h \in \mathcal{H}} |\varepsilon(h) - \hat{\varepsilon}_n(h)| \geq \varepsilon\right) \leq 8\Pi_{\mathcal{H}}(n)\exp\left(-n\varepsilon^2/32\right)$$

Although the above theorem is stated for binary classification error, we can extend it to any bounded error. This will only change the multiplicative constant of the bound.

C PROOFS

For all the proofs presented here, the following lemma shown by Blitzer et al. (2008) will be repeatedly used:

Lemma C.1 ((Blitzer et al., 2008))**.** $\forall h, h' \in \mathcal{H}, \quad |\varepsilon_S(h, h') - \varepsilon_T(h, h')| \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$.

C.1 PROOF OF THM. 3.1

One technical lemma we will frequently use to prove Thm. 3.1 is the triangular inequality w.r.t. $\varepsilon_{\mathcal{D}}(h), \forall h \in \mathcal{H}$:

Lemma C.2. For any hypothesis class $\mathcal{H}$ and any distribution $\mathcal{D}$ on $\mathcal{X}$, the following triangular inequality holds:

$$\forall h, h', f \in \mathcal{H}, \quad \varepsilon_{\mathcal{D}}(h, h') \leq \varepsilon_{\mathcal{D}}(h, f) + \varepsilon_{\mathcal{D}}(f, h')$$

Proof.

$$\varepsilon_{\mathcal{D}}(h, h') = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[|h(\mathbf{x}) - h'(\mathbf{x})|] \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[|h(\mathbf{x}) - f(\mathbf{x})| + |f(\mathbf{x}) - f(\mathbf{x})|] = \varepsilon_{\mathcal{D}}(h, f) + \varepsilon_{\mathcal{D}}(f, h')$$

■

Now we are ready to prove Thm. 3.1:

Theorem 3.1. $\varepsilon_T(h) \leq \max_{i\in[k]} \varepsilon_{S_i}(h) + \lambda + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)$.

Proof. $\forall h \in \mathcal{H}$, define $i_h := \arg\max_{i\in[k]} \varepsilon_{S_i}(h, h^*)$:

$$\begin{aligned}
\varepsilon_T(h) &\leq \varepsilon_T(h^*) + \varepsilon_T(h, h^*) \\
&= \varepsilon_T(h^*) + \varepsilon_T(h, h^*) - \max_{i\in[k]} \varepsilon_{S_i}(h, h^*) + \max_{i\in[k]} \varepsilon_{S_i}(h, h^*) \\
&\leq \varepsilon_T(h^*) + |\varepsilon_T(h, h^*) - \varepsilon_{S_{i_h}}(h, h^*)| + \varepsilon_{S_{i_h}}(h, h^*) \\
&\leq \varepsilon_T(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_{S_{i_h}}) + \varepsilon_{S_{i_h}}(h, h^*) \\
&\leq \varepsilon_T(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) + \varepsilon_{S_{i_h}}(h, h^*) \\
&\leq \varepsilon_T(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) + \varepsilon_{S_{i_h}}(h) + \varepsilon_{S_{i_h}}(h^*) \\
&\leq \varepsilon_T(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) + \max_{i\in[k]} \varepsilon_{S_i}(h) + \max_{i\in[k]} \varepsilon_{S_i}(h^*) \\
&= \max_{i\in[k]} \varepsilon_{S_i}(h) + \lambda + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k)
\end{aligned}$$

The first and the fifth inequalities are due to the triangle inequality, and the third inequality is based on Lemma C.1. The second holds due to the property of $|\cdot|$ and the others follow by the definition of $\mathcal{H}$-divergence. ■

C.2 PROOF OF THM. 3.2

Theorem 3.2. Let $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ be the target distribution and k source distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis class where $VC\dim(\mathcal{H}) = d$. If $\widehat{\mathcal{D}}_T$ and $\{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k$ are the empirical distributions of $\mathcal{D}_T$ and $\{\mathcal{D}_{S_i}\}_{i=1}^k$ generated with m $i.i.d.$ samples from each domain, then for $\epsilon > 0$, we have:

$$\Pr\left(\left|d_{\mathcal{H}}(\mathcal{D}_T; \{\mathcal{D}_{S_i}\}_{i=1}^k) - d_{\mathcal{H}}(\widehat{\mathcal{D}}_T; \{\widehat{\mathcal{D}}_{S_i}\}_{i=1}^k)\right| \geq \epsilon\right) \leq 4k \left(\frac{em}{d}\right)^d \exp\left(-m\epsilon^2/8\right)$$

Proof.

$$
\begin{aligned}
&\Pr\left(\left|d_{\mathcal{H}}(\mathcal{D}_T;\{\mathcal{D}_{S_i}\}_{i=1}^k) - d_{\mathcal{H}}(\hat{\mathcal{D}}_T;\{\hat{\mathcal{D}}_{S_i}\}_{i=1}^k)\right| \geq \epsilon\right) \\
&= \Pr\left(\left|\max_{i\in[k]}\sup_{A\in\mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}_T}(A) - \Pr_{\mathcal{D}_{S_i}}(A)| - \max_{i\in[k]}\sup_{A\in\mathcal{A}_{\mathcal{H}}} |\Pr_{\hat{\mathcal{D}}_T}(A) - \Pr_{\hat{\mathcal{D}}_{S_i}}(A)|\right| \geq \frac{\epsilon}{2}\right) \\
&\leq \Pr\left(\max_{i\in[k]}\sup_{A\in\mathcal{A}_{\mathcal{H}}}\left||\Pr_{\mathcal{D}_T}(A) - \Pr_{\mathcal{D}_{S_i}}(A)| - |\Pr_{\hat{\mathcal{D}}_T}(A) - \Pr_{\hat{\mathcal{D}}_{S_i}}(A)|\right| \geq \frac{\epsilon}{2}\right) \\
&= \Pr\left(\exists i\in[k], \exists A\in\mathcal{A}_{\mathcal{H}} : \left||\Pr_{\mathcal{D}_T}(A) - \Pr_{\mathcal{D}_{S_i}}(A)| - |\Pr_{\hat{\mathcal{D}}_T}(A) - \Pr_{\hat{\mathcal{D}}_{S_i}}(A)|\right| \geq \frac{\epsilon}{2}\right) \\
&\leq \sum_{i=1}^k \Pr\left(\exists A\in\mathcal{A}_{\mathcal{H}} : \left||\Pr_{\mathcal{D}_T}(A) - \Pr_{\mathcal{D}_{S_i}}(A)| - |\Pr_{\hat{\mathcal{D}}_T}(A) - \Pr_{\hat{\mathcal{D}}_{S_i}}(A)|\right| \geq \frac{\epsilon}{2}\right) \\
&\leq \sum_{i=1}^k \Pr\left(\exists A\in\mathcal{A}_{\mathcal{H}} : |\Pr_{\mathcal{D}_T}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)| + |\Pr_{\mathcal{D}_{S_i}}(A) - \Pr_{\hat{\mathcal{D}}_{S_i}}(A)| \geq \frac{\epsilon}{2}\right) \\
&\leq 2k\Pr\left(\exists A\in\mathcal{A}_{\mathcal{H}} : |\Pr_{\mathcal{D}_T}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)| \geq \frac{\epsilon}{4}\right) \\
&\leq 2k\cdot\Pi_{\mathcal{A}_{\mathcal{H}}}(m)\Pr\left(|\Pr_{\mathcal{D}_T}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)| \geq \frac{\epsilon}{4}\right) \\
&\leq 2k\cdot\Pi_{\mathcal{A}_{\mathcal{H}}}(m)\cdot 2\exp(-2m\epsilon^2/16) \\
&\leq 4k\left(\frac{em}{d}\right)^d\exp(-m\epsilon^2/8)
\end{aligned}
$$

The first inequality holds due to the sub-additivity of the $\max$ function, and the second inequality is due to the union bound. The third inequality holds because of the triangle inequality, and we use the averaging argument to establish the fourth inequality. The fifth inequality is an application of the VC-inequality, and the sixth is by the Hoeffding's inequality. Finally, we use the Sauer's lemma to prove the last inequality. ■

C.3 Proof of Thm. 3.3

We now show the detailed proof of Thm. 3.3.

Proof.

$$
\begin{aligned}
\Pr\left(\sup_{h\in\mathcal{H}}\left|\max_{i\in[k]}\varepsilon_{S_i}(h) - \max_{i\in[k]}\hat{\varepsilon}_{S_i}(h)\right| \geq \epsilon\right) &\leq \Pr\left(\sup_{h\in\mathcal{H}}\max_{i\in[k]}|\varepsilon_{S_i}(h) - \hat{\varepsilon}_{S_i}(h)| \geq \epsilon\right) \\
&= \Pr\left(\max_{i\in[k]}\sup_{h\in\mathcal{H}}|\varepsilon_{S_i}(h) - \hat{\varepsilon}_{S_i}(h)| \geq \epsilon\right) \\
&\leq \sum_{i=1}^k \Pr\left(\sup_{h\in\mathcal{H}}|\varepsilon_{S_i}(h) - \hat{\varepsilon}_{S_i}(h)| \geq \epsilon\right) \\
&\leq k\cdot\Pi_{\mathcal{H}}(m)\Pr\left(|\varepsilon_{S_i}(h) - \hat{\varepsilon}_{S_i}(h)| \geq \epsilon\right) \\
&\leq k\cdot\Pi_{\mathcal{H}}(m)\cdot 2\exp(-2m\epsilon^2) \\
&\leq 2k\left(\frac{me}{d}\right)^d\exp(-2m\epsilon^2)
\end{aligned}
$$

Again, the first inequality is due to the subadditivity of the $\max$ function, and the second inequality holds due to the union bound. We apply the VC-inequality to bound the third inequality, and Hoeffding's inequality to bound the fourth. Again, the last one is due to Sauer's lemma. ■

C.4 DERIVATION OF THE DISCREPANCY DISTANCE AS CLASSIFICATION ERROR

We show that the $\mathcal{H}$-divergence is equivalent to a binary classification accuracy in discriminating instances from different domains. Suppose $\mathcal{A}_{\mathcal{H}}$ is symmetric, i.e., $A \in \mathcal{A}_{\mathcal{H}} \Leftrightarrow \mathcal{X} \backslash A \in \mathcal{A}_{\mathcal{H}}$, and we have samples $\{S_i\}_{i=1}^k$ and T from $\{\mathcal{D}_{S_i}\}_{i=1}^k$ and $\mathcal{D}_T$ respectively, each of which is of size m, then:

$$
\begin{aligned}
d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_T; \{\hat{\mathcal{D}}_{S_i}\}_{i=1}^k) &= \max_{i\in[k]} \sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} |\Pr_{\hat{\mathcal{D}}_T}(A) - \Pr_{\hat{\mathcal{D}}_{S_i}}(A)| \\
&= \max_{i\in[k]} \sup_{h\in\mathcal{H}\Delta\mathcal{H}} |\Pr_{\mathbf{x}\sim\hat{\mathcal{D}}_T}(h(\mathbf{x})=1) - \Pr_{\mathbf{x}\sim\hat{\mathcal{D}}_{S_i}}(h(\mathbf{x}=1))| \\
&= \max_{i\in[k]} \sup_{h\in\mathcal{H}\Delta\mathcal{H}} 1 - \left(\Pr_{\mathbf{x}\sim\hat{\mathcal{D}}_T}(h(\mathbf{x})=1) + \Pr_{\mathbf{x}\sim\hat{\mathcal{D}}_{S_i}}(h(\mathbf{x}=0))\right) \\
&= \max_{i\in[k]} \left(1 - 2 \min_{h\in\mathcal{H}\Delta\mathcal{H}} \left(\frac{1}{2m}\sum_{\mathbf{x}\sim\hat{\mathcal{D}}_T} \mathbb{I}(h(\mathbf{x})=1) + \frac{1}{2m}\sum_{\mathbf{x}\sim\hat{\mathcal{D}}_{S_i}} \mathbb{I}(h(\mathbf{x}=0))\right)\right)
\end{aligned}
$$

D DETAILS ABOUT EXPERIMENTS

In this section, we describe more details about the datasets and the experimental settings. We extensively evaluate the proposed methods on three datasets: 1). We first evaluate our methods on Amazon Reviews dataset (Chen et al., 2012) for sentiment analysis. 2). We evaluate the proposed methods on the digits classification datasets including MNIST (LeCun et al., 1998), MNIST-M (Ganin et al., 2016), SVHN (Netzer et al., 2011), and SynthDigits (Ganin et al., 2016). 3). We further evaluate the proposed methods on the public dataset WebCamT (Zhang et al., 2017) for vehicle counting. It contains 60,000 labeled images from 12 city cameras with different distributions. Due to the substantial difference between these datasets and their corresponding learning tasks, we will introduce more detailed dataset description, network architecture, and training parameters for each dataset respectively in the following subsections.

D.1 DETAILS ON AMAZON REVIEWS EVALUATION

Amazon reviews dataset includes four domains, each one composed of reviews on a specific kind of product (Books, DVDs, Electronics, and Kitchen appliances). Reviews are encoded as 5000 dimensional feature vectors of unigrams and bigrams. The labels are binary: 0 if the product is ranked up to 3 stars, and 1 if the product is ranked 4 or 5 stars.

We take one product domain as target and the other three as source domains. Each source domain has 2000 labeled examples and the target test set has 3000 to 6000 examples. We implement the Hard-Max and Soft-Max methods according to Alg. 1, based on a basic network with one input layer (5000 units) and three hidden layers (1000, 500, 100 units). The network is trained for 50 epochs with dropout rate 0.7. We compare Hard-Max and Soft-Max with three baselines: *Baseline 1: MLPNet*. It is the basic network of our methods (one input layer and three hidden layers), trained for 50 epochs with dropout rate 0.01. *Baseline 2: Marginalized Stacked Denoising Autoencoders (mSDA)* (Chen et al., 2012). It takes the unlabeled parts of both source and target samples to learn a feature map from input space to a new representation space. As a denoising autoencoder algorithm, it finds a feature representation from which one can (approximately) reconstruct the original features of an example from its noisy counterpart. *Baseline 3: DANN*. We implement DANN based on the algorithm described in (Ganin et al., 2016) with the same basic network as our methods. Hyper parameters of the proposed and baseline methods are selected by cross validation. Table 5 summarizes the network architecture and some hyper parameters.

D.2 DETAILS ON DIGIT DATASETS EVALUATION

We evaluate the proposed methods on the digits classification problem. Following the experiments in (Ganin et al., 2016), we combine four popular digits datasets-MNIST, MNIST-M, SVHN, and SynthDigits to build the multi-source domain dataset. MNIST is a handwritten digits database with $60,000$ training examples, and $10,000$ testing examples. The digits have been size-normalized and

Table 5: Network parameters for proposed and baseline methods

Method	Input layer	Hidden layers	Epochs	Dropout	Domains	Domain adaptation weight	Gamma
MLPNet	5000	(1000, 500, 100)	50	0.01	N/A	N/A	N/A
DANN	5000	(1000, 500, 100)	50	0.01	1	0.01	N/A
MDAN	5000	(1000, 500, 100)	50	0.7	3	0.1	10

centered in a 28×28 image. MNIST-M is generated by blending digits from the original MNIST set over patches randomly extracted from color photos from BSDS500 (Arbelaez et al., 2011; Ganin et al., 2016). It has $59,001$ training images and $9,001$ testing images with 32×32 resolution. An output sample is produced by taking a patch from a photo and inverting its pixels at positions corresponding to the pixels of a digit. For DA problems, this domain is quite distinct from MNIST, for the background and the strokes are no longer constant. SVHN is a real-world house number dataset with $73,257$ training images and $26,032$ testing images. It can be seen as similar to MNIST, but comes from a significantly harder, unsolved, real world problem. SynthDigits consists of $500;000$ digit images generated by Ganin et al. (2016) from WindowsTM fonts by varying the text, positioning, orientation, background and stroke colors, and the amount of blur. The degrees of variation were chosen to simulate SVHN, but the two datasets are still rather distinct, with the biggest difference being the structured clutter in the background of SVHN images.

We take MNIST-M, SVHN, and MNIST as target domain in turn, and the remaining three as sources. We implement the Hard-Max and Soft-Max versions according to Alg. 1 based on a basic network, as shown in Fig. 5. The baseline methods are also built on the same basic network structure to put them on a equal footing. The network structure and parameters of MDANs are illustrated in Fig. 5. The learning rate is initialized by 0.01 and adjusted by the first and second order momentum in the training process. The domain adaptation parameter of MDANs is selected by cross validation. In each mini-batch of MDANs training process, we randomly sample the same number of unlabeled target images as the number of the source images.

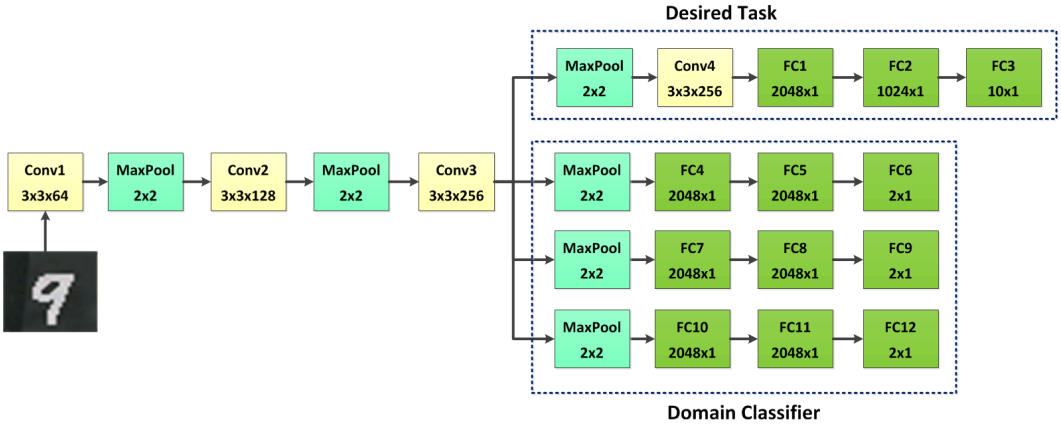

Figure 5: MDANs network architecture for digit classification

D.3 Details on WebCamT Vehicle Counting

WebCamT is a public dataset for large-scale city camera videos, which have low resolution (352×240), low frame rate (1 frame/second), and high occlusion. WebCamT has $60,000$ frames annotated with rich information: bounding box, vehicle type, vehicle orientation, vehicle count, vehicle re-identification, and weather condition. The dataset is divided into training and testing sets, with 42,200 and 17,800 frames, respectively, covering multiple cameras and different weather conditions. WebCamT is an appropriate dataset to evaluate domain adaptation methods, for it covers multiple city cameras and each camera is located in different intersection of the city with different perspectives and scenes. Thus, each camera data has different distribution from others. The dataset is quite challenging and in high demand of domain adaptation solutions, as it has $6,000,000$ unlabeled images from 200

cameras with only $60,000$ labeled images from 12 cameras. The experiments on WebCamT provide an interesting application of our proposed MDANs: when dealing with spatially and temporally large-scale dataset with much variations, it is prohibitively expensive and time-consuming to label large amount of instances covering all the variations. As a result, only a limited portion of the dataset can be annotated, which can not cover all the data domains in the dataset. MDAN provide an effective solution for this kind of application by adapting the deep model from multiple source domains to the unlabeled target domain.

We evaluate the proposed methods on different numbers of source cameras. Each source camera provides 2000 labeled images for training and the test set has 2000 images from the target camera. In each mini-batch, we randomly sample the same number of unlabeled target images as the source images. We implement the Hard-Max and Soft-Max version of MDANs according to Alg. 1, based on the basic vehicle counting network FCN described in (Zhang et al., 2017). Please refer to (Zhang et al., 2017) for detailed network architecture and parameters. The learning rate is initialized by 0.01 and adjusted by the first and second order momentum in the training process. The domain adaptation parameter is selected by cross validation. We compare our method with two baselines: *Baseline 1: FCN*. It is our basic network without domain adaptation as introduced in work (Zhang et al., 2017). *Baseline 2: DANN*. We implement DANN on top of the same basic network following the algorithm introduced in work (Ganin et al., 2016).

E More Related Work

A number of adaptation approaches have been studied in recent years. From the theoretical aspect, several theoretical results have been derived in the form of upper bounds on the generalization target error by learning from the source data. A keypoint of the theoretical frameworks is estimating the distribution shift between source and target. Kifer et al. (2004) proposed the $\mathcal{H}$-divergence to measure the similarity between two domains and derived a generalization bound on the target domain using empirical error on the source domain and the $\mathcal{H}$-divergence between the source and the target. This idea has later been extended to multisource domain adaptation (Blitzer et al., 2008) and the corresponding generalization bound has been developed as well. Ben-David et al. (2010) provide a generalization bound for domain adaptation on the target risk which generalizes the standard bound on the source risk. This work formalizes a natural intuition of DA: reducing the two distributions while ensuring a low error on the source domain and justifies many DA algorithms. Based on this work, Mansour et al. (2009a) introduce a new divergence measure: discrepancy distance, whose empirical estimate is based on the Rademacher complexity (Koltchinskii, 2001) (rather than the VC-dim). Other theoretical works have also been studied such as (Mansour & Schain, 2012) that derives the generalization bounds on the target error by taking use of the robustness properties introduced in (Xu & Mannor, 2012). See (Cortes et al., 2008; Mansour et al., 2009a;c) for more details.

Following the theoretical developments, many DA algorithms have been proposed, such as instance-based methods (Tsuboi et al., 2009); feature-based methods (Becker et al., 2013); and parameter-based methods (Evgeniou & Pontil, 2004). The general approach for domain adaptation starts from algorithms that focus on linear hypothesis class (Blitzer et al., 2006; Germain et al., 2013; Cortes & Mohri, 2014). The linear assumption can be relaxed and extended to the non-linear setting using the kernel trick, leading to a reweighting scheme that can be efficiently solved via quadratic programming (Huang et al., 2006; Gong et al., 2013). Recently, due to the availability of rich data and powerful computational resources, non-linear representations and hypothesis classes have been increasingly explored (Glorot et al., 2011; Baktashmotlagh et al., 2013; Chen et al., 2012; Ajakan et al., 2014; Ganin et al., 2016). This line of work focuses on building common and robust feature representations among multiple domains using either supervised neural networks (Glorot et al., 2011), or unsupervised pretraining using denoising auto-encoders (Vincent et al., 2008; 2010).

Recent studies have shown that deep neural networks can learn more transferable features for DA (Glorot et al., 2011; Donahue et al., 2014; Yosinski et al., 2014). Bousmalis et al. (2016) develop domain separation networks to extract image representations that are partitioned into two subspaces: domain private component and cross-domain shared component. The partitioned representation is utilized to reconstruct the images from both domains, improving the DA performance. Reference (Long et al., 2015) enables classifier adaptation by learning the residual function with reference to the target classifier. The main-task of this work is limited to the classification problem. Ganin

et al. (2016) propose a domain-adversarial neural network to learn the domain indiscriminate but main-task discriminative features. Although these works generally outperform non-deep learning based methods, they only focus on the single-source-single-target DA problem, and much work is rather empirical design without statistical guarantees. Hoffman et al. (2012) present a domain transform mixture model for multisource DA, which is based on non-deep architectures and is difficult to scale up.

Adversarial training techniques that aim to build feature representations that are indistinguishable between source and target domains have been proposed in the last few years (Ajakan et al., 2014; Ganin et al., 2016). Specifically, one of the central ideas is to use neural networks, which are powerful function approximators, to approximate a distance measure known as the $\mathcal{H}$-divergence between two domains (Kifer et al., 2004; Ben-David et al., 2007; 2010). The overall algorithm can be viewed as a zero-sum two-player game: one network tries to learn feature representations that can fool the other network, whose goal is to distinguish representations generated from the source domain between those generated from the target domain. The goal of the algorithm is to find a Nash-equilibrium of the game, or the stationary point of the min-max saddle point problem. Ideally, at such equilibrium state, feature representations from the source domain will share the same distributions as those from the target domain, and, as a result, better generalization on the target domain can be expected by training models using only labeled instances from the source domain.