LAMDA: Label Matching Deep Domain Adaptation with New Theoretical Analysis

Abstract

Recent progress in deep domain adaptation has shown that it can achieve higher predictive performance with better modeling capacity on complex domains (e.g., image, structural data, and sequential data) than its shallow rivals. The underlying idea of deep domain adaptation is to bridge the gap between source and target domains in a joint space so that a supervised classifier trained on labeled source data can be effectively transferred to the target domain. While this approach is powerful and practical, limited theoretical understandings have been developed to support its underpinning principle. In this paper, we provide a rigorous framework to explain why it is possible to close the gap between the target and source domains in the joint space. More specifically, we first study the loss incurred when performing transfer learning from the source to the target domain which provides a theory that can explain and generalize existing work in deep domain adaptation. This also enables us to further explain why closing the gap in the joint space can directly minimize the loss incurred for transfer learning between the two domains. Finally, based on the theoretical results developed, we propose the Label Matching Domain Adaptation (LAMDA) approach that outperforms the state-of-the-art baselines on the real-world datasets.

1 Introduction

Learning a discriminative classifier or other predictors in the presence of a shift between source (training) and target (testing) distributions is known as domain adaptation (DA) where its aim is to devise automatic methods to perform transfer learning from a source domain with labels to a target domain without labels. Existing approaches can be broadly categorized as shallow or deep DA. A number of shallow domain adaptation methods have been proposed in which data representations/features are given and fixed, notably [5, 30, 44, 26, 23, 1, 29].

Moving beyond using fixed features, deep domain adaptation (DDA) has recently been proposed to learn joint representations for both source and target data in order to minimize the divergence between them [19, 36, 40, 54, 17]. To do so, the source and target data are mapped to a joint feature space via a generator and the gap between source and target distributions is bridged in this joint space by minimizing the divergence between distributions induced from the source and target domains on this space. Popular choices of divergence include the Jensen-Shannon divergence [19, 60, 36, 54, 17], the maximum mean discrepancy [27] (MMD) distance [36], and the Wasserstein (WS) distance [13]. Although the idea of bridging the gap of the source and target domains in a joint feature space is an intuitive, plausible, and promising approach, to our best of knowledge, limited work exists to rigorously explain and provide a theoretical underpinning for this problem.

Our contributions. Using an optimal transport-based framework, we first study and analyze theoretical aspects of deep DA problem, connect its results to existing approaches, and then conduct extensive experiments to demonstrate its merits. Our contributions can be summarized as

- *Contribution 1*: We propose and study a scenario for unsupervised DA in which we do not assume that the hypothesis spaces for the source and target domains are identical, and generally study the setting where the hypothesis spaces for the target and source domains are shifted by a transformation mapping T. The purpose of the transformation mapping T is to reduce the data shift between source and target domains and to indicate which data instance $T(\boldsymbol{x})$ sampled from the source data distribution can be used as a substitute for a data instance $\boldsymbol{x}$ sampled from the target data distribution. Similar to other works [41, 2, 47, 74], we find that the performance of transfer learning totally depends on the data distribution shift and label

Preliminary work. Under review by AISTATS 2021. Do not distribute.

mismatch (shift) between two labeling mechanisms of source and target domains. Furthermore, our results also hold for any continuous loss function with mild assumptions and probabilistic label assignment (cf. Theorem 2).

- *Contribution 2:* We further assume that the transformation T is a deep neural network which can be decomposed into $T = H^2 \circ H^1$, wherein H^1 is the sub-network that maps data instances to an intermediate layer or a joint space as in other works of DDA. Under this assumption, we theoretically demonstrate that if we sort out the data distribution shift by learning a transformation T which minimizes a Wasserstein (WS) distance [65, 52] between the source distribution and the pushforward distribution of the target distribution via the transformation T, we equivalently bride the gap between source and target distributions in the joint space and minimize a reconstruction loss w.r.t the ground metric c of the WS distance (cf. Theorem 6). We speculate to interpret what it does mean by label mismatch (shift) in the joint space which further indicates that a perfect workaround for label mismatch does not seem possible due to the lack of target labels and the power of deep neural networks. More specifically, a sufficiently power deep net can map one class in the target domain to a wrong class in the source domain in the joint space for which the gap between the entire source and target distributions in the joint space is also negligible. To mitigate the negative impact of this label mismatching, we need to invoke inductive biases of models and data characteristics.

- *Contribution 3*: Inspired by the theoretical results, we propose ***LA****bel* ***M****atching* ***D****omain* ***A****daptation* (LAMDA) approach with the aim to minimize the discrepancy gap between two domains and simultaneously reduce the label mismatch in the joint space. Different from existing works, LAMDA employs a multi-class discriminator to be able to be aware of source class regions and an optimal transport [65, 52] based cost to encourage target examples for moving to their matching source class region in the joint space. We conduct the extensive experiments on the synthetic and real-world datasets to verify the theoretical results obtained and study the behaviors when the transport transformation T causes the total/partial match or mismatch of two labeling assignment mechanisms in the joint space, followed by the experiments on the real-world datasets to compare LAMDA with state-of-the-art baselines. The experimental results on the real-world datasets show that our LAMDA is able to reduce the label mismatch in the joint space, hence achieving better performances.

2 Related Work

The theory of domain adaptation has been studied from several perspectives [48]. Several work have been proposed to characterize the gap between general losses of source and target, typically [41, 2, 47, 74, 11]. Other work, typically [4, 3, 74], studies the impossibility theorems for domain adaptation, aiming to indicate the conditions under which it is nearly impossible to perform a good transfer learning. The PAC-Bayesian theory concerning how the PAC–Bayesian theory can help to theoretically understand domain adaptation through the weighted majority vote learning point of view has been rigorously studied in [20, 21]. Optimal transport theory has been leveraged in [12] to cope with data and label shift in domain adaptation. Specifically, our theory development, motivations, and obtained results in Section 3.2 are totally different to those in [12]. In addition, we also compare our LAMDA to DeepJDOT [14] (a deep domain adaptation approach developed based on theoretical foundation of [12]) and other OT-based DDA approaches SWD [34], DASPOT [71], and RWOT [72].

3 Main Theoretical Results

Let the data spaces of the source and target domains be $\mathcal{X}^s$ and $\mathcal{X}^t$ respectively. These are endowed with data generation densities $p^s(\boldsymbol{x})$ and $p^t(\boldsymbol{x})$ whose probability measures are $\mathbb{P}^s$ and $\mathbb{P}^t$. We also denote the supervisor distributions that assign labels to data samples in the source and target domains by $p^s(y \mid \boldsymbol{x})$ and $p^t(y \mid \boldsymbol{x})$ [63].

On the source domain, denote by $\mathcal{H}^s := \{h^s : \mathcal{X}^s \to \mathbb{R}\}$ the hypothesis set whose elements are used to predict labels source data. Throughout this paper, we assume that $T : \mathcal{X}^t \to \mathcal{X}^s$ is a mapping. Based on the formulation of hypothesis set $\mathcal{H}^s$, we define hypothesis set on target domain as $\mathcal{H}^t := \{h^t : \mathcal{X}^t \to \mathbb{R} \mid h^t(\cdot) = h^s(T(\cdot)) \text{ for some } h^s \in \mathcal{H}^s\}$.

The intuition behind these definitions and assumptions is that with $\boldsymbol{x} \sim \mathbb{P}^t$, we use the mapping T to reduce the difference between two domains and then apply a hypothesis $h^s \in \mathcal{H}^s$ to predict the label of $\boldsymbol{x}$. This gives rise to the question about the key properties of the transformation T so that we can employ the hypothesis $h^t = h^s \circ T$ to predict labels of target data where $\circ$ represents the composition function.

Let $P^{\#} := T_{\#}\mathbb{P}^t$ be the pushforward probability distribution induced by transporting $\mathbb{P}^t$ via T, hence consequently inducing a new domain termed the *transport domain* whose data are generated from $p^{\#}(\boldsymbol{x})$

being the density of $\mathbb{P}^{\#}$. We further define the supervisor distribution for the transport domain as $p^{\#}(y \mid T(\boldsymbol{x})) = p^t(y \mid \boldsymbol{x})$ for any $\boldsymbol{x} \sim \mathbb{P}^t$. To ease the presentation, we denote the general expected loss as:

$$R^{a,b}(h) := \int \ell(y, h(\boldsymbol{x}))\, p^b(y \mid \boldsymbol{x})\, p^a(\boldsymbol{x})\, dyd\boldsymbol{x},$$

where a, b are in the set $\{s, t, \#\}$ and $\ell(\cdot,\cdot)$ specifies a loss function. In addition, we shorten $R^{a,a}$ as R^a. Furthermore, given a hypothesis $h^s \in \mathcal{H}^s$ and $h^t = h^s \circ T$, we measure the variance of general losses of h^s when predicting on the source domain and general losses of h^t when predicting on the target domain as:

$$\Delta R\left(h^s, h^t\right) := \left|R^t\left(h^t\right) - R^s\left(h^s\right)\right|.$$

Finally, for the simplicity of the results in the paper, we consider solely the case of binary classification where the label $y \in \{-1, 1\}$. Please refer to our supplementary material for the relevant background and the details of all proof.

3.1 Gap between target and source domains

In this subsection, we investigate the variance $\Delta R(h^s, h^t)$ between the expected loss in target domain $R^t(h^t)$ and the expected loss in source domain $R^s(h^s)$ where $h^t = h^s \circ T$. We embark on with the following simple yet key proposition indicating the connection between $R^t(h^t)$ and $R^{\#}(h^s)$.

Proposition 1. *As long as $h^t = h^s \circ T$, we have $R^t(h^t) = R^{\#}(h^s)$.*

To derive a relation between $R^t(h^t)$ and $R^s(h^s)$, we make the following mild assumption with loss function:

(A.1) $\sup_{h^s \in \mathcal{H}^s, \boldsymbol{x} \in \mathcal{X}^s, y \in \{-1,1\}} |\ell(y, h^s(\boldsymbol{x}))| := M < \infty.$

With simple algebra manipulation, the above assumption is satisfied when ℓ is a bounded loss, e.g., logistic or 0-1 loss or ℓ is any continuous loss, $\mathcal{X}^s$ is compact, and $\sup_{\boldsymbol{x} \in \mathcal{X}^s} |h^s(\boldsymbol{x})| < \infty$. Equipped with Assumption (A.1), we have the following key result demonstrating the upper bound of $R^t(h^t)$ in terms of $R^s(h^s)$.

Theorem 2. *Assume that Assumption (A.1) holds. Then, for any hypothesis $h^s \in \mathcal{H}^s$, the following inequality holds:*

$$\Delta R\left(h^s, h^t\right) \leq M\left(WS_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) + \mathbb{E}_{\mathbb{P}^t}\left[\|\Delta p(y \mid \boldsymbol{x})\|_1\right]\right)$$

, where $\Delta p(y \mid \boldsymbol{x})$ is given by $\Delta p(y \mid \boldsymbol{x}) := \|p^t(y \mid \boldsymbol{x}) - p^s(y \mid T(\boldsymbol{x}))\|_1$, and $WS_{c_{0/1}}(\cdot,\cdot)$ is the Wasserstein distance with respect to the cost function $c_{0/1}(\boldsymbol{x}, \boldsymbol{x}') = \mathbf{1}_{\boldsymbol{x} \neq \boldsymbol{x}'}$, which returns 1 if $\boldsymbol{x} \neq \boldsymbol{x}'$ and 0 otherwise.

Remark 3. If the following assumptions hold:

(i) The transformation mapping $T(\boldsymbol{x}) = \boldsymbol{x}$, i.e., we use the same hypothesis set for both the source and target domains,

(ii) The loss $\ell(y, h(\boldsymbol{x})) = \frac{1}{2}|y - h(\boldsymbol{x})|$ where we restrict to consider hypothesis $h : \mathcal{X} \to \{-1, 1\}$,

then we recover Theorem 1 in [2].

Remark 4. When $\mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) = 0$ (i.e., $T_{\#}\mathbb{P}^t = \mathbb{P}^s$), and there is a harmony between two supervisors of source and target domains (i.e., $p^s(y \mid T(\boldsymbol{x})) = p^t(y \mid \boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbb{P}^t$), as suggested by Theorem 2, we can do a perfect transfer learning without loss of performance. This fact is summarized in the following corollary.

Corollary 5. *Assume that $T_{\#}\mathbb{P}^t = \mathbb{P}^s$ and the source and target supervisor distributions are harmonic in the sense that $p^s(y \mid T(\boldsymbol{x})) = p^t(y \mid \boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbb{P}^t$. Then, we can do a perfect transfer learning between the source and target domains.*

3.2 Optimization via Wasserstein metric

Corollary 5 suggest that we can do a perfect transfer learning from the source to target domains if we can point out a map that transports the target to source distributions and two supervisor distributions are harmonic via this map. This is consistent with what is achieved in Theorem 2 for which the upper bound of the loss variance $\Delta R(h^s, h^t)$ vanishes. Particularly, the upper bound in Theorem 2 consists of two terms wherein the first term quantifies how distant the transport and source domains and the second term relates to the discrepancy of two supervisor distributions. Still, from Theorem 2, we obtain the following inequality:

$$R^t(h_t) \leq R^s(h_s) + M\left(\mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) + \mathbb{E}_{\mathbb{P}^t}\left[\|\Delta p(y \mid \boldsymbol{x})\|_1\right]\right),$$

which requires us to find the best hypothesis h_s^* and transformation T^* for minimizing the general loss $R^s(h_s)$ and the remaining term. To minimize the remaining term, due to the lack of target labels, it is natural to focus on minimizing the first term $\mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right)$ by restricting the transformation T in the family of those what can transport the target to source distributions. By this restriction, the problem of interest boils down to answering the question: *among the maps T that transport the target to source distributions which transformation incurs the minimal discrepancy as specified in the second term of the upper bound in Theorem 2.*

We further tackle the task of finding the maps that transport the target to source distributions via the Wasserstein distance with respect to the ground metric c and $p > 0$ as:

$$\min_H WS_{c,p}\left(H_{\#}\mathbb{P}^t, \mathbb{P}^s\right), \tag{1}$$

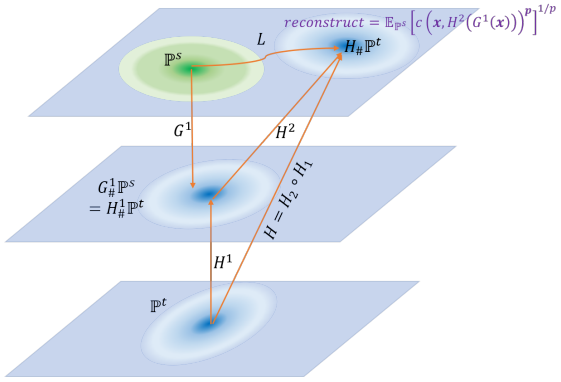

Figure 1: The mapping $H = H^2 \circ H^1$ maps from the target to source domains. We minimize $D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)$ to close the discrepancy gap of the source and target domains in the joint space. In addition, we further minimize the reconstruction terms to avoid the mode collapse.

where $WS_{c,p}\left(\mathbb{P}, \mathbb{Q}\right) = \inf_{T_{\#}\mathbb{P}=\mathbb{Q}} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}}\left[c\left(\boldsymbol{x}, T\left(\boldsymbol{x}\right)\right)^p\right]^{1/p}$ is a Wasserstein distance between two distributions $\mathbb{P}$, $\mathbb{Q}$.

Let $\mathcal{Z}$ be an intermediate space (i.e., the joint space $\mathcal{Z} = \mathbb{R}^m$). We consider the composite mappings H: $H\left(\boldsymbol{x}\right) = H^2\left(H^1\left(\boldsymbol{x}\right)\right)$ where H^1 is a mapping from the target domain $\mathcal{X}^t$ to the joint space $\mathcal{Z}$ and H^2 maps from the joint space $\mathcal{Z}$ to the source domain $\mathcal{X}^s$ (note that if $\mathcal{Z} = \mathcal{X}^s$ then $H^2 = id$ is the identity function). Based on that structure on H, we can recast the optimization with Wasserstein metric in Eq. (1) to the following optimization problem:

$$\min_{H^1, H^2} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right). \tag{2}$$

In the following theorem, we demonstrate that the above optimization problem can be equivalently transformed into another form involving the joint space (see Figure 1 for an illustration of that theorem).

Theorem 6. *The optimization problem* (2) *is equivalent to the following optimization problem:*

$$\min_{H^1, H^2} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}, \tag{3}$$

where G^1 is a map from the source domain $\mathcal{X}^s$ to the joint space $\mathcal{Z}$.

A few comments are in order. First, it is interesting to interpret G^1 and H^1 as two generators that map the source and target domains to the common joint space $\mathcal{Z}$ respectively. The constraint $H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s$ further indicates that the gap between the source and target distributions is closed in the joint space via two generators G^1 and H^1. Furthermore, H^2 maps from the joint space to the source domain and aims to reconstruct G^1. Similar to [59], we do relaxation and arrive at the optimization problem:

$$\min_{H^1, H^2, G^1}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} + \alpha D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)\right), \tag{4}$$

where $D\left(\cdot,\cdot\right)$ specifies a divergence between two distributions over the joint space and $\alpha > 0$. When the trade-off parameter α approaches $+\infty$, the solution of the relaxation problem in Eq. (4) approaches the optimal solution in Eq. (3).

Second, if we denote the source training set by $\mathcal{D}^s = \left\{\left(\boldsymbol{x}^s_1, y_1\right), ..., \left(\boldsymbol{x}^s_{N_s}, y_{N_s}\right)\right\}$, to enable the transfer learning, we can train a supervised classifier $\mathcal{C}$ on $\mathcal{A}\left(\mathcal{D}^s\right) = \left\{\left(\mathcal{A}\left(\boldsymbol{x}^s_1\right), y_1\right), \ldots, \left(\mathcal{A}\left(\boldsymbol{x}^s_{N_s}\right), y_{N_s}\right)\right\}$. Then, the final optimization problem becomes

$$\min_{H^1, H^2, G^1}\left(\beta\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} + \alpha D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^s}\left[\ell\left(y, \mathcal{C}\left(\mathcal{A}\left(\boldsymbol{x}\right)\right)\right)\right]\right), \tag{5}$$

where $\beta > 0$, $\mathcal{A}$ is either G^1 or the identity map, and we overload $\mathcal{D}^s$ to represent the empirical distribution over the source training set. Moreover, to reduce the discrepancy gap $D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)$ in Eq. (5), one can use the GAN principle [25] to implicitly minimize a JS divergence or explicitly minimize other divergences and distances (e.g., a $f-$divergence, a MMD or WS distance).

Finally, since G^1 and H^1 are two maps from the source and target domains to the joint space, we can further define two source and target supervisor distributions on the joint space as $p^{\#,s}\left(y \mid G^1\left(\boldsymbol{x}\right)\right) = p^s\left(y \mid \boldsymbol{x}\right)$ for $\boldsymbol{x} \sim \mathbb{P}^s$ and $p^{\#,t}\left(y \mid H^1\left(\boldsymbol{x}\right)\right) = p^t\left(y \mid \boldsymbol{x}\right)$ for $\boldsymbol{x} \sim \mathbb{P}^t$. With respect to the joint space, the second term of the upper bound in Theorem 2 can be rewritten as in the following corollary.

Corollary 7. *The second term of the upper bound in Theorem 2 can be rewritten as*

$$\mathbb{E}_{\mathbb{P}^t}\left[\left\|p^{\#,s}\left(y \mid G^1\left(H^2\left(H^1\left(\boldsymbol{x}\right)\right)\right)\right) - p^{\#,t}\left(y \mid H^1\left(\boldsymbol{x}\right)\right)\right\|_1\right]. \tag{6}$$

We now analyze the ideal scenario to speculate the data distribution and label shifts in the joint space $\mathcal{Z}$. The optimization problem in (3) peaks its minimization at 0 when $G^1_{\#}\mathbb{P}^s = H^1_{\#}\mathbb{P}^t$ and $G^1 \circ H^2 = id$ (i.e., the identity function), which further implies that

$$H_{\#}\mathbb{P}^t = H^2_{\#}\left(H^1_{\#}\mathbb{P}^t\right) = H^2_{\#}\left(G^1_{\#}\mathbb{P}^s\right) = \left(G^1 \circ H^2\right)_{\#}\mathbb{P}^s = \mathbb{P}^s,$$

and $\mathrm{WS}_{c,p}\left(H_{\#}\mathbb{P}^t, \mathbb{P}^s\right) = 0$. Under that ideal scenario, the label mismatch term in Eq. (6) reduces to

$$\mathbb{E}_{\mathbb{P}^t}\left[\left\|p^{\#,s}\left(y \mid H^1\left(\boldsymbol{x}\right)\right) - p^{\#,t}\left(y \mid H^1\left(\boldsymbol{x}\right)\right)\right\|_1\right]. \tag{7}$$

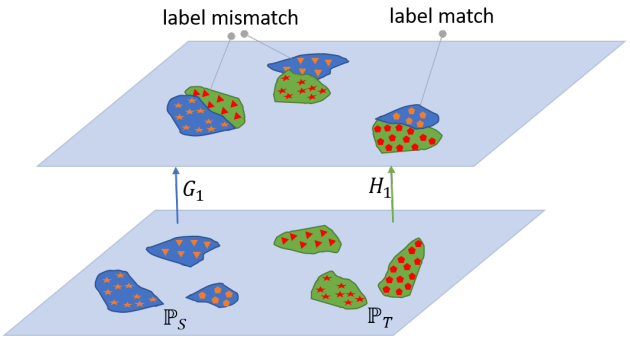

Figure 2: Label match and mismatch in the joint space.

We note that because $G^1_{\#}\mathbb{P}^s = H^1_{\#}\mathbb{P}^t$, $H^1(\boldsymbol{x})$ with $\boldsymbol{x} \sim \mathbb{P}^t$ is moved to a class region of source data (e.g., the class region of class m^s) in the joint space and would be classified to class m^s by a source classifier (i.e., the one that mimics $p^{\#,s}\left(y \mid H^1(\boldsymbol{x})\right)$). Assume that the ground-truth label of $\boldsymbol{x}$ is m^t, the minimization of the label mismatch (shift) term in Eq. (7) suggests that $m^s = m^t$ or H^1 should transport $\boldsymbol{x}$ to the proper class region to reduce the label mismatch. However, in unsupervised domain adaptation, since we do not possess any target labels and the neural network generator H^1 can totally map a class region of the target domain to a wrong class region of the source domain in the joint space (cf. Figure 2), it is impossible to tackle perfectly the label mismatch (shift) term. In the ablation study in Section 5.1.2, we investigate the influence of the label mismatch to the performance of transfer learning, which shows that this label mismatch in the joint space can significantly affect the performance of transfer learning. Although it is nearly impossible to perfectly tackle the label mismatch problem, we propose an approach to mitigate the negative impact of the label mismatching in the next section.

4 Label Matching Domain Adaptation

As shown our ablation study (cf. Section 5.1.2), reducing label mismatch in the joint space, captured by $D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)$, is a key factor to improve predictive performance of deep domain adaptation. When bridging the discrepancy gap $D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)$ (cf. Eq. (4)) between the source and target domains in the joint space, we propose optimal transport based approach in which each unlabeled target example aims to find the most suitable class in the source domain to move to based on the guidance from a multi-class discriminator d that can emphasize the class regions in the source domain. We name the proposed method as ***LA**bel **M**atching **D**omain **A**daptation* (LAMDA).

Methodological idea. To bridge the discrepancy gap $D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)$, we employ the GAN principle. Moreover, to be able to simultaneously discriminate the source and target examples, and distinguish the classes of the source domain, we employ a multi-class discriminator d with $M+1$ probability output (M is the number of classes) in which for $\boldsymbol{x} \sim \mathbb{P}^s$ and $1 \leq m \leq M$, the m-th probability output specifies the probability of that example generated from the m-th class mixture of the source domain, i.e., $d_m\left(G^1(\boldsymbol{x})\right) = \mathbb{P}\left(y = m \mid \boldsymbol{x}\right)$ and for $\boldsymbol{x} \sim \mathbb{P}^t$, the $M+1$ probability output specifies the probability of that example generated from the target distribution, i.e., $d_{M+1}\left(H^1(\boldsymbol{x})\right) = \mathbb{P}\left(y = M+1 \mid \boldsymbol{x}\right)$.

Training method. Since the discriminator can discriminate the source and target examples, and distinguish the classes of the source domain, we solve the following OP for d:

$$\max_d \Bigg(\mathcal{L}_d := \sum_{m=1}^{M} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^s \wedge y=m}\left[\log d_m\left(G^1(\boldsymbol{x})\right)\right] + \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^t}\left[\log d_{M+1}\left(H^1(\boldsymbol{x})\right)\right] + \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[\log\left(1 - d_{M+1}\left(G^1(\boldsymbol{x})\right)\right)\right] \Bigg).$$

To train the generators G^1, H^1, we update them as follows:

i) $G^1(\boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbb{P}^s$ moves to the region of *high values* for $d_{M+1}(\cdot)$ (i.e., *the region of target examples*) by $\max_{G^1} I\left(G^1\right)$ where $I\left(G^1\right) := \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[\log d_{M+1}\left(G^1(\boldsymbol{x})\right)\right]$.

ii) $H^1(\boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbb{P}^t$ moves to *one of class mixture* in the source domain accordingly. Recalling that $d_m(\boldsymbol{x})$ represents the *likelihood* of $\boldsymbol{x}$ w.r.t the m-th class mixture, we employ $-\log \mathbb{P}(y = m \mid \boldsymbol{x}) = -\log d_m\left(H^1(\boldsymbol{x})\right)$ as the *cost incurred* if we move $H^1(\boldsymbol{x})$ to $\mathcal{D}^s_m = \{(x,y) \in \mathcal{D}^s \mid y = m\}$. To specify the probabilities that transports $\boldsymbol{x} \sim \mathbb{P}^t$ to the class mixtures, we use a transportation probability network $T(\boldsymbol{x})$ for which $T_m(\boldsymbol{x})$ points out probability to transport $\boldsymbol{x}$ to $\mathcal{D}^s_m$. Therefore, the *total transport cost* incurred (see Figure 3) is

$$TC\left(H^1\right) := \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^t}\left[-\sum_{m=1}^{M} T_m(\boldsymbol{x}) \log d_m\left(H^1(\boldsymbol{x})\right)\right].$$

In addition, we also push $H^1(\boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbb{P}^t$ the the region of *low values* for $d_{M+1}(\cdot)$ (i.e., *the region of source examples*) by $\max_{H^1} J\left(H^1\right)$ where $J\left(H^1\right) := \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^t}\left[\log\left(1 - d_{M+1}\left(H^1(\boldsymbol{x})\right)\right)\right]$. We obtain the following OP for H^1:

$$\min_{H^1}\left(TC\left(H^1\right) - J\left(H^1\right)\right).$$

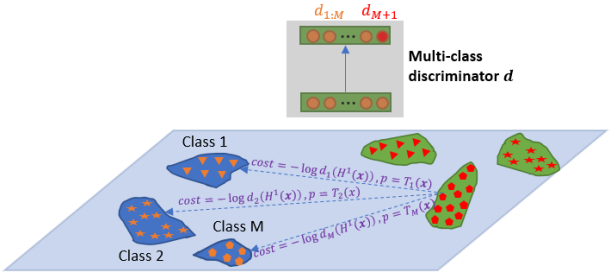

Figure 3: Transport cost of a target data instance $\boldsymbol{x}$ w.r.t the multi-class discriminator d and the transportation probability network T.

Finally, the OP to update $G^1, H^{1:2}$ by minimizing

$$\min_{G^1,H^{1:2}} \mathcal{L}_{gen},$$

where we have defined

$$\mathcal{L}_{gen} := \alpha\left[-I\left(G^1\right) - J\left(H^1\right) + TC\left(H^1\right)\right] - \beta R\left(H^2, G^1\right)$$

in which $R\left(H^2, G^1\right) = \mathbb{E}_{\mathbb{P}^s}\left[\left\|H^2\left(G^1\left(\boldsymbol{x}\right)\right) - \boldsymbol{x}\right\|^2\right]$ is the reconstruction term. Here we note that in our experiments, we use the classifier $\mathcal{C}$ (cf. Eq. (4)) for the transportation probability network T (i.e., $T := \mathcal{C}$).

More specifically, let us denote

$$\mathcal{L}_{\mathcal{C}} := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^s}\left[\ell\left(y, \mathcal{C}\left(G^1\left(\boldsymbol{x}\right)\right)\right)\right].$$

We then update $G^1, H^{1:2}$, and $\mathcal{C}$ by minimizing

$$\mathcal{L}_{\mathcal{C}} + \mathcal{L}_{gen}.$$

Please refer to the supplementary material for the min-max objective function and the technical detail of how to update the components $\mathcal{C}, G^1, H^{1:2}$, and d in our LAMDA.

5 Experiment

5.1 Ablation study

We conduct experiments to verify the our proposed theory and speculate the influence of class alignment in the joint space to the performance of transfer learning.

5.1.1 Verification of The Proposed Theory on Synthetic Dataset

Synthetic Dataset for the Source and Target Domains. We generate two synthetic labeled datasets for the source and target domains. More details of how to generate source and target examples can be found in our supplementary material.

Deep Domain Adaptation on the Synthetic Dataset. More details of the network architectures of $G^1, H^1, H^2, \mathcal{C}, \mathcal{D}$, the cost function, and the objective function can be found in our supplementary material.

Verification of Our Theory for Unsupervised Domain Adaptation. In this experiment, we assume that none of data example in the target domain has label. We measure three terms, namely $\left|R\left(h^t\right) - R\left(h^s\right)\right|$, $WS\left(\mathbb{P}^s, \mathbb{P}^{\#}\right)$, and $\mathbb{E}_{\mathbb{P}^t}\left[\left\|\Delta p\left(y \mid \boldsymbol{x}\right)\right\|_1\right]$ ($M = 1$ since we are using the logistic loss) as defined in Theorem 2 across the training progress. Actually, we approximate $R\left(h^t\right), R\left(h^s\right)$ using the corresponding empirical losses. As shown in Figure 4 (middle), the green plot is always above the blue plot and this empirically confirms the inequality in Theorem 2. Furthermore, the fact that three terms consistently decrease across the training progress indicates an improvement when $\mathbb{P}^{\#}$ is shifting toward $\mathbb{P}^s$. This improvement is also reflected in Figure 4 (left and right) wherein the target accuracy and empirical loss gradually increase and decrease accordingly.

5.1.2 The Effect of Class Alignment in the Joint Space.

In this experiment, we inspect the influence of the harmony of two labeling assignment mechanisms to the predictive performance. In particular, we assume that a portion ($r = 5\%, 15\%, 25\%, 50\%$) of the target domain has label and consider two settings: i) the labels of the target and source domains are totally properly matched in the joint space (i.e., 0 matches 0, 1 matches 1,..., and 9 matches 9) and ii) the labels of the target and source domains are totally improperly matches in the joint space (i.e., 0 matches 1, 1 matches 2,..., and 9 matches 0).

To push a specific labeled portion of the target domain to the corresponding label portion of the source domain in the joint space (the label i to i in the first setting and the label i to $(i+1) \bmod 10$ in the second setting for $i = 0, 1, \ldots, 9$), we again make use of GAN principle and employ additional discriminators to push the corresponding labeled portions together. Note that the parameters of the additional discriminators and the primary discriminator (used to push the target data toward source data in the joint space) are tied up to the penultimate layer.

It can be observed from Table 3 that for the case of proper matching, when increasing the ratio of labeled portion, we increase the chance to match the corresponding labeled portions properly, hence significantly improving the predictive performance. In contrast, for the case of improper matching, when increasing the ratio of labeled portion, we increase the chance to match the corresponding labeled portions improperly, hence significantly reducing the predictive performance.

5.2 Our LAMDA Versus the Baselines

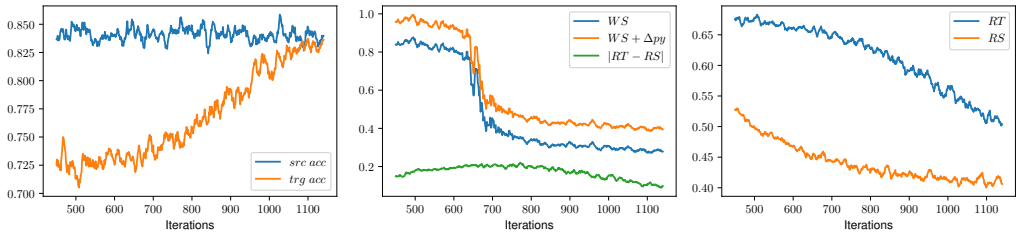

Figure 4: Left: the accuracies on the source and target datasets. Middle: the plots of three terms in Theorem 2. Right: the plot of empirical losses on the source and target datasets.

Table 1: The experimental results in percent of our LAMDA and the baselines. The best performance is emphasized in bold and the runner-up performance is underlined.

Source	**MNIST**	**USPS**	**MNIST**	**SVHN**	**MNIST**	**DIGITS**	**SIGNS**	**CIFAR**	**STL**
Target	**USPS**	**MNIST**	**MNIST-M**	**MNIST**	**SVHN**	**SVHN**	**GTSRB**	**STL**	**CIFAR**
MMD [36]	-	-	76.9	71.1	-	88.0	91.1	-	-
DANN [19]	-	-	81.5	71.1	35.7	90.3	88.7	-	-
DRCN [22]	-	-	-	82.0	40.1	-	-	66.4	58.7
DSN [7]	-	-	83.2	82.7	-	91.2	93.1	-	-
kNN-Ad [53]	-	-	86.7	78.8	40.3	-	-	-	-
PixelDA [6]	-	-	98.2	-	-	-	-	-	-
ATT [49]	-	-	94.2	86.2	52.8	92.9	96.2	-	-
Π-model [16]	-	-	-	92.0	71.4	94.2	98.4	76.3	64.2
ADDA [61]	89.4	90.1	-	76.0	-	-	-	-	-
CyCADA [29]	95.6	96.5	-	90.4	-	-	-	-	-
MSTN [70]	92.9	97.6	-	91.7	-	-	-	-	-
CDAN [37]	95.6	98.0	-	89.2	-	-	-	-	-
MCD [50]	94.2	94.1	-	96.2	-	-	94.4	-	-
PFAN [9]	95.0	-	-	93.9	57.6	-	-	-	-
DADA [58]	96.1	96.5	-	95.6	-	-	-	-	-
DeepJDOT[14]	95.7	96.4	92.4	96.7	30.8	84.2	70.0	61.6	49.6
DASPOT [71]	97.5	96.5	94.9	96.2	-	-	-	-	-
GPDA [31]	96.5	96.4	-	98.2	-	-	96.2	-	-
SWD [34]	98.1	97.1	90.9	98.9	49.5	88.7	98.6	65.3	52.1
rRevGrad+CAT [15]	94.0	96.0	-	98.8	-	-	-	-	-
SHOT [35]	98.0	**98.4**	-	98.9	-	-	-	-	-
RWOT [72]	98.5	97.5	-	98.8	-	-	-	-	-
LAMDA	**99.5**	98.3	**98.4**	**99.5**	**82.1**	**95.9**	**99.2**	**78.0**	**71.6**

We conduct the experiments to compare our LAMDA against the state-of-the-art baselines on the digit, traffic sign and natural scene, Office-Home, and Office-31 datasets. The network architectures and implementation specification can be found in our supplementary material. In addition to the baselines in general DDA, we also compare our LAMDA to the ones developed based on the theory of optimal transport including DeepJDOT [14], SWD [34], DASPOT [71], and RWOT [72].

5.2.1 Experimental Results on Digit, Traffic Sign and Natural Scene Datasets

We choose some datasets widely used in the domain adaptation literature, including MNIST [33], MNIST-M [18], Synthetic Digits (SYN DIGITS) [18], Street View House Numbers (SVHN) [43], Synthetic Traffic Signs (SIGNS) [42], German Traffic Signs Recognition Benchmark (GTSRB) [56], CIFAR-10 (CIFAR) [32], and STL-10 (STL) [10]. The experimental results in Table 1 shows that our LAMDA outperforms other baselines on most of digit datasets. To demonstrate that our LAMDA can reduce the label mismatch, we use t-SNE [62] to visualize the source and target data in the joint space. As shown in Figures 5, our LAMDA successfully forces the clusters of the same class in two domains mixing up together to reduce the label mismatch, hence achieving high predictive performance. Those observations again support our theoretical claim.

5.2.2 Experimental Results on Office-Home Dataset

This dataset was introduced in [64], which consists of roughly 15,500 images in a total of 65 object classes. Images belong to 4 different domains: Artistic (Ar), Clip Art (Cl), Product (Pr) and Real-world (Rw). Due to the shortage of data for training and test, this dataset is much more challenging for the domain adaptation task. The input of all baselines and our proposed model is the features extracted via the pre-trained VGG-16

Table 2: The test set accuracy (%) on Office-Home dataset using VGG-16 as feature extractor.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Mean
GFK [24]	21.60	31.72	38.83	21.63	34.94	34.20	24.52	25.73	42.92	32.88	28.96	50.89	32.40
TCA [44]	19.93	32.08	35.71	19.00	31.36	31.74	21.92	23.64	42.12	30.74	27.15	48.68	30.34
CORAL [57]	27.10	36.16	44.32	26.08	40.03	40.33	27.77	30.54	50.61	38.48	36.36	57.11	37.91
JDA [38]	25.34	35.98	42.94	24.52	40.19	40.90	25.96	32.72	49.25	35.10	35.35	55.35	36.97
DAN [36]	30.66	42.17	54.13	32.83	47.59	49.58	29.07	34.05	56.70	43.58	38.25	62.73	43.46
DANN [19]	33.33	42.96	54.42	32.26	49.13	49.76	30.44	38.14	56.76	44.71	42.66	64.65	44.94
DAH [64]	31.64	40.75	51.73	34.69	51.93	52.79	29.91	39.63	60.71	44.99	45.13	62.54	45.54
DeepJDOT [14]	39.73	50.41	62.49	39.52	54.35	53.15	36.72	39.24	63.55	**52.29**	45.43	70.45	50.67
LAMDA	**42.11**	**59.23**	**62.84**	**39.92**	**64.19**	**60.09**	**38.27**	**48.51**	**66.74**	50.62	**56.52**	**75.68**	**55.39**

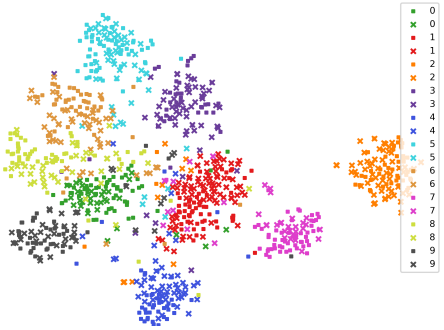

Figure 5: Visualization in the joint space of LAMDA for DIGITS → SVHN. The square/cross markers represent the source/target data.

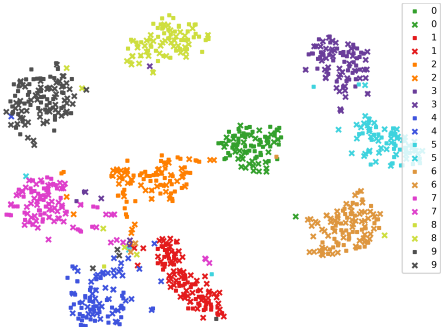

Figure 6: Visualization in the joint space of LAMDA for MNIST → MNIST-M.

[55]. The experimental results in Table 2 on Office-Home shows that our LAMDA surpasses the baselines on most of cases.

5.2.3 Experimental Results on Office-31 Dataset

We conduct experiments of Office-31 dataset using ResNet-50 [28] as a feature extractor. Experimental results show that our LAMDA surpasses the baselines on most of cases with the mean accuracy exceeding the runner-up baseline by 2.2%.

6 Conclusion

Deep domain adaptation is a recent powerful learning framework which aims to address the problem of scarcity of qualified labeled data for supervised learning.

Table 3: The variation of predictive performance in percentage as increasing the ratio of labeled portion when the labels of the target domain are properly or improperly matched to those in the source domain. Note that we emphasize in bold and italic/bold the best and worse performance.

	Proper match				**Improper match**				**Base**
r	5%	15%	25%	50%	5%	15%	25%	50%	0%
MNIST→MNIST-M	86.4	88.8	92.9	**93.2**	75.5	70.2	64.5	***58.4***	81.5
SVHN→MNIST	72.3	74.1	76.2	**77.5**	69.8	60.8	56.8	***66.4***	71.0

Table 4: The test set accuracy (%) on Office-31 dataset using ResNet-50 as a feature extractor.

Method	A→W	A→D	D→W	W→D	D→A	W→A	Mean
RTN [39]	84.5	77.5	96.8	99.4	66.2	64.8	81.6
MADA [46]	90.0	87.8	97.4	99.6	70.3	66.4	85.2
GTA [51]	89.5	87.7	97.9	99.8	72.8	71.4	86.5
iCAN [73]	92.5	90.1	98.8	**100.0**	72.1	69.9	87.2
CDAN-E [37]	94.1	92.9	98.6	**100.0**	71.0	69.3	87.7
JDDA [8]	82.6	79.8	95.2	99.7	57.4	66.7	80.2
SymNets [75]	90.8	93.9	98.8	**100.0**	74.6	72.5	88.4
TADA [69]	94.3	91.6	98.7	99.8	72.9	73.0	88.4
MEDA [66]	86.2	85.3	97.2	99.4	72.4	74.0	85.7
CAPLS [68]	90.6	88.6	98.6	99.6	75.4	76.3	88.2
TPN [45]	91.2	89.9	97.7	99.5	70.5	73.5	87.1
SPL [67]	92.7	93.0	98.7	99.8	76.4	76.8	89.6
DeepJDOT [14]	88.9	88.2	98.5	99.6	72.1	70.1	86.2
MDD[74]	94.5	93.5	98.4	**100.0**	74.6	72.2	88.9
RWOT [72]	95.1	94.5	**99.5**	**100.0**	77.5	77.9	90.8
LAMDA	**95.2**	**96.0**	98.5	99.8	**87.3**	**84.4**	**93.0**

To enable transferring the learning across the source and target domains, deep domain adaptation tries to bridge the gap between the source and target distributions in a joint feature space. Although this idea is powerful and has empirically demonstrated its success in several recent work, its theoretical underpinnings are lacking and limited. To this end, using the main tool of the Wasserstein distances, we have established a firm theoretical foundation for deep domain adaptation with extensive experimental results to demonstrate its merits. Our theory provides a much more stronger theoretical results with more realistic assumption for real-world applications compared with existing work. It further consolidates the rationale for deep domain adaptation approach using a joint space for DDA.

References

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

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, May 2010.

[3] S. Ben-David and R. Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, pages 139–153, 2012.

[4] S. Ben-David and R. Urner. Domain adaptation—can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, 70(3):185–202, March 2014.

[5] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, July 2006.

[6] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.

[7] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351, 2016.

[8] C. Chen, Z. Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *AAAI*, 2019.

[9] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang. Progressive feature alignment for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 627–636. Computer Vision Foundation / IEEE, 2019.

[10] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[11] C. Cortes, M. Mohri, and Andrés M. Medina. Adaptation based on generalized discrepancy. *The Journal of Machine Learning Research*, 20(1):1–30, 2019.

[12] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017.

[13] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2017.

[14] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11208 of *Lecture Notes in Computer Science*, pages 467–483, 2018.

[15] Z. Deng, Y. Luo, and J. Zhu. Cluster alignment with a teacher for unsupervised domain adaptation, 2019.

[16] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

[17] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

[18] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation, 2014.

[19] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189, 2015.

[20] P. Germain, A. Habrard, F. Laviolette, and E. Morvant. A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, ICML'13, 2013.

[21] P. Germain, A. Habrard, F. Laviolette, and E. Morvant. A new pac-bayesian perspective on domain

adaptation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 859–868, 2016.

[22] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[23] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 222–230, Atlanta, Georgia, USA, 17–19 Jun 2013.

[24] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[26] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 999–1006, Washington, DC, USA, 2011. IEEE Computer Society.

[27] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

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

[29] J. Hoffman, E. Tzeng, T. Park, J-Y Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1989–1998, 2018.

[30] J. Huang, A. Gretton, Karsten M. B., B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, 2007.

[31] M. Kim, P. Sahu, B. Gholami, and V. Pavlovic. Unsupervised visual domain adaptation: A deep max-margin gaussian process approach. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4375–4385, 2019.

[32] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[34] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10285–10295. Computer Vision Foundation / IEEE, 2019.

[35] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, 2020.

[36] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 97–105, Lille, France, 2015.

[37] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1640–1650. Curran Associates, Inc., 2018.

[38] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *2013 IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.

[39] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems 29*, pages 136–144. 2016.

[40] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2208–2217, 2017.

[41] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1041–1048. 2009.

[42] B. Moiseev, A. Konev, A. Chigorin, and A. Konushin. Evaluation of traffic sign recognition methods trained on synthetically generated data. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 576–583. Springer, 2013.

[43] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[44] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1187–1192, 2009.

[45] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, pages 2234–2242, 2019.

[46] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3934–3941, 2018.

[47] I. Redko, A. Habrard, and M. Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753, 2017.

[48] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani. *Advances in Domain Adaptation Theory*. Elsevier, 2019.

[49] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2988–2997. JMLR. org, 2017.

[50] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[51] S. Sankaranarayanan, Y. Balaji, Carlos D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[52] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, pages 99–102, 2015.

[53] O. Sener, H O Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016.

[54] R. Shu, H. Bui, H. Narui, and S. Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.

[55] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[56] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pages 1453–1460, 2011.

[57] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2058–2065. AAAI Press, 2016.

[58] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation, 2019.

[59] I. O. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. *CoRR*, abs/1711.01558, 2018.

[60] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. *CoRR*, 2015.

[61] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[62] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[63] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, second edition, November 1999.

[64] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pages 5385–5394, 2017.

[65] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.

[66] J. Wang, Wenjie Feng, Y. Chen, H. Yu, M. Huang, and Philip S. Yu. Visual domain adaptation with manifold embedded distribution alignment. *Proceedings of the 26th ACM international conference on Multimedia*, 2018.

[67] Q. Wang and T. P. Breckon. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6243–6250, 2020.

[68] Q. Wang, P. Bu, and T. P. Breckon. Unifying unsupervised domain adaptation and zero-shot visual recognition. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[69] X. Wang, L. Li, W. Ye, M. Long, and J. Wang. Transferable attention for domain adaptation. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 5345–5352, 2019.

[70] S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5423–5432. PMLR, 10–15 Jul 2018.

[71] Yujia Xie, Minshuo Chen, Haoming Jiang, Tuo Zhao, and Hongyuan Zha. On scalable and efficient computation of large scale optimal transport. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6882–6892, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[72] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR 2020*, June 2020.

[73] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, pages 3801–3809, 2018.

[74] Y. Zhang, Y. Liu, M. Long, and M. I. Jordan. Bridging theory and algorithm for domain adaptation. *CoRR*, abs/1904.05801, 2019.

[75] Y. Zhang, H. Tang, K. Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5026–5035, 2019.

Supplementary Material for *LAMDA: Label Matching Deep Domain Adaptation with New Theoretical Analysis*

In this supplementary material, we provide complete detail for all proofs presented in our main paper together with the related background so that it can be as self-contained as possible. In the following part, we present the experiment on a synthetic dataset to verify our theory, followed by the experimental settings and datasets for our LAMDA.

1 Related Background

In this section, we present the related background for our paper. We depart with the introduction of pushforward measure followed by the definition of optimal transport and the introduction of a standard machine learning setting.

1.1 Pushforward Measure

Given two probability spaces $(\mathcal{X}, \mathcal{F}, \mu)$ and $(\mathcal{Y}, \mathcal{G})$ where $\mathcal{X}, \mathcal{Y}$ are two sample spaces, $\mathcal{F}, \mathcal{G}$ are two σ-algebras over $\mathcal{X}, \mathcal{Y}$ respectively, and μ is a probability measure, a map $T : \mathcal{X} \to \mathcal{Y}$ is said to be $(\mathcal{Y}, \mathcal{G})$- $(\mathcal{X}, \mathcal{F})$ measurable if for every $A \in \mathcal{G}$, the inverse $T^{-1}(A) \in \mathcal{F}$. The $(\mathcal{Y}, \mathcal{G})$- $(\mathcal{X}, \mathcal{F})$ measurable map T when applied to $(\mathcal{X}, \mathcal{F}, \mu)$ induces a distribution ν over $(\mathcal{Y}, \mathcal{G})$ which is defined as:

$$\nu(A) = \mu\left(T^{-1}(A)\right), \forall A \in \mathcal{G}$$

We also say that the map T transport the probability measure μ to ν and denote as $\nu = T_{\#}\mu$. Furthermore, if μ and ν are two given atomless probability measures over $(\mathcal{X}, \mathcal{F})$ and $(\mathcal{Y}, \mathcal{G})$, there exists a bijection $T : \mathcal{X} \to \mathcal{Y}$ that transports μ to ν. This is known as measurable isomorphism and formally stated in [18] (Chapter 1, Page 19).

Theorem 1. *Given two probability spaces $(\mathcal{X}, \mathcal{F}, \mu)$ and $(\mathcal{Y}, \mathcal{G}, \nu)$ with two atomless probability μ, ν over two Polish spaces $\mathcal{X}$, $\mathcal{Y}$ (i.e., separably complete metric spaces), there exist a bijection $T : \mathcal{X} \to \mathcal{Y}$ that transports μ to ν, i.e., $T_{\#}\mu = \nu$.*

1.2 Optimal Transport

Given two probability measures $(\mathcal{X}, \mu)$ and $(\mathcal{Y}, \nu)$ and a cost function $c(\boldsymbol{x}, \boldsymbol{x}')$, under the conditions stated in Theorems 1.32 and 1.33 [14], two following definitions of Wasserstein (WS) distance are equivalent:

$$\mathrm{WS}_{c,p}(\mu, \nu) = \inf_{T_{\#}\mu=\nu} \mathbb{E}_{\boldsymbol{x}\sim\mu}\left[c(\boldsymbol{x}, T(\boldsymbol{x}))^p\right]^{1/p}$$

$$\mathrm{WS}_{c,p}(\mu, \nu) = \inf_{\pi\in\Gamma(\mu,\nu)} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim\pi}\left[c(\boldsymbol{x}, \boldsymbol{x}')^p\right]^{1/p}$$

where $p > 0$ and $\Gamma(\mu, \nu)$ specifies the set of joint distributions over $\mathcal{X} \times \mathcal{Y}$ which admits μ and ν as marginals.

1.3 Machine Learning Setting and General Loss

According to [17], a standard machine learning system consists of three components: the generator, the supervisor, and the hypothesis class.

Generator The generator is the mechanism to generate data examples $\boldsymbol{x} \in \mathbb{R}^d$ and is mathematically formulated by an existed but unknown distribution $p(\boldsymbol{x})$.

Supervisor The supervisor is the mechanism to assign labels y (e.g., $y \in \{1, 2, \ldots, M\}$ for the classification problem and $y \in \mathbb{R}$ for the regression problem) to a data example $\boldsymbol{x}$ and is mathematically formulated as a conditional distribution $p(y \mid \boldsymbol{x})$.

Hypothesis class This specifies the hypothesis set $\mathcal{H} = \{h_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$ parameterized by $\boldsymbol{\theta}$ which is used to predict label for the data examples $\boldsymbol{x}$.

Given a loss function $l(x, y; \boldsymbol{\theta}) = \ell(y, h_{\boldsymbol{\theta}}(\boldsymbol{x}))$ where $\ell : \mathbb{R}^2 \to \mathbb{R}$ and $\ell(y, y')$ specifies the loss suffered if predicting the data example $\boldsymbol{x}$ with the label y' while its true label is y, the general loss of the hypothesis $h_{\boldsymbol{\theta}}$ is defined as the expected loss caused by $h_{\boldsymbol{\theta}}$:

$$R(\boldsymbol{\theta}) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell(y, h_{\boldsymbol{\theta}}(\boldsymbol{x}))\right] = \int \ell(y, h_{\boldsymbol{\theta}}(\boldsymbol{x}))\, p(\boldsymbol{x}, y)\, d\boldsymbol{x}dy$$

The optimal parameter $\boldsymbol{\theta}^* \in \Theta$ is sought by minimizing the general loss as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}}\, R(\boldsymbol{\theta})$$

2 Theoretical Results

2.1 Gap between target and source domains

In this section, we investigate the variance $\Delta R(h^s, h^t)$ between the expected loss in target domain $R^t(h^t)$ and the expected loss in source domain $R^s(h^s)$ where $h^t = h^s \circ T$. We embark on with the following simple yet key proposition indicating the connection between $R^t(h^t)$ and $R^{\#}(h^s)$.

Proposition 2. *As long as $h^t = h^s \circ T$, we have $R^t(h^t) = R^{\#}(h^s)$.*

Proof. The proof of the proposition is directly from the definitions of h^t, h^s, and expected losses. In particular, we find that

$$R^{\#}(h^s) = \int \ell(y, h^s(\boldsymbol{x}))\, p^{\#}(y \mid \boldsymbol{x})\, p^{\#}(\boldsymbol{x})\, d\boldsymbol{x}dy = \mathbb{E}_{\mathbb{P}^{\#}}\left[\int \ell(y, h^s(\boldsymbol{x}))\, p^{\#}(y \mid \boldsymbol{x})\, dy\right].$$

Recall that, T transports the target distribution $\mathbb{P}^t$ to the source distribution $\mathbb{P}^{\#}$, we achieve that

$$\begin{aligned} R^{\#}(h^s) &= \mathbb{E}_{\mathbb{P}^t}\left[\int \ell(y, h^s(T(\boldsymbol{x})))\, p^{\#}(y \mid T(\boldsymbol{x}))\, dy\right] \\ &= \mathbb{E}_{\mathbb{P}^t}\left[\int \ell\left(y, h^t(\boldsymbol{x})\right) p^t(y \mid \boldsymbol{x})\, dy\right] = R^t\left(h^t\right), \end{aligned}$$

where the second equality is due to the connection $h^t = h^s \circ T$. As a consequence, we reach the conclusion of the proposition. □

Theorem 3. *Assume that Assumption (A.1) holds. Then, for any hypothesis $h^s \in \mathcal{H}^s$, the following inequality holds:*

$$\Delta R\left(h^s, h^t\right) \leq M\left(WS_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) + \mathbb{E}_{\mathbb{P}^t}\left[\left\|\Delta p\left(y \mid \boldsymbol{x}\right)\right\|_1\right]\right),$$

where $\Delta p\left(y \mid \boldsymbol{x}\right)$ is given by $\Delta p\left(y \mid \boldsymbol{x}\right) := \left\|p^t\left(y \mid \boldsymbol{x}\right) - p^s\left(y \mid T\left(\boldsymbol{x}\right)\right)\right\|_1$, and $WS_{c_{0/1}}\left(\cdot, \cdot\right)$ is the Wasserstein distance with respect to the cost function $c_{0/1}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathbf{1}_{\boldsymbol{x} \neq \boldsymbol{x}'}$, which returns 1 if $\boldsymbol{x} \neq \boldsymbol{x}'$ and 0 otherwise.

Proof. Invoking the result from Proposition 2 and the basic triangle inequality, we obtain that

$$\begin{aligned}
\Delta R\left(h^s, h^t\right) =& \left|R^t\left(h^t\right) - R^s\left(h^s\right)\right| = \left|R^{\#}\left(h^s\right) - R^s\left(h^s\right)\right| \\
=& \left|R^{\#}\left(h^s\right) - R^{\#,s}\left(h^s\right) + R^{\#,s}\left(h^s\right) - R^s\left(h^s\right)\right| \\
\leq& \left|R^{\#}\left(h^s\right) - R^{\#,s}\left(h^s\right)\right| + \left|R^{\#,s}\left(h^s\right) - R^s\left(h^s\right)\right|.
\end{aligned}$$

To achieve the conclusion of the theorem, it is sufficient to upper bound the two terms $\left|R^{\#}\left(h^s\right) - R^{\#,s}\left(h^s\right)\right|$ and $\left|R^{\#,s}\left(h^s\right) - R^s\left(h^s\right)\right|$. For the first term, according the definition of expected losses, we find that

$$\begin{aligned}
\left|R^{\#}\left(h^s\right) - R^{\#,s}\left(h^s\right)\right| &= \left|\int \ell\left(h^s\left(\boldsymbol{x}\right), y\right)\left(p^{\#}\left(y \mid \boldsymbol{x}\right) - p^s\left(y \mid \boldsymbol{x}\right)\right) p^{\#}\left(\boldsymbol{x}\right) d\boldsymbol{x} dy\right| \\
&\leq \int \ell\left(h^s\left(\boldsymbol{x}\right), y\right)\left|p^{\#}\left(y \mid \boldsymbol{x}\right) - p^s\left(y \mid \boldsymbol{x}\right)\right| p^{\#}\left(\boldsymbol{x}\right) d\boldsymbol{x} dy \\
&\leq M \mathbb{E}_{\mathbb{P}^{\#}}\left[\left\|p^{\#}\left(y \mid \boldsymbol{x}\right) - p^s\left(y \mid \boldsymbol{x}\right)\right\|_1\right] \\
&= M \mathbb{E}_{\mathbb{P}^t}\left[\left\|p^{\#}\left(y \mid T\left(\boldsymbol{x}\right)\right) - p^s\left(y \mid T\left(\boldsymbol{x}\right)\right)\right\|_1\right] \qquad (1) \\
&= M \mathbb{E}_{\mathbb{P}^t}\left[\left\|p^t\left(y \mid \boldsymbol{x}\right) - p^s\left(y \mid T\left(\boldsymbol{x}\right)\right)\right\|_1\right], \qquad (2)
\end{aligned}$$

where the last equality is from the fact that $T_{\#}\mathbb{P}^t = \mathbb{P}^{\#}$.

For the second term, similar argument as the above argument leads to

$$\begin{aligned}
\left|R^{\#,s}\left(h^s\right) - R^s\left(h^s\right)\right| &= \left|\int \ell\left(h^s\left(\boldsymbol{x}\right), y\right) p^s\left(y \mid \boldsymbol{x}\right)\left[p^{\#}\left(\boldsymbol{x}\right) - p^s\left(\boldsymbol{x}\right)\right] d\boldsymbol{x} dy\right| \\
&\leq \int \ell\left(h^s\left(\boldsymbol{x}\right), y\right) p^s\left(y \mid \boldsymbol{x}\right)\left|p^{\#}\left(\boldsymbol{x}\right) - p^s\left(\boldsymbol{x}\right)\right| d\boldsymbol{x} dy \\
&\leq M \int p^s\left(y \mid \boldsymbol{x}\right)\left|p^{\#}\left(\boldsymbol{x}\right) - p^s\left(\boldsymbol{x}\right)\right| d\boldsymbol{x} dy \\
&= M \int \left|p^{\#}\left(\boldsymbol{x}\right) - p^s\left(\boldsymbol{x}\right)\right| d\boldsymbol{x} = M \mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right), \qquad (3)
\end{aligned}$$

where the second equality is due to Fubini's theorem with the switch of integrals and the final equality is from the fact that cost matrix $c_{0/1}$ is given by $c_{0/1}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathbf{1}_{\boldsymbol{x} \neq \boldsymbol{x}'}$, which returns 1 if $\boldsymbol{x} \neq \boldsymbol{x}'$ and 0 otherwise (for the second equality, please refer to [7], Page 7 and the coupling characterization of total variance distance).

Combining the results from (1) and (3), we arrive at the bound that

$$\begin{aligned}
\Delta R\left(h^s, h^t\right) &\leq M\left(\mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) + \mathbb{E}_{\mathbb{P}^t}\left[\left\|p^t\left(y \mid \boldsymbol{x}\right) - p^s\left(y \mid T\left(\boldsymbol{x}\right)\right)\right\|_1\right]\right) \\
&= M\left(\mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) + \mathbb{E}_{\mathbb{P}^t}\left[\left\|\Delta p\left(y \mid \boldsymbol{x}\right)\right\|_1\right]\right).
\end{aligned}$$

As a consequence, we reach the conclusion of the theorem. □

Remark 4. If the following assumptions hold:

(i) The transformation mapping $T(\boldsymbol{x}) = \boldsymbol{x}$, i.e., we use the same hypothesis set for both the source and target domains,

(ii) The loss $\ell(y, h(\boldsymbol{x})) = \frac{1}{2}|y - h(\boldsymbol{x})|$ where we restrict to consider hypothesis $h : \mathcal{X} \to \{-1, 1\}$,

then we recover the theoretical result obtained in [2].

Remark 5. When $\mathrm{WS}_{c_{0/1}}\left(\mathbb{P}^s, \mathbb{P}^{\#}\right) = 0$, i.e., $T_{\#}\mathbb{P}^t = \mathbb{P}^s$, and there is a harmony between two supervisors of source and target domain, i.e., $p^t(y \mid \boldsymbol{x}) = p^s(y \mid T(\boldsymbol{x}))$), Theorem 3 suggests that we can perfectly do transfer learning without loss of performance. This fact is summarized in the following corollary.

Corollary 6. *Assume that $T_{\#}\mathbb{P}^t = \mathbb{P}^s$ and the source and target supervisor distributions are harmonic in the sense that $p^s(y \mid T(\boldsymbol{x})) = p^t(y \mid \boldsymbol{x})$ for $\boldsymbol{x} \sim \mathbb{P}_t$. Then, we can do a perfect transfer learning between the source and target domains.*

Proof. For any $h^s \in \mathcal{H}^s$, denote $h^t = h^s \circ T$, we have

$$\begin{aligned} R^s(h^s) = &\mathbb{E}_{\mathbb{P}^s}\left[\int \ell(y, h^s(\boldsymbol{x}))\, p^s(y \mid \boldsymbol{x})\, dy\right] \\ =&\mathbb{E}_{\mathbb{P}^t}\left[\int \ell(y, h^s(T(\boldsymbol{x})))\, p^s(y \mid T(\boldsymbol{x}))\, dy\right] \\ =&\mathbb{E}_{\mathbb{P}^t}\left[\int \ell\left(y, h^t(\boldsymbol{x})\right) p^t(y \mid \boldsymbol{x})\, dy\right] = R^t\left(h^t\right), \end{aligned}$$

, where the second equality is from the fact T transport $\mathbb{P}^t$ to $\mathbb{P}^s$. □

2.2 Optimization via Wasserstein metric

Let $\mathcal{Z}$ be an intermediate space (i.e., the joint space $\mathcal{Z} = \mathbb{R}^m$). We consider the composite mappings H: $H(\boldsymbol{x}) = H^2\left(H^1(\boldsymbol{x})\right)$ where H^1 is an injective mapping from the target domain $\mathcal{X}^t$ to the joint space $\mathcal{Z}$ and H^2 maps from the joint space $\mathcal{Z}$ to the source domain $\mathcal{X}^s$ (note that if $\mathcal{Z} = \mathcal{X}^s$ then $H^2 = id$ is the identity function). Based on that structure on H, we consider the following optimization problem:

$$\min_{H^1, H^2} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right). \tag{4}$$

In the following theorem, we demonstrate that the above optimization problem can be equivalently transformed into another form involving the joint space (see Figure 1 for an illustration of that theorem).

Theorem 7. *The optimization problem* (4) *is equivalent to the following optimization problem:*

$$\min_{H^1, H^2} \min_{G^1 : H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1(\boldsymbol{x})\right)\right)^p\right]^{1/p}, \tag{5}$$

where G^1 is an injective map from the source domain $\mathcal{X}^s$ to the joint space $\mathcal{Z}$.

Proof. From the definition of Wasserstein metric, we obtain that

$$\mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right) = \min_{L : L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c(\boldsymbol{x}, L(\boldsymbol{x}))^p\right]^{1/p}.$$

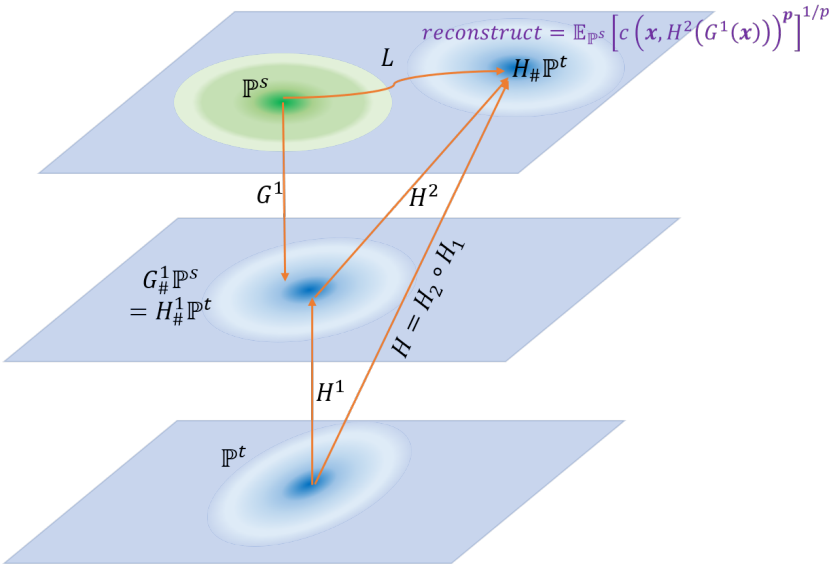

Figure 1: The mapping $H = H^2 \circ H^1$ maps from the target to source domains. We minimize $D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)$ to close the discrepancy gap of the source and target domains in the joint space.

Therefore, we can rewrite the optimization problem (5) as follows:

$$\min_{H^{1:2}} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right) = \min_{H^{1:2}} \min_{L:L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p}.$$

We first prove that

$$\min_{H^{1:2}} \min_{G^1:H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} \geq \min_{H^{1:2}} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right).$$

Given the mappings $H^{1:2}$, for any mapping G^1 satisfying the equation $H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s$, we let $H' = H^2 \circ G^1$. Then, we arrive at $H'_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t$. Hence, we find that

$$\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H'\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} \geq \min_{L:L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p}.$$

The above inequality directly leads to

$$\min_{G^1:H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} \geq \min_{L:L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p}.$$

As a consequence, we achieve the following inequality

$$\begin{aligned}\min_{H^{1:2}} \min_{G^1:H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} &\geq \min_{H^{1:2}} \min_{L:L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} \\ &= \min_{H^{1:2}} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right).\end{aligned}$$

We now prove that

$$\min_{H^{1:2}} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right) \geq \min_{H^{1:2}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.$$

Given the mapping H^1, we consider the distribution $\mathbb{Q}$ over the source domain such that there exists a map H^2 for which $H^2_{\#}\left(H^1_{\#}\mathbb{P}^t\right) = \mathbb{Q}$. For any mapping L satisfying the equation $L_{\#}\mathbb{P}^s = \mathbb{Q}$, we can find mappings U, V such that $U_{\#}\mathbb{P}^s = H^1_{\#}\mathbb{P}^t$ and $L = V \circ U$. To this end, there exists a bijective mapping V satisfying $V_{\#}\left(H^1_{\#}\mathbb{P}^t\right) = \mathbb{Q}$ since these two distributions are atomless (see Theorem 1). Additionally, we can set $U = V^{-1} \circ L$. It is obvious that $U_{\#}\mathbb{P}^s = H^1_{\#}\mathbb{P}^t$ and $L = V \circ U$ from the definitions of U and V. Therefore, we have that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} =& \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, V\left(U\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} \\ \geq& \min_{H^2: H^2_{\#}\left(H^1_{\#}\mathbb{P}^t\right)=\mathbb{Q}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.\end{aligned}$$

Invoking the above equality, we find that

$$\min_{L: L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} \geq \min_{H^2: H^2_{\#}\left(H^1_{\#}\mathbb{P}^t\right)=\mathbb{Q}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.$$

With that inequality, we directly achieve the following inequality

$$\begin{aligned}&\min_{\mathbb{Q}} \min_{L: L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} \\ \geq& \min_{\mathbb{Q}} \min_{H^2: H^2_{\#}\left(H^1_{\#}\mathbb{P}^t\right)=\mathbb{Q}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.\end{aligned}$$

$$\min_{H^2} \min_{L: L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} \geq \min_{H^2} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.$$

Note that from the definitions of $\mathbb{Q}$ and H^2, it is obvious that

$$\min_{\mathbb{Q}} \min_{L: L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} = \min_{H^2} \min_{L: L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p}.$$

$$\begin{aligned}&\min_{H^2} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} \\ =& \min_{\mathbb{Q}} \min_{H^2: H^2_{\#}\left(H^1_{\#}\mathbb{P}^t\right)=\mathbb{Q}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.\end{aligned}$$

By varying the mapping H^1 in both sides of the above inequality, we arrive at the following inequality

$$\min_{H^1, H^2} \min_{L: L_{\#}\mathbb{P}^s = H_{\#}\mathbb{P}^t} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, L\left(\boldsymbol{x}\right)\right)^p\right]^{1/p} \geq \min_{H^1, H^2} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.$$

Hence, we obtain that

$$\min_{H^{1:2}} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right) \geq \min_{H^{1:2}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.$$

Finally, we reach the conclusion as:

$$\min_{H^{1:2}} \mathrm{WS}_{c,p}\left(\left(H^2 \circ H^1\right)_{\#}\mathbb{P}^t, \mathbb{P}^s\right) = \min_{H^{1:2}} \min_{G^1: H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p}.$$

□

It is interesting to interpret G^1 and H^1 as two generators that map the source and target domains to the common joint space $\mathcal{Z}$ respectively. The constraint $H^1_{\#}\mathbb{P}^t = G^1_{\#}\mathbb{P}^s$ further indicates that the gap between the source and target distributions is closed in the joint space via two generators G^1 and H^1. Furthermore, H^2 maps from the joint space to the source domain and aims to reconstruct G^1. Similar to [16], we do relaxation and arrive at the optimization problem:

$$\min_{H^1, H^2, G^1}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[c\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)^p\right]^{1/p} + \alpha D\left(G^1_{\#}\mathbb{P}^s, H^1_{\#}\mathbb{P}^t\right)\right), \tag{6}$$

where $D\left(\cdot,\cdot\right)$ specifies a divergence between two distributions over the joint space and $\alpha > 0$.

It is obvious that when the trade-off parameter α approaches $+\infty$, the solution of the relaxation problem in Eq. (6) approaches the optimal solution in Eq. (5).

Since G^1 and H^1 are two maps from the source and target domains to the joint space, we can further define two source and target supervisor distributions on the joint space as $p^{\#,s}\left(y \mid G^1\left(\boldsymbol{x}\right)\right) = p^s\left(y \mid \boldsymbol{x}\right)$ and $p^{\#,t}\left(y \mid H^1\left(\boldsymbol{x}\right)\right) = p^t\left(y \mid \boldsymbol{x}\right)$. With respect to the joint space, the second term of the upper bound in Theorem 3 can be rewritten as in the following corollary.

Corollary 8. *The second term of the upper bound in Theorem 3 can be rewritten as*

$$\mathbb{E}_{\mathbb{P}^t}\left[\left\|p^{\#,s}\left(y \mid G^1\left(H^2\left(H^1\left(\boldsymbol{x}\right)\right)\right)\right) - p^{\#,t}\left(y \mid H^1\left(\boldsymbol{x}\right)\right)\right\|_1\right]. \tag{7}$$

Proof. The proof is trivial from the definitions of $p^{\#,s}\left(y \mid G^1\left(\boldsymbol{x}\right)\right) = p^s\left(y \mid \boldsymbol{x}\right)$ and $p^{\#,t}\left(y \mid H^1\left(\boldsymbol{x}\right)\right) = p^t\left(y \mid \boldsymbol{x}\right)$. □

3 Experiments

3.1 Experiment on Synthetic Data

3.1.1 Synthetic Dataset for the Source and Target Domains

We generate two synthetic labeled datasets for the source and target domains. We generate the $10,000$ data examples of the source dataset from the mixture of two Gaussian distributions: $p^s\left(\boldsymbol{x}\right) = \pi^s_1\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}^s_1, \Sigma^s_1\right) + \pi^s_2\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}^s_2, \Sigma^s_2\right)$ where $\pi^s_1 = \pi^s_2 = \frac{1}{2}$, $\boldsymbol{\mu}^s_1 = [1, 1, ..., 1] \in \mathbb{R}^{10}$, $\boldsymbol{\mu}^s_2 = [2, 2, ..., 2] \in \mathbb{R}^{10}$ and $\Sigma^s_1 = \Sigma^s_2 = \mathbb{I}_{10}$. Similarly, we generate the another $10,000$ data examples of the target dataset from the mixture of two Gaussian distributions: $p^t\left(\boldsymbol{x}\right) = \pi^t_1\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}^t_1, \Sigma^t_1\right) + \pi^t_2\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}^t_2, \Sigma^t_2\right)$ where $\pi^t_1 = \frac{1}{3}$, $\pi^t_2 = \frac{2}{3}$, $\boldsymbol{\mu}^t_1 = [4, 4, ..., 4] \in \mathbb{R}^{10}$, $\boldsymbol{\mu}^t_2 = [5, 5, ..., 5] \in \mathbb{R}^{10}$ and $\Sigma^t_1 = \Sigma^t_2 = \mathbb{I}_{10}$. For each data example in the source and target domains, we assign label $y = 0$ if this data example is generated from the first Gauss and $y = 1$ if this data example is generated from the second Gauss using Bayes' s rule.

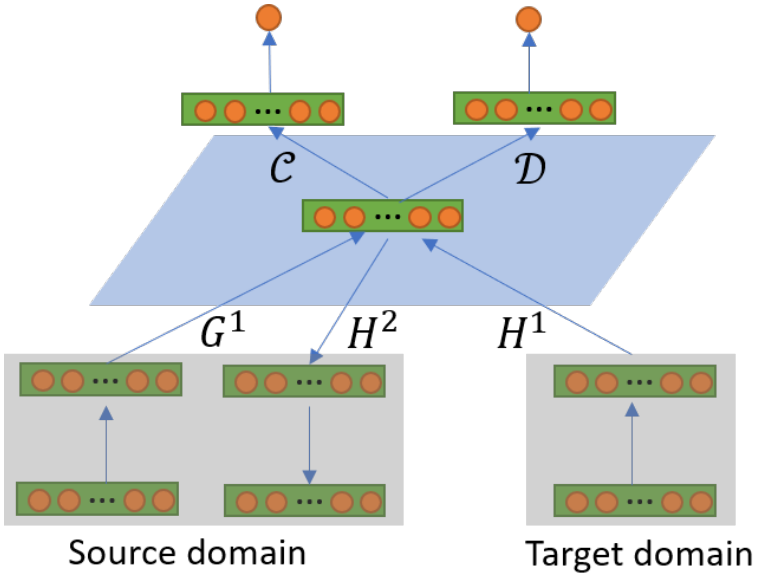

Figure 2: Architecture of networks for deep domain adaptation.

3.1.2 Deep Domain Adaptation on the Synthetic Dataset

Figure 2 shows the architectures of networks used in our experiments on the synthetic datasets. Two generators G^1, H^1 with the same architectures $(10 \to 5\,(\text{ReLu}) \to 5\,(\text{ReLu}))$ map the source and target data to the intermediate joint layer. Note that different from other works in deep domain adaptation, we did not tie G^1 and H^1. The network H^2 with the architecture $(10 \to 5\,(\text{ReLu}) \to 5\,(\text{ReLu}))$ maps from the intermediate joint layer to the source and target domains respectively. The purpose of G^2, H^2 is to reconstruct H^1, G^1 respectively. To break the gap between the source and target domains in the joint layer, we employ GAN principle [8, 6] wherein we invoke a discriminator network $\mathcal{D}$ $(5 \to 5\,(\text{ReLu}) \to 1\,(\text{sigmoid}))$ to discriminate the source and target data examples in the joint space. The classifier network $\mathcal{C}$ $(5 \to 5\,(\text{ReLu}) \to 1\,(\text{sigmoid}))$ is employed to classify the labeled source data examples. To approximate the 0/1 cost function, we use the modified sigmoid function [13]: $c_\gamma\left(\boldsymbol{x}, \boldsymbol{x}'\right) = 2/\left[1 + \exp\left\{-\gamma \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2\right\}\right] - 1$ with $\gamma = 100$. It can be seen that when $\gamma \to +\infty$, the cost function c_γ approaches the 0/1 cost function. More specifically, we need to update $G^{1:2}, H^{1:2}, \mathcal{C}$, and $\mathcal{D}$ as follows:

$$\left(G^1, H^{1:2}, \mathcal{C}\right) = \underset{G^1, H^{1:2}, \mathcal{C}}{\operatorname{argmin}}\, \mathcal{I}\left(G^1, H^{1:2}, \mathcal{C}\right) \text{ and } \mathcal{D} = \underset{\mathcal{D}}{\operatorname{argmax}}\, \mathcal{J}\left(\mathcal{D}\right),$$

where α is set to 0.1 and we have defined

$$\begin{aligned}
\mathcal{I}\left(G^1, H^{1:2}, \mathcal{C}\right) = & \\
& + \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^s}\left[c_\gamma\left(\boldsymbol{x}, H^2\left(G^1\left(\boldsymbol{x}\right)\right)\right)\right] + \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}^s}\left[\ell\left(y, \mathcal{C}\left(G^1\left(\boldsymbol{x}\right)\right)\right)\right] \\
& + \alpha\left[\mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^s}\left[\log\left(\mathcal{D}\left(G^1\left(\boldsymbol{x}\right)\right)\right)\right] + \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^t}\left[\log\left(1 - \mathcal{D}\left(H^1\left(\boldsymbol{x}\right)\right)\right)\right]\right] \\
\mathcal{J}\left(\mathcal{D}\right) = & \, \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^s}\left[\log\left(\mathcal{D}\left(G^1\left(\boldsymbol{x}\right)\right)\right)\right] + \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^t}\left[\log\left(1 - \mathcal{D}\left(H^1\left(\boldsymbol{x}\right)\right)\right)\right].
\end{aligned}$$

Based on the classifier $\mathcal{C}$ on the joint space, we can identify the corresponding hypotheses on the source and target domains as: $h^s\left(\boldsymbol{x}\right) = \mathcal{C}\left(G^1\left(\boldsymbol{x}\right)\right)$ and $h^t\left(\boldsymbol{x}\right) = \mathcal{C}\left(H^1\left(\boldsymbol{x}\right)\right)$.

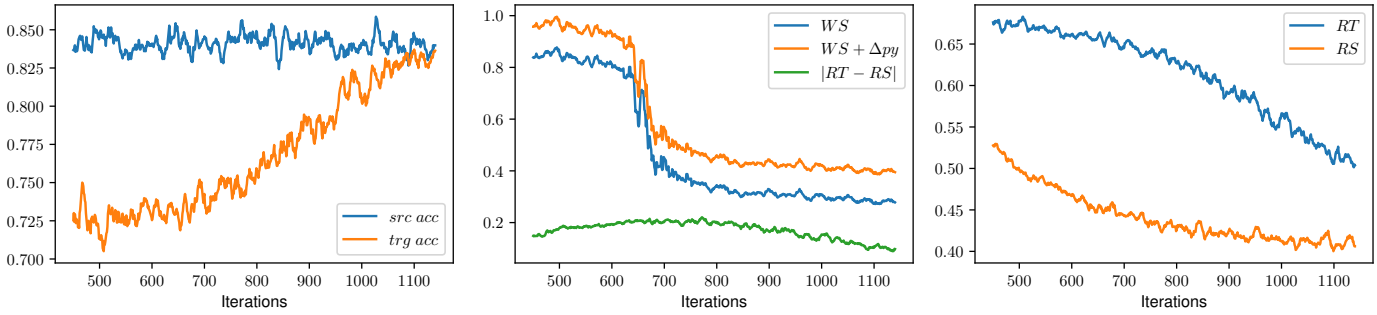

Figure 3: Left: the accuracies on the source and target datasets. Middle: the plots of three terms in Theorem 3. Right: the plot of empirical losses on the source and target datasets.

3.1.3 Verification of Our Theory for Unsupervised Domain Adaptation

In this experiment, we assume that none of data example in the target domain has label. We measure three terms, namely $\left|R\left(h^t\right)-R\left(h^s\right)\right|$, $WS\left(\mathbb{P}^s,\mathbb{P}^{\#}\right)$, and $\mathbb{E}_{\mathbb{P}^t}\left[\left\|\Delta p\left(y\mid\boldsymbol{x}\right)\right\|_1\right]$ ($M=1$ since we are using the logistic loss) as defined in Theorem 3 across the training progress. Actually, we approximate $R\left(h^t\right), R\left(h^s\right)$ using the corresponding empirical losses. As shown in Figure 3 (middle), the green plot is always above the blue plot and this empirically confirms the inequality in Theorem 3. Furthermore, the fact that three terms consistently decrease across the training progress indicates an improvement when $\mathbb{P}^{\#}$ is shifting toward $\mathbb{P}^s$. This improvement is also reflected in Figure 3 (left and right) wherein the target accuracy and empirical loss gradually increase and decrease accordingly.

3.2 Experimental Setting for our LAMDA

3.2.1 The Objective Function of LAMDA

Note that we set $G^1=H^1=G$ and $H^2=H$. Let us further denote:

$$\mathcal{L}_C(\mathcal{D}^s)=\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^s}\left[\ell\left(y,\mathcal{C}\left(G\left(\boldsymbol{x}\right)\right)\right)\right],$$

$$\begin{aligned}\mathcal{L}_G=&-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[\log\, d_{M+1}\left(G(\boldsymbol{x})\right)\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^t}\left[\log\left(1-d_{M+1}\left(G\left(\boldsymbol{x}\right)\right)\right)\right]\\&-\gamma\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^t}\left[\sum_{m=1}^{M}\mathcal{C}_m\left(\boldsymbol{x}\right)\log d_m\left(G\left(\boldsymbol{x}\right)\right)\right],\end{aligned}$$

$$\begin{aligned}\mathcal{L}_d=&-\sum_{m=1}^{M}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^s\wedge y=m}\left[\log d_m\left(G\left(\boldsymbol{x}\right)\right)\right]-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^t}\left[\log d_{M+1}\left(G\left(\boldsymbol{x}\right)\right)\right]\\&-\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}^s}\left[\log\left(1-d_{M+1}\left(G\left(\boldsymbol{x}\right)\right)\right)\right].\end{aligned}$$

$$\mathcal{L}_R=\mathbb{E}_{\mathbb{P}^s}\left[\left\|H\left(G\left(\boldsymbol{x}\right)\right)-\boldsymbol{x}\right\|^2\right]$$

To update G, C, H, we solve:

$$\min\left(\mathcal{L}_C(\mathcal{D}^s)+\alpha\mathcal{L}_G+\beta\mathcal{L}_R\right).$$

To update d, we solve:

$$\min\,\mathcal{L}_d.$$

3.2.2 Experimental Datasets

Digit/scene datasets

We split datasets into training sets and test sets as described in Table 1. We resize the resolution of each sample in digit (MNIST, MNIST-M, SVHN, DIGITS), and natural image datasets (CIFAR, STL) to 32×32, and normalize the value of each pixel to the range of $[-1, 1]$.

MNIST[1]. The dataset is commonly used in domain adaptation literature. To adapt from MNIST to MNIST-M or SVHN, the MNIST images are replicated from single greyscale channel to obtain digit images which has three channels.

MNIST-M[2]. Following by the implementation in [6], we generate the MNIST-M images by replacing the black background of MNIST images by the color ones. We eventually obtain the same number of training and test samples as the MNIST dataset.

SVHN[3]. The dataset consists of images obtained by detecting house numbers from Google Street View images. This dataset is a benmark for recognizing digits and numbers in real-world images.

DIGITS[4]. There are roughly 500,000 images are generated using various data augmentation schemes, i.e., varying the text, positioning, orientation, background, stroke color, and the amount of blur.

CIFAR[5]. The CIFAR-10 [10] dataset includes 50,000 training images and 10,000 test images. However, to adapt with STL dataset, we base on [4] to remove one non-overlapping class ("frog"). The numbers of training examples and test examples therefore are reduced to 45,000 and 9,000 respectively.

STL[6]. Similar to CIFAR-10, we remove class named "monkey" to obtain a 9-class classification problem. Also, STL-10 images are down-scaled to a resolution of 96×96 to 32×32.

Office-31 and Office-Home datasets

The resolutions of images in Office-Caltech10 and Office-31 are resized to 224×224 and 227×227 for finetuning pre-trained the pre-trained VGG-16 [15] and ResNet-50 [9], respectively.

Table 1: Data preparation for our model.

Dataset	#train	#test	#classes	Category	Resolution
MNIST [11]	60,000	10,000	10	Digits	28×28
MNIST-M [5]	60,000	10,000	10	Digits	28×28
SVHN [12]	73,257	26,032	10	Digits	32×32
DIGITS [5]	479,400	9,553	10	Digits	32×32
CIFAR [10]	45,000	9,000	9	Natural images	32×32
STL [3]	4,500	7,200	9	Natural images	96×96

3.2.3 Network Architectures

We use small and large network architecture for specific datasets, which are described in Table 2 and 3. Noticeably, batch normalization layers are applied on the top of convolutional layers (6 for the generator and 3 for the classifier) to prevent the overfitting. We note that the architecture of H is same as that of generator G, but the Conv2D layers are replaced by

[1] http://yann.lecun.com/exdb/mnist
[2] http://yaroslav.ganin.net
[3] http://ufldl.stanford.edu/housenumbers
[4] http://yaroslav.ganin.net
[5] http://www.cs.toronto.edu/~kriz/cifar.html
[6] http://ai.stanford.edu/~acoates/stl10

Table 2: Small and large network architecture of LAMDA. We use the small network for digit the large network for natural scene image datasets. The parameter a for Leaky ReLU (lReLU) activation function is set to 0.1.

Architecture	Small Network	Large Network
Input size	$32 \times 32 \times 3$	$32 \times 32 \times 3$
Generator	instance normalization	instance normalization
	3×3 conv. 64 lReLU	3×3 conv. 96 lReLU
	3×3 conv. 64 lReLU	3×3 conv. 96 lReLU
	3×3 conv. 64 lReLU	3×3 conv. 96 lReLU
	2×2 max-pool, stride 2	2×2 max-pool, stride 2
	dropout, $p = 0.5$	dropout, $p = 0.5$
	Gaussian noise, $\sigma = 1$	Gaussian noise, $\sigma = 1$
	3×3 conv. 64 lReLU	3×3 conv. 192 lReLU
	3×3 conv. 64 lReLU	3×3 conv. 192 lReLU
	3×3 conv. 64 lReLU	3×3 conv. 192 lReLU
	2×2 max-pool, stride 2	3×3 max-pool, stride 2
	dropout, $p = 0.5$	dropout, $p = 0.5$
	Gaussian noise, $\sigma = 1$	Gaussian noise, $\sigma = 1$
Classifier	3×3 conv. 64 lReLU	3×3 conv. 192 lReLU
	3×3 conv. 64 lReLU	3×3 conv. 192 lReLU
	3×3 conv. 64 lReLU	3×3 conv. 192 lReLU
	global average pool	global average pool
	#classes dense, softmax	*#classes* dense, softmax

Conv2D Transpose layers and max-pooling layers are replaced by UpSampling layers. For Office-31 and Office-Home, we removed the dense layers of the pretrained models and replaced by one dense layer.

Table 3: The architecture of discriminator.

3×3 conv. 64 lReLU
3×3 conv. 64 lReLU
3×3 conv. 64 lReLU
global average pool
#classes+1 dense, softmax

3.2.4 Hyperparameter setting

We apply Adam Optimizer ($\beta_1 = 0.5$,$\beta_2 = 0.999$) with the learning rate 0.001. All experiements was trained for 80,000 iterations, and the batch size for each dataset is set to 128. We set β is searched in the grid $\{0, 0.05\}$, α is searched in the grid $\{0.1, 0.5\}$, and γ is searched in$\{0.1, 0.5\}$. We implement our LAMDA in Python (version 3.5) using Tensorflow (version 1.9.0) [1] and run our experiments on a computer with a CPU named Intel Xeon Processor E5-1660 which has 8 cores at 3.0 GHz and 128 GB of RAM, and a GPU called NVIDIA GeForce GTX Titan X with 12 GB memory.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, May 2010.

[3] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[4] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

[5] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation, 2014.

[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189, 2015.

[7] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *INTERNAT. STATIST. REV.*, pages 419–435, 2002.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[13] T. Nguyen and S. Sanner. Algorithms for direct 0–1 loss optimization in binary classification. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1085–1093, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[14] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, pages 99–102, 2015.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[16] I. O. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. *CoRR*, abs/1711.01558, 2018.

[17] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, second edition, November 1999.

[18] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.