

# CONTINUOUS TRANSFER LEARNING

## ABSTRACT

Transfer learning has been successfully applied across many high-impact applications. However, most existing work focuses on the static transfer learning setting, and very little is devoted to modeling the time evolving target domain, such as the online reviews for movies. To bridge this gap, in this paper, we focus on the continuous transfer learning setting with a time evolving target domain. One major challenge associated with continuous transfer learning is the time evolving relatedness of the source domain and the current target domain as the target domain evolves over time. To address this challenge, we first derive a generic generalization error bound on the current target domain with flexible domain discrepancy measures. Furthermore, a novel label-informed  $\mathcal{C}$ -divergence is proposed to measure the shift of joint data distributions (over input features and output labels) across domains. It could be utilized to instantiate a tighter error upper bound in the continuous transfer learning setting, thus motivating us to develop an adversarial Variational Auto-encoder algorithm named CONTE by minimizing the  $\mathcal{C}$ -divergence based error upper bound. Extensive experiments on various data sets demonstrate the effectiveness of our CONTE algorithm.

## 1 INTRODUCTION

Transfer learning has achieved significant success across multiple high-impact application domains (Pan & Yang, 2009). Compared to conventional machine learning methods assuming both training and test data have the same data distribution, transfer learning allows us to learn the target domain with limited label information by leveraging a related source domain with abundant label information (Ying et al., 2018). However, in many real applications, the target domain is constantly evolving over time.

For example, the online movie reviews are changing over the years: some famous movies were not well received by the mainstream audience when they were first released, but became famous only years later (e.g., *Citizen Cane*, *Fight Club*, and *The Shawshank Redemption*); whereas the online book reviews typically do not have this type of dynamics. It is challenging to transfer knowledge from the static source domain (e.g., the book reviews) to the time evolving target domain (e.g., the movie reviews). Therefore, in this paper, we study the transfer learning setting with a static source domain and a continuously time evolving target domain (see Figure 1), which has not attracted much attention from the research community and yet is commonly seen across many real applications. The unique challenge for continuous transfer learning lies in the time evolving nature of the task relatedness between the static source domain and the time evolving target domain. Although the change in the target data distribution in consecutive time stamps might be small, over time, the cumulative change in the target domain might even lead to negative transfer (Rosenstein et al., 2005).

Existing theoretical analysis on transfer learning (Ben-David et al., 2010; Mansour et al., 2009) showed that the target error is typically bounded by the source error, the domain discrepancy of marginal data distributions and the difference of labeling functions. However, it has been observed (Zhao et al., 2019; Wu et al., 2019) that marginal feature distribution alignment might not guarantee the minimization of the target error in real world scenarios. This indicates that in the context of continuous transfer learning, marginal feature distribution alignment would lead to the sub-optimal solution (or even negative transfer) with undesirable predictive performance when directly transferring from  $\mathcal{D}_S$  to the target domain  $\mathcal{D}_{T_t}$  at the  $t^{\text{th}}$  time stamp. This paper aims to bridge the gap in terms of both the theoretical analysis and the empirical solutions for the target domain with a time evolving distribution, which lead to a novel continuous transfer learning algorithm as

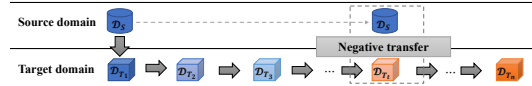


Figure 1: Illustration of continuous transfer learning. It learns a predictive function in  $\mathcal{D}_{T_t}$  using knowledge from both source domain  $\mathcal{D}_S$  and historical target domain  $\mathcal{D}_{T_i} (i = 1, \dots, t-1)$ . Directly transferring from the source domain  $\mathcal{D}_S$  to the target domain  $\mathcal{D}_{T_t}$  might lead to negative transfer with undesirable predictive performance.

well as the characterization of negative transfer. The main contributions of this paper are summarized as follows: (1) We derive a generic error bound for continuous transfer learning setting with flexible domain divergence measures; (2) We propose a label-informed domain discrepancy measure ( $\mathcal{C}$ -divergence) with its empirical estimate, which instantiates a tighter error bound for continuous transfer learning setting; (3) Based on the proposed  $\mathcal{C}$ -divergence, we design a novel adversarial Variational Auto-encoder algorithm (CONTE) for continuous transfer learning; (4) Extensive experimental results on various data sets verify the effectiveness of the proposed CONTE algorithm.

The rest of the paper is organized as follows. Section 2 introduces the notation and our problem definition. We derive a generic error bound for continuous transfer learning setting in Section 3. Then we propose a novel  $\mathcal{C}$ -divergence in Section 4, followed by a instantiated error bound and a novel continuous transfer learning algorithm in Section 5. The experimental results are provided in Section 6. We summarize the related work in Section 7, and conclude the paper in Section 8.

## 2 PRELIMINARIES

In this section, we introduce the notation and problem definition of continuous transfer learning.

### 2.1 NOTATION

We use  $\mathcal{X}$  and  $\mathcal{Y}$  to denote the input space and label space. Let  $\mathcal{D}_S$  and  $\mathcal{D}_T$  denote the source and target domains with data distribution  $p_S(\mathbf{x}, y)$  and  $p_T(\mathbf{x}, y)$  over  $\mathcal{X} \times \mathcal{Y}$ , respectively. Let  $\mathcal{H}$  be a hypothesis class on  $\mathcal{X}$ , where a hypothesis is a function  $h : \mathcal{X} \rightarrow \mathcal{Y}$ . The notation is summarized in Table 3 in the appendices.

### 2.2 PROBLEM DEFINITION

Transfer learning (Pan & Yang, 2009) refers to the knowledge transfer from source domain to target domain such that the prediction performance on the target domain could be significantly improved as compared to learning from the target domain alone. However, in some applications, the target domain is changing over time, hence the time evolving relatedness between the source and target domains. This motivates us to consider the transfer learning setting with the time evolving target domain, which is much less studied as compared to the static transfer learning setting. We formally define the continuous transfer learning problem as follows.

**Definition 2.1.** (*Continuous Transfer Learning*) Given a source domain  $\mathcal{D}_S$  (available at time stamp  $j = 1$ ) and a time evolving target domain  $\{\mathcal{D}_{T_j}\}_{j=1}^n$  with time stamp  $j$ , *continuous transfer learning* aims to improve the prediction function for target domain  $\mathcal{D}_{T_{t+1}}$  using the knowledge from source domain  $\mathcal{D}_S$  and the historical target domain  $\mathcal{D}_{T_j}$  ( $j = 1, \dots, t$ ).

Notice that the source domain  $\mathcal{D}_S$  can be considered a special initial domain for the time-evolving target domain. Therefore, for notation simplicity, we will use  $\mathcal{D}_{T_0}$  to represent the source domain in this paper. It assumes that there are  $m_{T_0}$  labeled source examples drawn independently from a source domain  $\mathcal{D}_{T_0}$  and  $m_{T_j}$  labeled target examples drawn independently from a target domain  $\mathcal{D}_{T_j}$  at time stamp  $j$ .

## 3 A GENERIC ERROR BOUND

Given a static source domain and a time evolving target domain, continuous transfer learning aims to improve the target predictive function over  $\mathcal{D}_{T_{t+1}}$  using the source domain and historical target domain. We begin by considering the binary classification setting, i.e.,  $\mathcal{Y} = \{0, 1\}$ . The source error of a hypothesis  $h$  can be defined as follows:  $\epsilon_{T_0}(h) = \mathbb{E}_{(\mathbf{x}, y) \sim p_{T_0}(\mathbf{x}, y)} [\mathcal{L}(h(\mathbf{x}), y)]$  where  $\mathcal{L}(\cdot, \cdot)$  is the loss function. Its empirical estimate using source labeled examples is denoted as  $\hat{\epsilon}_{T_0}(h)$ . Similarly, we define the target error  $\epsilon_{T_j}(h)$  and the empirical estimate of the target error  $\hat{\epsilon}_{T_j}(h)$  over the target distribution  $p_{T_j}(\mathbf{x}, y)$  at time stamp  $j$ . A natural domain discrepancy measure over joint distributions on  $\mathcal{X} \times \mathcal{Y}$  between features and class labels can be defined as follows:

$$d_1(\mathcal{D}_{T_0}, \mathcal{D}_T) = \sup_{Q \in \mathcal{Q}} |\Pr_{\mathcal{D}_{T_0}}[Q] - \Pr_{\mathcal{D}_T}[Q]| \quad (1)$$

where  $\mathcal{Q}$  is the set of measurable subsets under  $p_{T_0}(\mathbf{x}, y)$  and  $p_T(\mathbf{x}, y)$ <sup>1</sup>. Then, the error bound of continuous transfer learning is given by the following theorem.

**Theorem 3.1.** *Assume the loss function  $\mathcal{L}$  is bounded with  $0 \leq \mathcal{L} \leq M$ . Given a source domain  $\mathcal{D}_{T_0}$  and historical target domain  $\{\mathcal{D}_{T_i}\}_{i=1}^t$ , for  $h \in \mathcal{H}$ , the target domain error  $\epsilon_{T_{t+1}}$  on  $\mathcal{D}_{t+1}$  is*

<sup>1</sup>Note that it is slightly different from  $L_1$  or variation divergence in (Ben-David et al., 2010) with only marginal distribution of features involved.

bounded as follows.

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + M \sum_{j=0}^t \mu^{t-j} d_1(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) \right)$$

where  $\mu \geq 0$  is the domain decay rate<sup>2</sup> indicating the importance of source or historical target domain over  $\mathcal{D}_{T_{t+1}}$ , and  $\bar{\mu} = \sum_{j=0}^t \mu^{t-j}$ .

**Remark.** In particular, we have the following arguments. (1) It is not tractable to accurately estimate  $d_1$  from finite examples in real scenarios (Ben-David et al., 2010); (2) This error bound could be much tighter when considering other advanced domain discrepancy measures, e.g.,  $\mathcal{A}$ -distance (Ben-David et al., 2007), discrepancy distance (Mansour et al., 2009), etc. (3) There are two special cases: when  $\mu = 0$ , the error bound of  $\mathcal{D}_{T_{t+1}}$  would be simply determined by the latest historical target data  $\mathcal{D}_{T_t}$ , and if  $\mu$  goes to infinity,  $\mathcal{D}_{T_{t+1}}$  is just determined by the source data  $\mathcal{D}_{T_0}$  because intuitively the coefficient  $\mu^{t-j}/\bar{\mu}$  of historical target domain data  $\mathcal{D}_{T_j}$  ( $j = 1, \dots, t$ ) converges to zero.

**Corollary 3.2.** With the assumption in Theorem 3.1 and assume that the loss function  $\mathcal{L}$  is symmetric (i.e.,  $\mathcal{L}(y_1, y_2) = \mathcal{L}(y_2, y_1)$  for  $y_1, y_2 \in \mathcal{Y}$ ) and obeys the triangle inequality, Then

- (1) if  $\mathcal{A}$ -distance (Ben-David et al., 2007) is adopted to measure the distribution shift, i.e.,  $d_{\mathcal{H}\Delta\mathcal{H}} = \sup_{h, h' \in \mathcal{H}} |\Pr_{\mathcal{D}_{T_0}}[h(\mathbf{x}) \neq h'(\mathbf{x})] - \Pr_{\mathcal{D}_T}[h(\mathbf{x}) \neq h'(\mathbf{x})]|$ , we have:

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + M \sum_{j=0}^t \mu^{t-j} \left( d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \frac{\lambda_j^*}{M} \right) \right)$$

where  $\lambda_j^* = \min_{h \in \mathcal{H}} \epsilon_{T_j}(h) + \epsilon_{T_{t+1}}(h)$ .

- (2) if discrepancy distance (Mansour et al., 2009) is adopted to measure the distribution shift, i.e.,  $d_{disc}(\mathcal{D}_{T_0}, \mathcal{D}_T) = \max_{h, h' \in \mathcal{H}} |\mathbb{E}_{\mathcal{D}_{T_0}}[\mathcal{L}(h(x), h'(x))] - \mathbb{E}_{\mathcal{D}_T}[\mathcal{L}(h(x), h'(x))]|$ , we have:

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \sum_{j=0}^t \mu^{t-j} (d_{disc}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \Omega_j) \right)$$

where  $\Omega_j = \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h_j^*(\mathbf{x}), y)] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_j^*(\mathbf{x}), h_{t+1}^*(\mathbf{x}))] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_{t+1}^*(\mathbf{x}), y)]$ , and  $h_j^* = \arg \min_{h \in \mathcal{H}} \epsilon_{T_j}(h)$  for  $j = 0, \dots, t, t+1$ .

The aforementioned domain discrepancy measures mainly focus on the marginal distribution over input features and have inspired a line of practical transfer learning algorithms (Ganin et al., 2016; Chen et al., 2019). However, recent work (Wu et al., 2019; Zhao et al., 2019) observed that the minimization of marginal distributions cannot guarantee the success of transfer learning in real scenarios. We propose to address this problem by incorporating the label information in the domain discrepancy measure (see next section).

## 4 LABEL-INFORMED DOMAIN DISCREPANCY

In this section, we introduce a novel label-informed domain discrepancy measure between the source domain  $\mathcal{D}_{T_0}$  and target domain  $\mathcal{D}_T$ , its empirical estimate, and a transfer signature based on this measure to identify potential negative transfer. The use of this discrepancy measure in continuous transfer learning will be discussed in the next section.

### 4.1 $\mathcal{C}$ -DIVERGENCE

For a hypothesis  $h \in \mathcal{H}$ , we denote  $I(h)$  to be the subset of  $\mathcal{X}$  such that  $\mathbf{x} \in I(h) \Leftrightarrow h(\mathbf{x}) = 1$ . In order to estimate the label-informed domain discrepancy from finite samples in practice, instead of Eq. (1), we propose the following  $\mathcal{C}$ -divergence between  $\mathcal{D}_{T_0}$  and  $\mathcal{D}_T$ , taking into consideration the joint distribution over features and class labels:

$$d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T) = \sup_{h \in \mathcal{H}} \left| \Pr_{\mathcal{D}_{T_0}}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_T}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \quad (2)$$

where  $\overline{I(h)}$  is the complement of  $I(h)$ .

<sup>2</sup>In this case, we assume  $\mu^0 = 1$  for any  $\mu \geq 0$ .

We show that some existing domain discrepancy methods (e.g., Ben-David et al. (2007)) can be seen as special cases of this definition by using the following relaxed covariate shift assumption.

**Definition 4.1.** (*Relaxed Covariate Shift Assumption*) The source and target domains satisfy the relaxed covariate shift assumption if for any  $h \in \mathcal{H}$ ,

$$\Pr_{\mathcal{D}_{T_0}}[y | I(h)] = \Pr_{\mathcal{D}_T}[y | I(h)] = \Pr[y | I(h)] \quad (3)$$

Notice that it would be equivalent to covariance shift assumption (Shimodaira, 2000; Johansson et al., 2019) when  $I(h)$  consists of only one example for all  $h \in \mathcal{H}$  (see Lemma A.6 for details).

**Lemma 4.2.** *With the relaxed covariate shift assumption, for any  $h \in \mathcal{H}$ , we have:*

$$d_C(\mathcal{D}_{T_0}, \mathcal{D}_T) = \sup_{h \in \mathcal{H}} \left| \left( \Pr_{\mathcal{D}_{T_0}}[I(h)] - \Pr_{\mathcal{D}_T}[I(h)] \right) \cdot \mathcal{S}_h + \Pr_{\mathcal{D}_T}[y = 1] - \Pr_{\mathcal{D}_{T_0}}[y = 1] \right|$$

where  $\mathcal{S}_h = \Pr[y = 1 | I(h)] - \Pr[y = 0 | I(h)]$ .

**Remark.** From Lemma 4.2, we can see that in the special case where  $\mathcal{S}_h$  is a constant for all  $h \in \mathcal{H}$  and  $\Pr_{\mathcal{D}_T}[y = 1] = \Pr_{\mathcal{D}_{T_0}}[y = 1]$ , the proposed  $\mathcal{C}$ -divergence is reduced to the  $\mathcal{A}$ -distance (Ben-David et al., 2007) defined on the marginal distribution of features. More generally speaking,  $\mathcal{C}$ -divergence can be considered as a weighted version of the  $\mathcal{A}$ -distance where the hypothesis whose characteristic function has a larger class-separability (i.e.,  $|\mathcal{S}_h|$ ) receives a higher weight. Intuitively, compared to  $\mathcal{A}$ -distance,  $\mathcal{C}$ -divergence would pay less attention to class-inseparable regions in the input feature space, which provide irrelevant information for learning the prediction function in the target domain.

Moreover, the following theorem states that in conventional transfer learning scenario with a static source domain and a static target domain, the target error is bounded in terms of  $\mathcal{C}$ -divergence across domains and the expected source error.

**Theorem 4.3.** *Assume that loss function  $\mathcal{L}$  is bounded, i.e., there exists a constant  $M > 0$  such that  $0 \leq \mathcal{L} \leq M$ . For a hypothesis  $h \in \mathcal{H}$ , we have the following bound:*

$$\epsilon_T(h) \leq \epsilon_{T_0}(h) + M \cdot d_C(\mathcal{D}_{T_0}, \mathcal{D}_T)$$

#### 4.2 EMPIRICAL ESTIMATE OF $\mathcal{C}$ -DIVERGENCE

In practice, it is difficult to calculate the proposed  $\mathcal{C}$ -divergence based on Eq. (2) as it uses the true underlying distributions. Therefore, we propose the following empirical estimate of the  $\mathcal{C}$ -divergence between  $\mathcal{D}_{T_0}$  and  $\mathcal{D}_T$  as follows. Assuming that the hypothesis class  $\mathcal{H}$  is symmetric (i.e.,  $1 - h \in \mathcal{H}$  if  $h \in \mathcal{H}$ ), the empirical  $\mathcal{C}$ -divergence is:

$$d_C(\hat{\mathcal{D}}_{T_0}, \hat{\mathcal{D}}_T) = 1 - \min_{h \in \mathcal{H}} \left| \frac{1}{m_{T_0}} \sum_{(\mathbf{x}, y): h(\mathbf{x}) \neq y} \mathbb{I}[(\mathbf{x}, y) \in \hat{\mathcal{D}}_{T_0}] + \frac{1}{m_T} \sum_{(\mathbf{x}, y): h(\mathbf{x}) = y} \mathbb{I}[(\mathbf{x}, y) \in \hat{\mathcal{D}}_T] \right| \quad (4)$$

where  $\hat{\mathcal{D}}_{T_0}$  and  $\hat{\mathcal{D}}_T$  denote the source and target domains with finite samples, respectively.  $\mathbb{I}[a]$  is the binary indicator function which is 1 if  $a$  is true, and 0 otherwise.

The following lemma provides the upper bound of the true  $\mathcal{C}$ -divergence using its empirical estimate.

**Lemma 4.4.** *For any  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$  over  $m_{T_0}$  labeled source examples  $\mathcal{B}_{T_0}$  and  $m_T$  labeled target examples  $\mathcal{B}_T$ , we have:*

$$d_C(\mathcal{D}_{T_0}, \mathcal{D}_T) \leq d_C(\hat{\mathcal{D}}_{T_0}, \hat{\mathcal{D}}_T) + \left( \hat{\mathfrak{R}}_{\mathcal{B}_{T_0}}(L_H) + \hat{\mathfrak{R}}_{\mathcal{B}_T}(L_H) \right) + 3 \left( \sqrt{\frac{\log \frac{4}{\delta}}{2m_{T_0}}} + \sqrt{\frac{\log \frac{4}{\delta}}{2m_T}} \right)$$

where  $\hat{\mathfrak{R}}_{\mathcal{B}}(L_H)(\mathcal{B} \in \{\mathcal{B}_{T_0}, \mathcal{B}_T\})$  denotes the Rademacher complexity (Mansour et al., 2009) over  $\mathcal{B}$  and  $L_H = \{(\mathbf{x}, y) \rightarrow \mathbb{I}[h(\mathbf{x}) = y] : h \in \mathcal{H}\}$  is a class of functions mapping  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  to  $\{0, 1\}$ .

#### 4.3 NEGATIVE TRANSFER CHARACTERIZATION

Informally, negative transfer is considered as the situation where transferring knowledge from the source domain has a negative impact on the target learner (Wang et al., 2019):  $\epsilon_T(A(\mathcal{D}_{T_0}, \mathcal{D}_T)) > \epsilon_T(A(\emptyset, \mathcal{D}_T))$  where  $A$  is the learning algorithm.  $\epsilon_T$  is the target error induced by algorithm  $A$ .  $\emptyset$  implies that it only considers the target data set for target learner. In this paper, we define a **transfer signature** to measure the transferability from source domain to target domain as follows.

$$TS(\mathcal{D}_T | \mathcal{D}_{T_0}) = \inf_{A \in \mathcal{G}} (\epsilon_T(A(\mathcal{D}_{T_0}, \mathcal{D}_T)) - \epsilon_T(A(\emptyset, \mathcal{D}_T))) \quad (5)$$

where  $\mathcal{G}$  is the set of all learning algorithms. We state that source domain knowledge is not transferable over target domain when  $TS(\mathcal{D}_T | \mathcal{D}_{T_0}) > 0$ . Specially, since  $A(\mathcal{D}_{T_0}, \mathcal{D}_T)$  learns an optimal classifier using both source and target data, we can define  $\epsilon_T(A(\mathcal{D}_{T_0}, \mathcal{D}_T)) = \epsilon_T(h_\alpha^*)$

where  $h_\alpha^* = \arg \min_{h \in \mathcal{H}(A)} \alpha \epsilon_T(h) + (1 - \alpha) \epsilon_{T_0}(h)$  and  $\mathcal{H}(A)$  is the hypothesis space induced by  $A$ . When we only consider the target domain with  $\alpha = 1$ ,  $\epsilon_T(A(\emptyset, \mathcal{D}_T)) = \epsilon_T(h_T^*)$  where  $h_T^* = \arg \min_{h \in \mathcal{H}(A)} \epsilon_T(h)$ . Then we have the following theorem regarding the transfer signature.

**Theorem 4.5.** *Assuming the loss function  $\mathcal{L}$  is bounded with  $0 \leq \mathcal{L} \leq M$ , we have*

$$\epsilon_T(h_\alpha^*) \leq \epsilon_T(h_T^*) + 2(1 - \alpha)Md_C(\mathcal{D}_{T_0}, \mathcal{D}_T)$$

Furthermore,

$$TS(\mathcal{D}_T || \mathcal{D}_{T_0}) \leq 2(1 - \alpha)Md_C(\mathcal{D}_{T_0}, \mathcal{D}_T)$$

**Remark.** We have the following observations: (1) Larger  $\mathcal{C}$ -divergence between domains is often associated with a higher transfer signature, which indicates that negative transfer can be characterized using the proposed  $\mathcal{C}$ -divergence; (2) Empirically, the larger amount of labeled target data could increase the value of  $\alpha$ , resulting in the learned classifier relying more on the target data, which is consistent with the observation in (Wang et al., 2019). One extreme case is where  $\alpha = 1$ , implying we have adequate labeled target examples for standard supervised learning on the target domain without transferring knowledge from the source domain.

## 5 PROPOSED ALGORITHM

In this section, we derive the continuous error bound based on our proposed  $\mathcal{C}$ -divergence, followed by a novel continuous transfer learning algorithm (CONTE) by minimizing the error upper bound. Notice that in the context of continuous transfer learning, we also use the proposed  $\mathcal{C}$ -divergence between the target domain at adjacent time stamps to measure the change in distribution over time.

### 5.1 CONTINUOUS ERROR BOUND WITH EMPIRICAL $\mathcal{C}$ -DIVERGENCE

The following theorem states that for a bounded loss function  $\mathcal{L}$ , the target error in continuous transfer learning can be bounded in terms of the empirical classification error within source and historical target domains, the empirical  $\mathcal{C}$ -divergence across domains as well as the empirical Rademacher complexity of the class of functions  $L_H = \{(\mathbf{x}, y) \rightarrow \mathbb{I}[h(\mathbf{x}) = y] : h \in \mathcal{H}\}$ .

**Theorem 5.1.** (Continuous Error Bound) *Assume the loss function  $\mathcal{L}$  is bounded with  $0 \leq \mathcal{L} \leq M$ . Given a source domain  $\mathcal{D}_{T_0}$  and historical target domain  $\{\mathcal{D}_{T_i}\}_{i=1}^t$ , for  $h \in \mathcal{H}$  and  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ , the target domain error  $\epsilon_{T_{t+1}}$  on  $\mathcal{D}_{T_{t+1}}$  is bounded as follows.*

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\mu} \left( \sum_{j=0}^t \mu^{t-j} \hat{\epsilon}_{T_j}(h) + M \sum_{j=0}^t \mu^{t-j} d_C(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}}) + M\Lambda \right)$$

$$\text{where } \Lambda = \sum_{j=0}^t \left( \hat{\mathfrak{R}}_{\mathcal{B}_{T_j}}(L_H) + \hat{\mathfrak{R}}_{\mathcal{B}_{T_{t+1}}}(L_H) + 3\sqrt{\frac{\log \frac{8}{\delta}}{2m_{T_j}}} + 3\sqrt{\frac{\log \frac{8}{\delta}}{2m_{T_{t+1}}}} + \sqrt{\frac{M^2 \log \frac{4}{\delta}}{2m_{T_j}}} \right).$$

**Remark.** Compared to continuous error bounds in Corollary 3.2 using existing domain divergence measures (Ben-David et al. (2007); Mansour et al. (2009)), our bound consists of only data-dependent terms (e.g., empirical source error and  $\mathcal{C}$ -divergence), whereas previous error bounds are determined by the error terms involving the intractable labeling function or optimal target hypothesis (see Corollary 3.2).

### 5.2 CONTE ALGORITHM

For continuous transfer learning, we leverage both the source domain and historical target domain data to learn the predictive function for the current time stamp. To this end, we propose to minimize the error bound in Theorem 5.1 for learning the predictive function on  $\mathcal{D}_{T_{t+1}}$ . Furthermore, we aim to learn a domain-invariant and time-invariant latent feature space such that the  $\mathcal{C}$ -divergence across domains and across time stamps could be minimized. Therefore, we present an adversarial Variational Auto-encoder (VAE) algorithm with the following overall objective function:

$$\mathcal{J}(T_0, T_1, T_2, \dots, T_{t+1}) = \sum_{j=0}^t \mu^{t-j} \left( \mathcal{L}_{clc}(T_j, T_{t+1}) + d_C(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}}) + \lambda \mathcal{L}_{ELBO}(T_j, T_{t+1}) \right) \quad (6)$$

where  $\mathcal{L}_{clc}(T_j, T_{t+1})$  represents the classification error over the labeled examples from  $\mathcal{D}_{T_j}$  and  $\mathcal{D}_{T_{t+1}}$ ,  $d_C(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}})$  is the empirical estimate of  $\mathcal{C}$ -divergence across domain. Thus the first two terms of Eq. (6) are associated with  $\hat{\epsilon}_{T_j}(h) + d_C(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}})$  in the error bound of Theorem 5.1. The third term  $\mathcal{L}_{ELBO}(T_j, T_{t+1})$  is the variational bound in the VAE framework (see Figure 4) when learning the latent feature space and  $\lambda > 0$  is a hyper-parameter. In this case, we have  $\mu \in [0, 1]$  because we assume that the data distribution of a time-evolving target domain shifts smoothly over time. Then we instantiate the terms of Eq. (6) as follows.

Inspired by semi-supervised VAE (Kingma et al., 2014), we propose to learn the feature space by maximizing the following likelihood across domains.

$\log p_\theta(\mathbf{x}, y) = \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}, y)||p_\theta(\mathbf{z}|\mathbf{x}, y)) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, y)}[\log p_\theta(\mathbf{x}, y, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}, y)]$  where  $\phi$  and  $\theta$  are the learnable parameters in the encoder and decoder respectively, and  $\mathbf{z}$  is the latent feature representation of the input example  $(\mathbf{x}, y)$ .  $\text{KL}(\cdot||\cdot)$  is Kullback–Leibler divergence. The evidence lower bound (ELBO), a lower bound on this log-likelihood, can be written as follows.

$$\mathcal{E}_{\theta, \phi}(\mathbf{x}, y) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, y)}[\log p_\theta(\mathbf{x}, y|\mathbf{z})] + \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}, y)||p(\mathbf{z})) \quad (7)$$

where  $\mathcal{E}_{\theta, \phi}(\mathbf{x}, y) \leq \log p_\theta(\mathbf{x}, y)$ . Similarly, we have the following ELBO to maximize the log-likelihood of  $p_\theta(\mathbf{x})$  when the label is not available:

$$\mathcal{U}_{\theta, \phi}(\mathbf{x}) = \sum_y (q_\phi(y|\mathbf{x}) \cdot \mathcal{E}_{\theta, \phi}(\mathbf{x}, y) - \mathbb{E}_{q_\phi(y|\mathbf{x})}[\log q_\phi(y|\mathbf{x})]) \quad (8)$$

where  $p_\theta(\mathbf{x}, y, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(y|\mathbf{z})p(\mathbf{z})$  with prior Gaussian distribution  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Therefore, the **variational bound**  $\mathcal{L}_{ELBO}(T_j, T_{t+1})$  is given below.

$$\mathcal{L}_{ELBO}(T_j, T_{t+1}) = - \sum_{i=1}^{m_{T_j} + m_{T_{t+1}}} \mathcal{E}_{\theta, \phi}(\mathbf{x}_i, y_i) - \sum_{i=1}^{u_{T_{t+1}}} \mathcal{U}_{\theta, \phi}(\mathbf{x}_i, y_i) \quad (9)$$

where  $u_{T_{t+1}}$  is the number of unlabeled training examples from  $\mathcal{D}_{T_{t+1}}$ . Besides, the **classification error**  $\mathcal{L}_{clc}(T_j, T_{t+1})$  can be expressed as follows.

$$\mathcal{L}_{clc}(T_j, T_{t+1}) = \sum_{i=1}^{m_{T_j} + m_{T_{t+1}}} \mathcal{L}(y_i, q_\phi(\cdot|\mathbf{x}_i)) \quad (10)$$

where  $q_\phi(\cdot)$  is the discriminative classifier formed by the distribution  $q_\phi(y|\mathbf{x})$  in Eq. (8), and  $\mathcal{L}(\cdot, \cdot)$  is the cross-entropy loss function in our experiments. To estimate the  $\mathcal{C}$ -divergence, we first define  $\tilde{h}$  to be a two-dimensional characteristic function with  $\tilde{h}(\mathbf{x}, y) = 1 \Leftrightarrow h(\mathbf{x}) = y \Leftrightarrow \{h(\mathbf{x}) = 1, y = 1\} \vee \{h(\mathbf{x}) = 0, y = 0\}$  for  $h \in \mathcal{H}$ . Then the empirical  $\mathcal{C}$ -divergence in Eq. (4) can be rewritten as follows.

$$d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}}) = 1 - \min_{\tilde{h}} \left| \frac{1}{m_{T_j}} \sum_{(\mathbf{x}, y): \tilde{h}(\mathbf{x}, y)=0} \mathbb{I}[(\mathbf{x}, y) \in \hat{\mathcal{D}}_{T_j}] + \frac{1}{m_{T_{t+1}}} \sum_{(\mathbf{x}, y): \tilde{h}(\mathbf{x}, y)=1} \mathbb{I}[(\mathbf{x}, y) \in \hat{\mathcal{D}}_{T_{t+1}}] \right|$$

Note that the latent feature representation  $\mathbf{z}$  learned by  $q_\phi(\mathbf{z}|\mathbf{x}, y)$  could capture the label-informed information of an example  $(\mathbf{x}, y)$ . Thus, the hypothesis  $\tilde{h}$  can be considered as the composition of a feature extraction  $q_\phi$  and a domain classifier  $\mathcal{F}_j$ , i.e.,  $\tilde{h}(\mathbf{x}, y) = \mathcal{F}_j(q_\phi(\mathbf{z}|\mathbf{x}, y))$ . Formally, the **empirical estimate of  $\mathcal{C}$ -divergence** is given below.

$$d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}}) = 1 - \min_{\mathcal{F}_j} \left| \frac{1}{m_{T_j}} \sum_{\mathbf{z}: \mathcal{F}_j(\mathbf{z})=0} \mathbb{I}[\mathbf{z} \in \hat{\mathcal{D}}_{T_j}] + \frac{1}{m_{T_{t+1}}} \sum_{\mathbf{z}: \mathcal{F}_j(\mathbf{z})=1} \mathbb{I}[\mathbf{z} \in \hat{\mathcal{D}}_{T_{t+1}}] \right| \quad (11)$$

The benefits of CONTE are twofold: first, it learns the latent feature space using both input  $\mathbf{x}$  and output  $y$ ; second, it minimizes a tighter error upper bound based on  $\mathcal{C}$ -divergence in Theorem 5.1. This framework can also be interpreted as a minimax game: the VAE learns a domain-invariant and time-invariant latent feature space, whereas the domain classifier  $\mathcal{F}_j$  aims to distinguish the examples from different domains and different time stamps. In this paper, we adopt the gradient reversal layer (Ganin et al., 2016) when updating the parameters of domain classifier  $\mathcal{F}_j$ , and thus CONTE can be optimized by back-propagation in an end-to-end manner (see Algorithm 1 in appendices).

However, we observe that (1) it is difficult to estimate the  $\mathcal{C}$ -divergence with only limited labeled target examples from  $\mathcal{D}_{T_{t+1}}$ ; (2) when learning the latent features  $\mathbf{z}$ , combining the data  $\mathbf{x}$  (e.g., one image) and class-label  $y$  directly might lead to over-emphasizing the data itself due to its high dimensionality compared to  $y$ . To address these problems, we propose the following *Pseudo-label Inference*, i.e., we infer the pseudo labels of unlabeled examples using the classifier  $q_\phi(y|\mathbf{x})$  for each training epoch. Using labeled source and target examples as well as unlabeled target examples with inferred pseudo labels, the  $\mathcal{C}$ -divergence could be estimated in a balanced setting. Furthermore, to enforce the compatibility between features  $\mathbf{x}$  and label  $y$ , we adopt a pre-encoder step to learn a dense representation for the input  $\mathbf{x}$ , and then learn the label-informed latent features  $\mathbf{z}$ .

## 6 EXPERIMENTAL RESULTS

**Synthetic Data:** We generate a synthetic data set in which each domain has 1000 positive examples and 1000 negative examples randomly generated from Gaussian distributions  $\mathcal{N}([1.5 \cos \theta, 1.5 \sin \theta]^T, 0.5 \cdot \mathbf{I}_{2 \times 2})$  and  $\mathcal{N}([1.5 \cos(-\theta), 1.5 \sin(-\theta)]^T, 0.5 \cdot \mathbf{I}_{2 \times 2})$ , respectively. We let  $\theta = 0$  for the source domain (denoted as  $S1$ ), and  $\theta = \frac{i \cdot \pi}{t}$  ( $i = 1, \dots, t$ ) for the time evolving target domain with  $t = 8$  time stamps (denoted as  $T1, \dots, T8$ ).

**Image Data:** We consider the following two tasks: digital classification (MNIST, SVHN) and image classification (Office-31 with three domains: Amazon, DSLR and Webcam; and Office-Home with

four domains: Art, Product, Clipart and Real World). Since standard domains are static in these data sets, we will simulate the time-evolving distribution shift on the target domain by adding noise (e.g., random salt&pepper noise, adversarial noise, rotation). Take SVHN→MNIST as an example, we will use SVHN as the static source domain, and MNIST as the target domain at the first time stamp. By adding adversarial noise to the MNIST images, we obtain a time-evolving target domain (denoted as T1,  $\dots$ , T11 in Table 1). For Office-31 and Office-Home, we add the random salt&pepper noise and rotation to generate the evolving target domain. More details can be found in the appendices.

**Baselines:** The baseline methods are as follows. (1) SourceOnly: training with only source data; (2) TargetERM: empirical risk minimization (ERM) on only target domain; (3) DAN (Long et al., 2015), CORAL (Sun & Saenko, 2016), DANN (Ganin et al., 2016), ADDA (Tzeng et al., 2017), WDGRL (Shen et al., 2018), DIFA (Volpi et al., 2018) and MDD (Zhang et al., 2019): training with feature distribution alignment. (4) CONTE: training with label-informed distribution alignment on the evolving target domain while  $\mu \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ . (5) CONTE $_{\infty}$ : a one-time transfer learning variant of CONTE that directly transfers from source domain to current target domain. We fix  $\lambda = 0.1$ , and all the methods use the same neural network architecture for feature extraction.

### 6.1 EVALUATION OF $\mathcal{C}$ -DIVERGENCE

We compare the proposed  $\mathcal{C}$ -divergence with conventional domain discrepancy measure  $\mathcal{A}$ -distance (Ben-David et al., 2007) on a synthetic data set with an evolving target domain. We assume that the hypothesis space  $\mathcal{H}$  consists of linear classifiers in the feature space. Figure 2 shows the domain discrepancy and target classification accuracy for each pair of source and target domains. We have the following observations. (1) The classification accuracy on the target domain significantly decreases from target domain T1 to T8. One explanation is that the joint distribution  $p(x, y)$  on the time evolving target domain gradually shifted. (2) The  $\mathcal{A}$ -distance increases from S1→T1 to S1→T4, and then decreases from S1→T4 to S1→T8. That is because it only estimates the difference of the marginal feature distribution  $p(x)$  between the source and target domains. (3) The  $\mathcal{C}$ -divergence keeps increasing from S1→T1 to S1→T8, which indicates the decreasing task relatedness between the source and the target domains. Therefore, compared with  $\mathcal{A}$ -distance<sup>3</sup>, the proposed  $\mathcal{C}$ -divergence better characterizes the transferability from the source to the target domains.

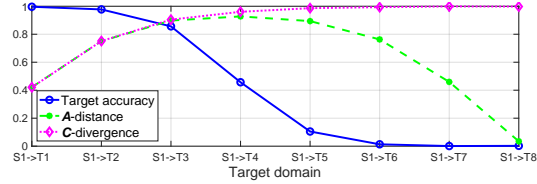


Figure 2: Comparison of domain discrepancy and target accuracy

### 6.2 EVALUATION OF ERROR BOUND

When there is only one time stamp involved in the target domain, Theorem 5.1 is reduced to the standard error bound in the conventional static transfer learning setting. We empirically compare this reduced error bound with the existing Rademacher complexity based error bound in (Mansour et al., 2009) (see Theorem A.4 in appendices for being self-contained).

We use the 0-1 loss function as  $\mathcal{L}$  and assume that the hypothesis space  $\mathcal{H}$  consists of linear classifiers in the feature space. Figure 3 shows the estimated error bounds and target error with the time evolving target domain (i.e., S1→T1,  $\dots$ , S1→T8 in a new synthetic data set with a slower time evolving target domain to ensure that the baseline bound is meaningful most of the time) where we choose  $h = h_{T_0}^*$ . It demonstrates that our  $\mathcal{C}$ -divergence based error bound is much tighter than the baseline. Notice that when transferring source domain S1 to target domain T8, our error bound is largely determined by the  $\mathcal{C}$ -divergence, whereas the baseline is determined by the difference between the optimal source and target hypotheses. Furthermore, given any hypothesis  $h \in \mathcal{H}$ , we may not be able to estimate the baseline bound when the optimal hypothesis is not available.

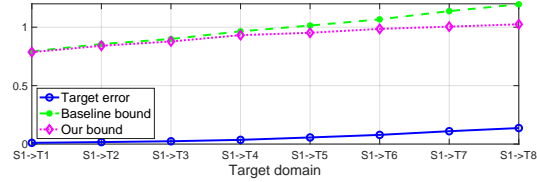


Figure 3: Comparison of error bounds

### 6.3 EVALUATION OF CONTINUOUS TRANSFER LEARNING

Tables 1 and 2 provide the continuous transfer learning results on digital and office-31 data sets where the classification accuracy on target domain is reported (the best results are highlighted in bold). It is observed that (1) the classification accuracy using SourceOnly algorithm significantly

<sup>3</sup>The results for other existing discrepancy measures follow a similar pattern and thus omitted for brevity

decreases on the evolving target domain due to the shift of joint data distribution  $p(\mathbf{x}, y)$  on target domain; (2) the performance of static baseline algorithms is largely affected by the distribution shift in the evolving target domain, and even worse than TargetERM in some cases (e.g., on T6-T11 from SVHN to evolving MNIST); (3) CONTE significantly outperforms  $\text{CONTE}_\infty$  as well as other competitors on target domain by a large margin (i.e., up to 30% improvement on the last time stamp of target domain) because it effectively leverages the historical target domain information to smoothly re-align the target distribution when the change of target domain distribution in consecutive time stamps is small.

Table 1: Transfer learning accuracy from SVHN (source) to time evolving MNIST (target)

Target Domain	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
SourceOnly	0.6998	0.6738	0.6336	0.5692	0.4747	0.4110	0.3087	0.2220	0.1481	0.0828	0.0764
TargetERM	0.7451	0.6997	0.6618	0.6314	0.6368	0.6359	0.6695	0.7133	0.7214	0.7450	0.7512
CORAL	0.8349	0.8410	0.7633	0.7063	0.6496	0.5900	0.5031	0.5101	0.4337	0.4156	0.4502
DANN	0.8666	0.8356	0.8018	0.7529	0.7309	0.6641	0.6614	0.5618	0.5204	0.5082	0.4594
ADDA	0.8667	0.8487	0.7982	0.7187	0.6804	0.5397	0.4366	0.3473	0.2636	0.1659	0.1259
WDGRL	0.8990	0.8602	0.8247	0.8222	0.7452	0.6877	0.6481	0.5896	0.5145	0.4952	0.5196
DIFA	0.9164	0.8993	0.8713	0.8273	0.7935	0.6661	0.5956	0.4381	0.3479	0.2448	0.1332
$\text{CONTE}_\infty$	<b>0.9747</b>	0.9552	0.9514	0.9279	0.8801	0.8833	0.8691	0.6979	0.7030	0.7415	0.7316
CONTE	<b>0.9747</b>	<b>0.9740</b>	<b>0.9803</b>	<b>0.9864</b>	<b>0.9908</b>	<b>0.9940</b>	<b>0.9950</b>	<b>0.9965</b>	<b>0.9970</b>	<b>0.9967</b>	<b>0.9975</b>

Table 2: Transfer learning accuracy on Office-31

	Amazon $\rightarrow$ Webcam					Webcam $\rightarrow$ DSLR				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
SourceOnly	0.7490	0.2255	0.2282	0.1275	0.1503	0.9651	0.4309	0.3329	0.1611	0.2027
TargetERM	0.5584	0.3933	0.4215	0.3396	0.3732	0.4966	0.4201	0.4188	0.3248	0.4067
DAN	0.8537	0.5007	0.4993	0.3638	0.4470	0.9772	0.7302	0.6161	0.4765	0.5302
CORAL	0.8711	0.5235	0.4819	0.3195	0.4054	0.9812	0.7289	0.6671	0.4846	0.5221
DANN	0.8389	0.4993	0.4121	0.3973	0.3382	0.9651	0.7356	0.6416	0.4510	0.5490
MDD	0.8940	0.6738	0.5490	0.5141	0.4295	0.9724	0.8738	0.7315	0.5047	0.5289
$\text{CONTE}_\infty$	<b>0.9154</b>	0.6376	0.5758	0.4591	0.4846	0.9785	0.8591	0.7289	0.4926	0.5557
CONTE	<b>0.9154</b>	<b>0.8134</b>	<b>0.8081</b>	<b>0.7611</b>	<b>0.7826</b>	0.9785	<b>0.9235</b>	<b>0.9208</b>	<b>0.8886</b>	<b>0.9154</b>

## 7 RELATED WORK

**Transfer Learning:** Transfer learning (Ying et al., 2018; Jang et al., 2019) improves the performance of a learning algorithm on the target domain by using the knowledge from the source domain. It is theoretically proven that the target error is well bounded (Ben-David et al., 2010; Mansour et al., 2009), followed by a line of practical algorithms (Shen et al., 2018; Long et al., 2017; 2018; Saito et al., 2018; Chen et al., 2019) with covariate shift assumption. However, it is observed that this assumption does not always hold in real-world scenarios (Rosenstein et al., 2005; Wang et al., 2019).

**Multi-source Domain Adaptation:** Multi-source domain adaptation improves the target prediction function from multiple source domains (Zhao et al., 2018; Hoffman et al., 2018; Wen et al., 2020). It is similar to our problem setting as source and historical target domains can be considered as multiple “source” domains when modeling the target domain at the current time stamp. However, only limited labeled target examples are provided in our problem setting, whereas multi-source domain adaptation requires that all source domains have adequate labeled examples.

**Continual Learning:** Continual lifelong learning (Parisi et al., 2019; Rusu et al., 2016; Hoffman et al., 2014; Bobu et al., 2018) involves the sequential learning tasks with the goal of learning a predictive function on the new task using knowledge from historical tasks. Most of them focused on mitigating catastrophic forgetting when learning new tasks from only one evolving domain, whereas our work studied the transferability between a source domain and a time evolving target domain.

## 8 CONCLUSION

In this paper, we study continuous transfer learning with a time evolving target domain, which has not been widely studied and yet is commonly seen in many real applications. We start by deriving a generic error bound of continuous transfer learning with flexible domain discrepancy measures. Then we propose a novel label-informed  $\mathcal{C}$ -divergence to measure the domain discrepancy incorporating the label information, and study its application in continuous transfer learning, which leads to an improved error bound. Based on this bound, we further propose a generic adversarial Variational Auto-encoder algorithm named CONTE for continuous transfer learning. Extensive experiments on both synthetic and real data sets demonstrate the effectiveness of our CONTE algorithm.



## REFERENCES

- Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, 2007.
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 2010.
- Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. In *International Conference on Learning Representations Workshop*, 2018.
- Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, 2019.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems*, 2018.
- Yunhun Jang, Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Learning what and where to transfer. In *International Conference on Machine Learning*, 2019.
- Fredrik D Johansson, Rajesh Ranganath, and David Sontag. Support and invertibility in domain-invariant representations. In *International Conference on Artificial Intelligence and Statistics*, 2019.
- Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014.
- Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 2015.
- Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, 2017.
- Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018.
- Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.
- Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009.
- German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.
- Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, 2005.

- Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 2000.
- Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.
- Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- Junfeng Wen, Russell Greiner, and Dale Schuurmans. Domain aggregation networks for multi-source domain adaptation. In *International Conference on Machine Learning*, pp. 10927–10937, 2020.
- Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, 2019.
- Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. In *International Conference on Machine Learning*, 2018.
- Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, 2019.
- Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pp. 8559–8570, 2018.
- Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. In *International Conference on Machine Learning*, 2019.

## A ADDITIONAL RESULTS

### A.1 NOTATION

The main notation used in this paper is summarized in Table 3.

Table 3: Notation	
Notation	Definition
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	Input space, class space, latent feature space
$\mathcal{D}_S, \{\mathcal{D}_{T_t}\}_{t=1}^n$	Source domain and evolving target domain
$\mathcal{D}_{T_0}$	Source domain $\mathcal{D}_S$
$\epsilon_S, \hat{\epsilon}_S$	Expected and estimated source error
$\epsilon_T, \hat{\epsilon}_T$	Expected and estimated target error
$p_S, p_T$	Probability density functions (pdf)
$\Pr_{\mathcal{D}_S}, \Pr_{\mathcal{D}_T}$	Probability mass functions (pmf)
$m_S, m_T$	Number of labeled source and target samples

### A.2 THEORETICAL ANALYSIS

We first introduce some useful existing lemmas and theorems for being self-contained, followed by the details regarding the proof for lemmas and theorems involved in this paper.

#### A.2.1 EXISTING DEFINITIONS, LEMMAS AND THEOREMS

**Definition A.1.** (*Rademacher Complexity* (Mansour et al., 2009)) Given a set of real-valued functions  $\mathcal{F}$  over  $\mathcal{X}$  and an example  $\mathcal{B} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \in \mathcal{X}^m$ , the empirical Rademacher complexity of  $\mathcal{F}$  is defined as follow:

$$\hat{\mathfrak{R}}_{\mathcal{B}}(\mathcal{F}) = \frac{2}{m} \mathbb{E}_{\sigma} \left[ \sup_{f \in \mathcal{F}} \left| \sum_{i=1}^m \sigma_i f(\mathbf{x}_i) \right| \middle| \mathcal{B} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \right]$$

where  $\sigma = (\sigma_1, \dots, \sigma_m)$  with each  $\sigma_i$  sampling from two values  $\{-1, +1\}$  according to an independent and uniform distribution.

**Lemma A.2.** (*McDiarmid’s inequality*) Let  $X_1, \dots, X_m$  be independently random variables taking values in the set  $\mathcal{X}$  and  $f : \mathcal{X}^m \rightarrow \mathbb{R}$  be a function over  $X_1, \dots, X_m$  that satisfies  $\forall i, \forall x_1, \dots, x_m, x'_i \in \mathcal{X}$ ,

$$|f(x_1, \dots, x_i, \dots, x_m) - f(x_1, \dots, x'_i, \dots, x_m)| \leq c_i$$

Then, for any  $\epsilon > 0$ ,

$$\Pr[f - \mathbb{E}[f] \geq \epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^m c_i^2}\right)$$

**Lemma A.3.** (*Hoeffding’s inequality*) If  $X_1, \dots, X_m$  are independently random variables with  $a_i \leq X_i \leq b_i$ , then for any  $\epsilon > 0$ ,

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

where  $\bar{X} = (X_1 + \dots + X_m)/m$  and  $\mathbb{E}[\bar{X}]$  is the expectation over  $\bar{X}$ .

We restate the conventional error bound based on Rademacher complexity (see Theorem 8 in Mansour et al. (2009)) as follows.

**Theorem A.4.** (*Error Bound in Mansour et al. (2009)*) Assume that the loss function  $\mathcal{L}$  is symmetric and obeys the triangle inequality. Then, for any hypothesis  $h \in \mathcal{H}$ , the following holds

$$\epsilon_T(h) \leq \epsilon_T(h_T^*) + \mathbb{E}_{\mathbf{x} \sim p_S(\mathbf{x})} [\mathcal{L}(h(\mathbf{x}), h_S^*(\mathbf{x}))] + \mathbb{E}_{\mathbf{x} \sim p_S(\mathbf{x})} [\mathcal{L}(h_T^*(\mathbf{x}), h_S^*(\mathbf{x}))] + d_{\mathcal{L}}(\mathcal{D}_S, \mathcal{D}_T)$$

where  $d_{\mathcal{L}}(\mathcal{D}_S, \mathcal{D}_T) = \max_{h, h' \in \mathcal{H}} |\mathbb{E}_{\mathbf{x} \sim p_S(\mathbf{x})} [\mathcal{L}(h(\mathbf{x}), h'(\mathbf{x}))] - \mathbb{E}_{\mathbf{x} \sim p_T(\mathbf{x})} [\mathcal{L}(h(\mathbf{x}), h'(\mathbf{x}))]|$ , and  $h_S^*, h_T^*$  denote the optimal hypotheses of  $\epsilon_S(h)$  and  $\epsilon_T(h)$ , respectively.

#### A.2.2 OUR RESULTS

Then we provide the theoretical analysis and proof regarding our lemmas and theorems as follows.

**Lemma A.5.** Assume that loss function  $\mathcal{L}$  is bounded, i.e., there exists  $M > 0$  such that  $0 \leq \mathcal{L} \leq M$ . For  $h \in \mathcal{H}$  and  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$  over  $m$  samples  $\mathcal{B}_S$  drawn from  $\mathcal{D}_S$ , we have:

$$\Pr[|\hat{\epsilon}_S(h) - \epsilon_S(h)| \geq \epsilon] \leq 2 \exp(-2m\epsilon^2/M^2)$$

*Proof.* It simply follows the Hoeffding's equality considering  $0 \leq \mathcal{L}(h(\mathbf{x}), y) \leq M$  for each sample.  $\square$

**Lemma A.6.** (Property of Relaxed Covariate Shift Assumption) If the covariate shift assumption between source and target domains holds, and source and target examples follow the IID assumption w.r.t.  $p_S(\mathbf{x}, y)$  and  $p_T(\mathbf{x}, y)$  respectively, then the relaxed covariate shift assumption holds. Furthermore, it would be equivalent to covariance shift assumption when  $I(h)$  consists of only one example for all  $h \in \mathcal{H}$ .

*Proof.* For either source or target domain, if its examples follow the IID assumption, then we have

$$\begin{aligned} \Pr(y|I(h))\Pr(I(h)) &= \Pr(y, I(h)) = \Pr(y, \mathbf{x}_1) \cdots \Pr(y, \mathbf{x}_n) \\ &= \Pr(y|\mathbf{x}_1)\Pr(\mathbf{x}_1) \cdots \Pr(y|\mathbf{x}_n)\Pr(\mathbf{x}_n) \\ &= \Pr(y|\mathbf{x}_1) \cdots \Pr(y|\mathbf{x}_n)\Pr(I(h)) \end{aligned}$$

where we denote  $\mathbf{x}_1, \dots, \mathbf{x}_n$  are the data points in the set  $I(h)$ . Then if covariate shift assumption holds, i.e.,  $\Pr_S(y|\mathbf{x}_i) = \Pr_T(y|\mathbf{x}_i)$  for all examples  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we have  $\Pr_S(y|I(h)) = \Pr_T(y|I(h))$  as shown in the relaxed covariance shift assumption (see Definition 4.1). It is easy to show that when  $I(h)$  consists of only one example for all  $h \in \mathcal{H}$ , it is equivalent to covariance shift assumption.  $\square$

**Lemma A.7.** (Triangle Inequality of  $\mathcal{C}$ -divergence) Given domains  $\mathcal{D}_1$ ,  $\mathcal{D}_2$  and  $\mathcal{D}_3$ , the  $\mathcal{C}$ -divergence satisfies the following triangle property:

$$d_{\mathcal{C}}(\mathcal{D}_1, \mathcal{D}_2) \leq d_{\mathcal{C}}(\mathcal{D}_1, \mathcal{D}_3) + d_{\mathcal{C}}(\mathcal{D}_2, \mathcal{D}_3) \quad (12)$$

*Proof.* Following the definition of  $\mathcal{C}$ -divergence in Eq. (2), it is easy to show the  $\mathcal{C}$ -divergence is symmetric with respect to its two arguments. Then we have

$$\begin{aligned} d_{\mathcal{C}}(\mathcal{D}_1, \mathcal{D}_2) &= \sup_{h \in \mathcal{H}} \left| \Pr_{\mathcal{D}_1}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_2}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \\ &= \sup_{h \in \mathcal{H}} \left| \Pr_{\mathcal{D}_1}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_3}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right. \\ &\quad \left. + \Pr_{\mathcal{D}_3}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_2}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \\ &\leq \sup_{h \in \mathcal{H}} \left| \Pr_{\mathcal{D}_1}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_3}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \\ &\quad + \sup_{h \in \mathcal{H}} \left| \Pr_{\mathcal{D}_3}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_2}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \\ &= d_{\mathcal{C}}(\mathcal{D}_1, \mathcal{D}_3) + d_{\mathcal{C}}(\mathcal{D}_2, \mathcal{D}_3) \end{aligned}$$

which completes the proof.  $\square$

**Proof of Theorem 3.1.** Theorem 3.1 states that assume the loss function  $\mathcal{L}$  is bounded with  $0 \leq \mathcal{L} \leq M$ . Given a source domain  $\mathcal{D}_{T_0}$  and historical target domain  $\{\mathcal{D}_{T_i}\}_{i=1}^t$ , for  $h \in \mathcal{H}$ , the target domain error  $\epsilon_{T_{t+1}}$  on  $\mathcal{D}_{t+1}$  is bounded as follows.

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + M \sum_{j=0}^t \mu^{t-j} d_1(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) \right)$$

where  $\mu \geq 0$  is the domain decay rate indicating the importance of source or historical target domain over  $\mathcal{D}_{T_{t+1}}$ , and  $\bar{\mu} = \sum_{j=0}^t \mu^{t-j}$ .

*Proof.*

$$\begin{aligned}
\epsilon_{T_{t+1}}(h) &\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \left| \epsilon_{T_{t+1}}(h) - \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) \right| \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} |\epsilon_{T_{t+1}}(h) - \epsilon_{T_j}(h)| \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \left( \sum_y \int |p_{T_{t+1}}(\mathbf{x}, y) - p_{T_j}(\mathbf{x}, y)| |\mathcal{L}(h(\mathbf{x}), y)| d\mathbf{x} \right) \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{M}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \left( \sum_y \int |p_{T_{t+1}}(\mathbf{x}, y) - p_{T_j}(\mathbf{x}, y)| d\mathbf{x} \right) \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{M}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} d_1(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}})
\end{aligned}$$

which completes the proof.  $\square$

**Proof of Corollary 3.2.** Corollary 3.2 states that with the assumption in Theorem 3.1 and assume that the loss function  $\mathcal{L}$  is symmetric (i.e.,  $\mathcal{L}(y_1, y_2) = \mathcal{L}(y_2, y_1)$  for  $y_1, y_2 \in \mathcal{Y}$ ) and obeys the triangle inequality, Then

- (1) if  $\mathcal{A}$ -distance (Ben-David et al., 2007) is adopted to measure the distribution shift across domains, i.e.,  $d_{\mathcal{H}\Delta\mathcal{H}} = \sup_{h, h' \in \mathcal{H}} |\Pr_{\mathcal{D}_{T_0}}[h(\mathbf{x}) \neq h'(\mathbf{x})] - \Pr_{\mathcal{D}_T}[h(\mathbf{x}) \neq h'(\mathbf{x})]|$ , the following holds:

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + M \sum_{j=0}^t \mu^{t-j} \left( d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \frac{\lambda_j^*}{M} \right) \right)$$

where  $\lambda_j^* = \min_{h \in \mathcal{H}} \epsilon_{T_j}(h) + \epsilon_{T_{t+1}}(h)$ .

- (2) if discrepancy distance (Mansour et al., 2009) is adopted to measure the distribution shift across domains, i.e.,  $d_{disc}(\mathcal{D}_{T_0}, \mathcal{D}_T) = \max_{h, h' \in \mathcal{H}} |\mathbb{E}_{\mathcal{D}_{T_0}}[\mathcal{L}(h(x), h'(x))] - \mathbb{E}_{\mathcal{D}_T}[\mathcal{L}(h(x), h'(x))]|$ , the following holds:

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \sum_{j=0}^t \mu^{t-j} (d_{disc}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \Omega_j) \right)$$

where  $\Omega_j = \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h_j^*(\mathbf{x}), y)] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_j^*(\mathbf{x}), h_{t+1}^*(\mathbf{x}))] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_{t+1}^*(\mathbf{x}), y)]$ , and  $h_j^* = \arg \min_{h \in \mathcal{H}} \epsilon_{T_j}(h)$  for  $j = 0, \dots, t, t+1$ .

*Proof.* (2) Given  $h_j^* = \arg \min_{h \in \mathcal{H}} \epsilon_{T_j}(h)$  for  $j = 0, \dots, t, t+1$ , we have

$$\begin{aligned}
\epsilon_{T_{t+1}}(h) &\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \left( \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h(\mathbf{x}), h_j^*(\mathbf{x}))] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_j^*(\mathbf{x}), h_{t+1}^*(\mathbf{x}))] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_{t+1}^*(\mathbf{x}), y)] \right) \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \left( \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h(\mathbf{x}), h_j^*(\mathbf{x}))] + d_{disc}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) \right. \\
&\quad \left. + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_j^*(\mathbf{x}), h_{t+1}^*(\mathbf{x}))] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_{t+1}^*(\mathbf{x}), y)] \right) \\
&= \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \sum_{j=0}^t \mu^{t-j} (d_{disc}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \Omega_j) \right)
\end{aligned}$$

where  $\Omega_j = \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h_j^*(\mathbf{x}), y)] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_j^*(\mathbf{x}), h_{t+1}^*(\mathbf{x}))] + \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_{t+1}^*(\mathbf{x}), y)]$ .

(1) Given  $h_j^* = \arg \min_{h \in \mathcal{H}} \epsilon_{T_j}(h) + \epsilon_{T_{t+1}}(h)$ , we have

$$\begin{aligned}
\epsilon_{T_{t+1}}(h) &\leq \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \left| \epsilon_{T_{t+1}}(h) - \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) \right| \\
&\leq \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} |\epsilon_{T_{t+1}}(h) - \epsilon_{T_j}(h)| \\
&= \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \left( \left| \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h(\mathbf{x}), y)] - \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h(\mathbf{x}), y)] \right| \right) \\
&\leq \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \left( \left| \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h(\mathbf{x}), y)] - \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h(\mathbf{x}), h_j^*(\mathbf{x}))] \right| \right) \\
&\quad + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \left( \left| \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h(\mathbf{x}), h_j^*(\mathbf{x}))] - \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h(\mathbf{x}), h_j^*(\mathbf{x}))] \right| \right) \\
&\quad + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \left( \left| \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h(\mathbf{x}), h_j^*(\mathbf{x}))] - \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h(\mathbf{x}), y)] \right| \right) \\
&\leq \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \left( \mathbb{E}_{\mathcal{D}_{T_{t+1}}}[\mathcal{L}(h_j^*(\mathbf{x}), y)] + Md_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \mathbb{E}_{\mathcal{D}_{T_j}}[\mathcal{L}(h_j^*(\mathbf{x}), y)] \right) \\
&\leq \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\mu} \sum_{j=0}^t \mu^{t-j} (Md_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) + \lambda_j^*)
\end{aligned}$$

which completes the proof.  $\square$

**Proof of Lemma 4.2.** Lemma 4.2 states that with relaxed covariate shift assumption, for any  $h \in \mathcal{H}$ , we have

$$d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T) = \sup_{h \in \mathcal{H}} \left| \left( \Pr_{\mathcal{D}_{T_0}}[I(h)] - \Pr_{\mathcal{D}_T}[I(h)] \right) \cdot \mathcal{S}_h + \Pr_{\mathcal{D}_T}[y = 1] - \Pr_{\mathcal{D}_{T_0}}[y = 1] \right|$$

where

$$\mathcal{S}_h = \Pr[y = 1|I(h)] - \Pr[y = 0|I(h)]$$

*Proof.* For any  $h \in \mathcal{H}$ , we have

$$\begin{aligned}
&\Pr_{\mathcal{D}_{T_0}}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_T}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \\
&= \Pr_{\mathcal{D}_{T_0}}[I(h), y = 1] + \Pr_{\mathcal{D}_{T_0}}[y = 0] - \Pr_{\mathcal{D}_{T_0}}[I(h), y = 0] \\
&\quad - \Pr_{\mathcal{D}_T}[I(h), y = 1] - \Pr_{\mathcal{D}_T}[y = 0] + \Pr_{\mathcal{D}_T}[I(h), y = 0] \\
&= 2\Pr_{\mathcal{D}_{T_0}}[I(h), y = 1] + 1 - \Pr_{\mathcal{D}_{T_0}}[y = 1] - \Pr_{\mathcal{D}_{T_0}}[I(h)] \\
&\quad - 2\Pr_{\mathcal{D}_T}[I(h), y = 1] - (1 - \Pr_{\mathcal{D}_T}[y = 1]) + \Pr_{\mathcal{D}_T}[I(h)] \\
&= \left( \Pr_{\mathcal{D}_{T_0}}[I(h)] - \Pr_{\mathcal{D}_T}[I(h)] \right) \left( 2\Pr_{\mathcal{D}_{T_0}}[y = 1|I(h)] - 1 \right) + \left( \Pr_{\mathcal{D}_T}[y = 1] - \Pr_{\mathcal{D}_{T_0}}[y = 1] \right) \\
&\quad + \Pr_{\mathcal{D}_T}[I(h)] \left( \Pr_{\mathcal{D}_{T_0}}[y = 1|I(h)] - \Pr_{\mathcal{D}_T}[y = 1|I(h)] \right)
\end{aligned}$$

With the relaxed covariate shift assumption  $\Pr_{\mathcal{D}_S}[y | I(h)] = \Pr_{\mathcal{D}_T}[y | I(h)] = \Pr[y | I(h)]$ , we have

$$\begin{aligned}
d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T) &= \sup_{h \in \mathcal{H}} \left| \Pr_{\mathcal{D}_{T_0}}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_T}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \\
&= \sup_{h \in \mathcal{H}} \left| \left( \Pr_{\mathcal{D}_{T_0}}[I(h)] - \Pr_{\mathcal{D}_T}[I(h)] \right) \left( 2\Pr[y = 1|I(h)] - 1 \right) + \left( \Pr_{\mathcal{D}_T}[y = 1] - \Pr_{\mathcal{D}_{T_0}}[y = 1] \right) \right| \\
&= \sup_{h \in \mathcal{H}} \left| \left( \Pr_{\mathcal{D}_{T_0}}[I(h)] - \Pr_{\mathcal{D}_T}[I(h)] \right) \cdot \mathcal{S}_h + \Pr_{\mathcal{D}_T}[y = 1] - \Pr_{\mathcal{D}_{T_0}}[y = 1] \right|
\end{aligned}$$

which completes the proof.  $\square$

**Proof of Theorem 4.3.** Theorem 4.3 states that if loss function  $\mathcal{L}$  is bounded, i.e., there exists  $M > 0$  such that  $0 \leq \mathcal{L} \leq M$ , for a hypothesis  $h \in \mathcal{H}$ , the target error can be bounded by the source error and the  $\mathcal{C}$ -divergence between the distributions  $\mathcal{D}_{T_0}$  and  $\mathcal{D}_T$ . Specifically, we have

$$\epsilon_T(h) \leq \epsilon_{T_0}(h) + M \cdot d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T)$$

*Proof.* Given  $\epsilon_{T_0}(h) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_{T_0}} [\mathcal{L}(h(\mathbf{x}), y)]$ , we have

$$\begin{aligned} \epsilon_T(h) &= \epsilon_{T_0}(h) + \epsilon_T(h) - \epsilon_{T_0}(h) \\ &\leq \epsilon_{T_0}(h) + \left| \Pr_{\mathcal{D}_{T_0}}[\mathcal{L}(h(\mathbf{x}), y)] - \Pr_{\mathcal{D}_T}[\mathcal{L}(h(\mathbf{x}), y)] \right| \\ &\leq \epsilon_{T_0}(h) + M \cdot \left| \Pr_{\mathcal{D}_{T_0}}[h(\mathbf{x}) \neq y] - \Pr_{\mathcal{D}_T}[h(\mathbf{x}) \neq y] \right| \\ &= \epsilon_{T_0}(h) + M \cdot \left| \Pr_{\mathcal{D}_{T_0}}[h(\mathbf{x}) = y] - \Pr_{\mathcal{D}_T}[h(\mathbf{x}) = y] \right| \\ &= \epsilon_{T_0}(h) + M \cdot \left| \Pr_{\mathcal{D}_{T_0}}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] - \Pr_{\mathcal{D}_T}[\{I(h), y = 1\} \cup \{\overline{I(h)}, y = 0\}] \right| \\ &\leq \epsilon_{T_0}(h) + M \cdot d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T) \end{aligned}$$

which completes the proof.  $\square$

**Proof of Lemma 4.4.** Lemma 4.4 states that for any  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$  over  $m_{T_0}$  labeled source samples  $\mathcal{B}_{T_0}$  and  $m_T$  labeled target samples  $\mathcal{B}_T$ , we have:

$$d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T) \leq d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_0}, \hat{\mathcal{D}}_T) + \left( \hat{\mathfrak{R}}_{\mathcal{B}_{T_0}}(L_H) + \hat{\mathfrak{R}}_{\mathcal{B}_T}(L_H) \right) + 3 \left( \sqrt{\frac{\log \frac{4}{\delta}}{2m_{T_0}}} + \sqrt{\frac{\log \frac{4}{\delta}}{2m_T}} \right)$$

*Proof.* Based on the Rademacher Bound (Mansour et al., 2009), with probability at least  $1 - \delta/2$  over  $m_S$  labeled source samples  $\mathcal{B}_S$ , we have

$$\mathbb{E}_{(\mathbf{x}, y) \sim p_{T_0}(\mathbf{x}, y)}[h(\mathbf{x}) = y] \leq \mathbb{E}_{(\mathbf{x}, y) \sim \hat{p}_{T_0}(\mathbf{x}, y)}[h(\mathbf{x}) = y] + \hat{\mathfrak{R}}_{\mathcal{B}_{T_0}}(L_H) + 3 \sqrt{\frac{\log \frac{4}{\delta}}{2m_{T_0}}}$$

where  $\hat{p}_S(\mathbf{x}, y)$  is the empirical estimated probability density function on source domain. Since  $\Pr_{\mathcal{D}_S}[h(\mathbf{x}) = y] = \mathbb{E}_{(\mathbf{x}, y) \sim p_S(\mathbf{x}, y)}[h(\mathbf{x}) = y]$  for any  $h \in \mathcal{H}$ . Thus,

$$d_{\mathcal{C}}(\mathcal{D}_{T_0}, \hat{\mathcal{D}}_{T_0}) \leq \hat{\mathfrak{R}}_{\mathcal{B}_{T_0}}(L_H) + 3 \sqrt{\frac{\log \frac{4}{\delta}}{2m_{T_0}}}$$

The same result holds for target domain. Based on the triangle inequality,

$$\begin{aligned} d_{\mathcal{C}}(\mathcal{D}_{T_0}, \mathcal{D}_T) &\leq d_{\mathcal{C}}(\mathcal{D}_{T_0}, \hat{\mathcal{D}}_{T_0}) + d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_0}, \hat{\mathcal{D}}_T) + d_{\mathcal{C}}(\mathcal{D}_T, \hat{\mathcal{D}}_T) \\ &\leq d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_0}, \hat{\mathcal{D}}_T) + \left( \hat{\mathfrak{R}}_{\mathcal{B}_{T_0}}(L_H) + \hat{\mathfrak{R}}_{\mathcal{B}_T}(L_H) \right) + 3 \left( \sqrt{\frac{\log \frac{4}{\delta}}{2m_{T_0}}} + \sqrt{\frac{\log \frac{4}{\delta}}{2m_T}} \right) \end{aligned}$$

which completes the proof.  $\square$

**Proof of Theorem 4.5.** Theorem 4.5 states that if loss function  $\mathcal{L}$  is bounded, let  $\epsilon_{\alpha}(h) = \alpha \epsilon_T(h) + (1 - \alpha) \epsilon_S(h)$ , then we have

$$\epsilon_T(h_{\alpha}^*) \leq \epsilon_T(h_T^*) + 2(1 - \alpha) M d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T)$$

Furthermore,

$$TS(\mathcal{D}_T || \mathcal{D}_S) \leq 2(1 - \alpha) M d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T)$$

*Proof.* It is easy to show  $|\epsilon_{\alpha}(h) - \epsilon_T(h)| = (1 - \alpha) |\epsilon_T(h) - \epsilon_S(h)| \leq (1 - \alpha) M \cdot d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T)$ . Then

$$\begin{aligned} \epsilon_T(h_{\alpha}^*) &\leq \epsilon_{\alpha}(h_{\alpha}^*) + (1 - \alpha) M d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T) \\ &\leq \epsilon_{\alpha}(h_T^*) + (1 - \alpha) M d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T) \\ &\leq \epsilon_T(h_T^*) + 2(1 - \alpha) M d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T) \end{aligned}$$

Then, the transfer signature can be bounded as follows.

$$\begin{aligned}
TS(\mathcal{D}_T || \mathcal{D}_S) &= \inf_{A \in \mathcal{G}} \left( \epsilon_T(A(\mathcal{D}_S, \mathcal{D}_T)) - \epsilon_T(A(\emptyset, \mathcal{D}_T)) \right) \\
&= \inf_{A \in \mathcal{G}} (\epsilon_T(h_\alpha^*) - \epsilon_T(h_T^*)) \\
&\leq \inf_{A \in \mathcal{G}} (2(1 - \alpha)Md_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T)) \\
&= 2(1 - \alpha)Md_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T)
\end{aligned}$$

where both  $M$  and  $d_{\mathcal{C}}(\mathcal{D}_S, \mathcal{D}_T)$  are model-agnostic.  $\square$

**Proof of Theorem 5.1.** Theorem 5.1 states that assume the loss function  $\mathcal{L}$  is bounded with  $0 \leq \mathcal{L} \leq M$ . Given a source domain  $\mathcal{D}_{T_0}$  and historical target domain  $\{\mathcal{D}_{T_i}\}_{i=1}^t$ , for  $h \in \mathcal{H}$  and  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ , the target domain error  $\epsilon_{T_{t+1}}$  on  $\mathcal{D}_{t+1}$  is bounded as follows.

$$\epsilon_{T_{t+1}}(h) \leq \frac{1}{\bar{\mu}} \left( \sum_{j=0}^t \mu^{t-j} \hat{\epsilon}_{T_j}(h) + M \sum_{j=0}^t \mu^{t-j} d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_j}, \hat{\mathcal{D}}_{T_{t+1}}) + M\Lambda \right)$$

$$\text{where } \Lambda = \sum_{j=0}^t \left( \hat{\mathcal{R}}_{\mathcal{B}_{T_j}}(L_H) + \hat{\mathcal{R}}_{\mathcal{B}_{T_{t+1}}}(L_H) + 3\sqrt{\frac{\log \frac{8}{\delta}}{2m_{T_j}}} + 3\sqrt{\frac{\log \frac{8}{\delta}}{2m_{T_{t+1}}}} + \sqrt{\frac{M^2 \log \frac{4}{\delta}}{2m_{T_j}}} \right).$$

*Proof.* Let  $\mathcal{D}_{ST} = \sum_{j=0}^t \mathcal{D}_j$  be a decay rate guided mixture distribution of source and historical target domains. Then for any  $h \in \mathcal{H}$  we have  $\epsilon_{ST}(h) = \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h)$ .

Using Theorem 4.3, Lemma 4.4 and Lemma A.5, the following holds

$$\begin{aligned}
\epsilon_{T_{t+1}}(h) &\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} (\epsilon_{T_{t+1}}(h) - \epsilon_{T_j}(h)) \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \epsilon_{T_j}(h) + \frac{M}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} d_{\mathcal{C}}(\mathcal{D}_{T_j}, \mathcal{D}_{T_{t+1}}) \\
&\leq \frac{1}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} \hat{\epsilon}_{T_i}(h) + \frac{M}{\bar{\mu}} \sum_{j=0}^t \mu^{t-j} d_{\mathcal{C}}(\hat{\mathcal{D}}_{T_i}, \hat{\mathcal{D}}_{T_{t+1}}) \\
&\quad + \frac{M}{\bar{\mu}} \sum_{j=0}^t \left( \hat{\mathcal{R}}_{\mathcal{B}_{T_j}}(L_H) + \hat{\mathcal{R}}_{\mathcal{B}_{T_{t+1}}}(L_H) + 3\sqrt{\frac{\log \frac{8}{\delta}}{2m_{T_j}}} + 3\sqrt{\frac{\log \frac{8}{\delta}}{2m_{T_{t+1}}}} + \sqrt{\frac{M^2 \log \frac{4}{\delta}}{2m_{T_j}}} \right)
\end{aligned}$$

which completes the proof.  $\square$

### A.3 PROPOSED FRAMEWORK

As illustrated in Algorithm 1, we first learn the predictive function on the first target domain using the knowledge from source domain by minimizing the objective function  $\mathcal{J}(T_0, T_1)$  (see Step 4). Then it predicts the labels for the target domain  $\mathcal{D}_{T_1}$ , which would be used to learn the predictive function for the target domain at time stamp 2 (see Step 5-6). In this case, both labeled training

examples and unlabeled training examples with pseudo-labels from historical target domain are used to minimize the objective function  $\mathcal{J}(T_0, T_1, \dots, T_{i+1})$  for learning the predictive function on  $\mathcal{D}_{T_{i+1}}$ . Repeat this procedure until the predictive function on the  $(t+1)^{\text{th}}$  target domain is optimized. This allows us to optimize the predictive function at any time stamp using the knowledge from source domain and historical target domain.

---

#### Algorithm 1 Continuous Transfer Learning (CONTE)

---

- 1: **Input:** Source domain  $\mathcal{D}_{T_0}$ , target domain  $\{\mathcal{D}_{T_i}\}_{i=1}^t$ .
  - 2: **Output:** Predictive function on  $\mathcal{D}_{T_{t+1}}$
  - 3: **for**  $i$  **in**  $[0, 1, \dots, t]$  **do**
  - 4:   Minimize  $\mathcal{J}(T_0, T_1, \dots, T_{i+1})$  using Eq. (6)
  - 5:   Obtain predictive function  $q_\phi(y|\mathbf{x})$  on  $\mathcal{D}_{T_i}$
  - 6:   Generate pseudo-labels on the unlabeled data in  $\mathcal{D}_{T_i}$  using learned  $q_\phi(y|\mathbf{x})$
  - 7: **end for**
-



Figure 4 provides an overview of our proposed transfer learning framework based on label-informed  $\mathcal{C}$ -divergence. It can be seen that key components to our frameworks are variational auto-encoder and domain discrepancy measure. The intuition of variational auto-encoder used in our framework is as follows: (1) it learns a label-informed latent representation using both data feature and data label in order to estimate the  $\mathcal{C}$ -divergence between source and target domains; (2) it could learn the discriminative classifier  $q(\cdot|\mathbf{x})$  in a semi-supervised manner using knowledge from both labeled source examples and limited labeled target examples as well as adequate unlabeled target examples. Then, the domain discrepancy  $d_{\mathcal{C}}$  could be estimated using the label-informed latent representation from source and target domains such that the minimization of  $\mathcal{C}$ -divergence  $d_{\mathcal{C}}$  enables the better alignment of data distributions across domains. In addition, Figure 4(b)(c) provides the probabilistic graphical model for our recognition (probabilistic encoder) and generation (probabilistic decoder) modules in our framework. It assumes that for probabilistic encoder  $q_{\phi}(\mathbf{x}, y, \mathbf{z}) = q_{\phi}(\mathbf{z}|y, \mathbf{x})q_{\phi}(y|\mathbf{x})q(\mathbf{x})$ , and for probabilistic decoder we have  $p_{\theta}(\mathbf{x}, y, \mathbf{z}) = p_{\theta}(\mathbf{x}|y, \mathbf{z})p_{\theta}(y|\mathbf{z})p(\mathbf{z})$ .

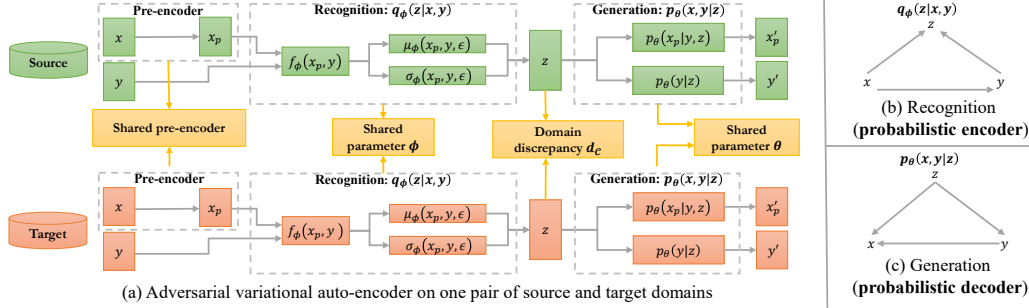


Figure 4: Overview of our proposed transfer learning framework (best viewed in color). (a) Adversarial variational auto-encoder learns domain-invariant hidden representation. (b) and (c) indicate the probabilistic graphical model for our recognition and generation modules.

#### A.4 EXPERIMENTAL DETAILS

We provide the experimental details, including data simulation, model configuration and additional results on digital image data sets. All our experiments are performed on a Windows machine with four 3.80GHz Intel Cores and 64GB RAM.

##### A.4.1 DATA SETS

**Synthetic Data:** Figure 5 provides the synthetic data set with a set of source and target data points where positive and negative samples are randomly sampled from two independent Gaussian distributions  $\mathcal{N}([1.5 \cos \theta, 1.5 \sin \theta]^T, 0.5 \cdot \mathbf{I}_{2 \times 2})$  and  $\mathcal{N}([1.5 \cos(-\theta), 1.5 \sin(-\theta)]^T, 0.5 \cdot \mathbf{I}_{2 \times 2})$ . We let  $\theta = 0$  for source domain (denoted as S1), and then the data points are rotated by setting  $\theta$  as  $\frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}, \pi$  to generate the target domain with time-evolving nature. The data distribution from target domain slightly shifts in each time stamp. Intuitively, it can be observed that source domain S1 has the similar data distribution as the target domain T1, whereas it is significantly different from the target domain T8 (specifically, they have the significantly different conditional distribution  $p(y|x)$  but similar marginal distribution  $p(x)$ ).

**Image Data:** In this paper, we use different methods to generate the time-evolving target domain, e.g., adding adversarial noise to digital images, or adding random salt&pepper noise and rotation to Office-31 and Office-Home, in order to simulate different situations which could lead to a time-evolving target domain in real scenarios.

- Digital images: We used three publicly available data sets: MNIST<sup>4</sup> (with 60,000/10,000 train/test examples), SVHN<sup>5</sup> (with 531,131/26,032 train/test examples) and USPS<sup>6</sup> (with 7,291 / 2,007 train/test examples). In our experiments, we generate the time-evolving target domain by adding the adversarial noise to the clean target image data (e.g. MNIST for

<sup>4</sup><http://yann.lecun.com/exdb/mnist/>

<sup>5</sup><http://ufldl.stanford.edu/housenumbers/>

<sup>6</sup><https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>



Figure 5: Synthetic source and target data (best viewed in color). For source domain (S1 at time stamp 1), positive samples are red ones and negative samples are violet ones. For target domain (T1,  $\dots$ , T8 at time stamp 1~8), positive samples are in blue and negative samples are in green.

transfer learning on SVHN $\rightarrow$ MNIST). The reason why we add the adversarial noise is that it could change the data distribution by adding the adversarial noise such that the generated adversarial examples largely fool the classifier learned on the clean examples. More specifically, we used the Fast Gradient Sign Method (FGSM) Goodfellow et al. (2015) to learn the adversarial noise on the image data sets. The adversarial noise generated by FGSM is defined as follows.

$$\tau = \omega \nabla_{\mathbf{x}} \mathcal{J}_{base}(\theta, \mathbf{x}, y)$$

where  $\omega \geq 0$  is the magnitude of adversarial noise and  $\mathcal{J}_{base}$  is the loss function of a neural network model (parameterized by  $\theta$ ) to be attacked over example  $(\mathbf{x}, y)$ . Here we simply use the pre-trained LeNet<sup>7</sup> model as the base model  $\mathcal{J}_{base}$ . Due to the transferability of adversarial examples, the adversarial examples generated by one model could easily fool another model. Therefore, give one target domain (i.e., MNIST for transfer learning on SVHN $\rightarrow$ MNIST), we can generate new target domain examples by adding the adversarial noise. When the magnitude of adversarial noise  $\omega$  linearly changes from 0.0 to 0.50 with an interval of 0.05, it would generate the evolving target domain examples. Figure 6 shows the image examples of a static source domain (SVHN) and a time evolving target domain (MNIST) for continuous transfer learning. For each time stamp in the target domain, the number of labeled target training examples is set as 100.

- Office-31<sup>8</sup> and Office-Home<sup>9</sup>: For Office-31 and Office-Home, we add the random salt&pepper noise and rotation to create the evolving target domain. To be more specific, given the target domain (e.g., Webcam), we rotate the target images with degree  $O_d$  and add the random salt&pepper noise with magnitude  $O_m$  at every time stamp  $t$  by using the following functions.

$$O_d = 45 \cdot t \quad \text{and} \quad O_m = 0.1 \cdot t \quad \text{where} \quad t = 0, 1, 2, 3, 4$$

Then we obtain a time-evolving target domain (denoted as T1  $\dots$ , T5 in Table 2). In this case, we choose the number of labeled target training examples at each time stamp to be  $\min(\tilde{m}_T, 50)$  where  $\tilde{m}_T$  here is the number of all examples in such a target domain.

<sup>7</sup><https://drive.google.com/drive/folders/1fn83DF14tWmit0RTKWRhPq5uVXt73e0h>

<sup>8</sup><https://people.eecs.berkeley.edu/~jhoffman/domainadapt/>

<sup>9</sup><http://hemantdhv.org/OfficeHome-Dataset/>

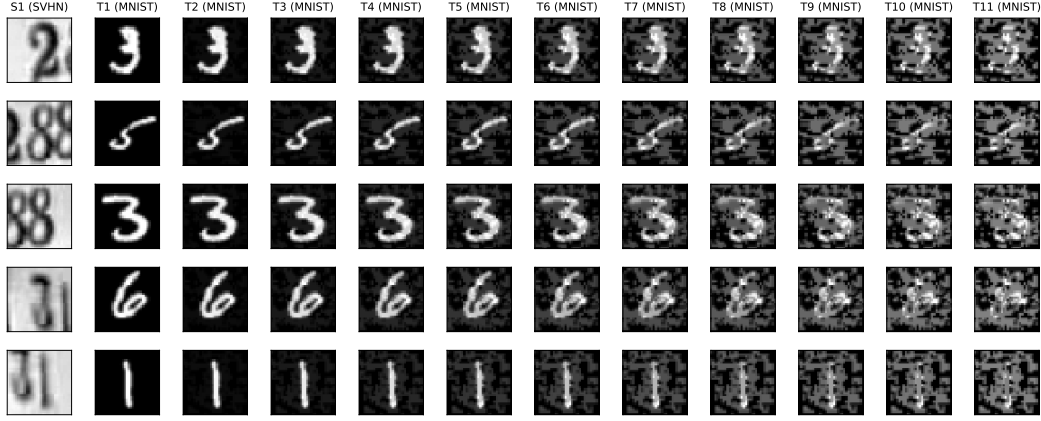


Figure 6: Examples of source domain (SVHN) and time-evolving target domain (MNIST). The first column is the source image examples in SVHN data set. The other columns are the target image examples from MNIST data set with different magnitude of adversarial noise.

#### A.4.2 MODEL CONFIGURATION

For digital image classification, the neural network architecture used in our experiments is shown in Figure 7 where we used the gradient reversal layer (GRL) (Ganin et al., 2016) to implement our proposed  $\mathcal{C}$ -divergence in the latent space. All the model parameters will be optimized from the scratch. We apply the Stochastic Gradient Descent (SGD) with the momentum of 0.9 to train our model where all the hidden parameters are initialized with Xavier initialization. The cross-entropy loss is adopted to measure the loss of label prediction and domain prediction. Following (Ganin et al., 2016), the learning rate  $\eta_p$  is adjusted when training the model:  $\eta_p = \frac{\eta_0}{(1+\alpha p)^\beta}$  where  $p$  is an epoch-dependent scalar linearly varying from 0 to 1, and  $\eta_0 = 0.01$ ,  $\alpha = 10$ ,  $\beta = 0.75$ . The total number of training epochs is 10,000 with batch size 32 in our experiments. The domain adaptation parameter in gradient reversal layer is given by:  $\lambda_p = \frac{2}{1+\exp(-\gamma p)} - 1$  where  $\gamma = 10$ .

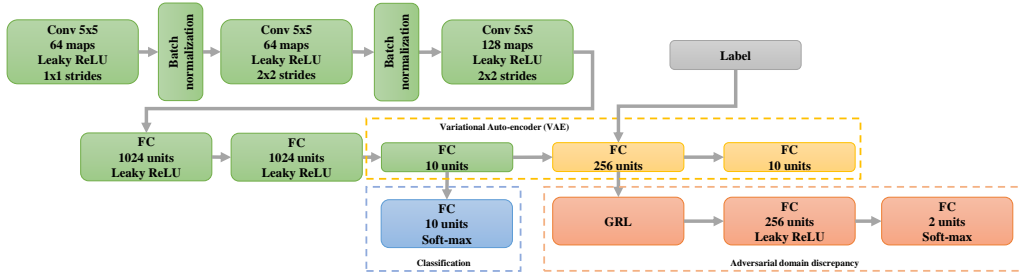


Figure 7: Neural network architecture used in our experiments. If data point is labeled, its class  $y$  is used, otherwise, it uses the class prediction as a pseudo-label for learning the latent representation in the Variational Auto-encoder (VAE) framework. We applied the gradient reversal layer (GRL) Ganin et al. (2016) to implement the adversarial domain discrepancy.

For Office-31 and Office-Home, we adopt ResNet-50 (He et al., 2016) pretrained on ImageNet as base network for feature extraction. In this case, the initial learning rate  $\eta_0$  on fine-tuning the ResNet parameters is 0.001 while other model parameters use the initial learning rate  $\eta_0 = 0.1$ . The total number of training epochs is 5,000 with batch size 20 for Office-31 and Office-Home.

#### A.4.3 EXPERIMENTAL RESULTS

**Evaluation of Continuous Transfer Learning:** Table 4 shows the continuous transfer learning results from MNIST to USPS when using adversarial attacks to generate the evolving target domain. The results are consistent with our observations in Section 6.3. It is observed that (1) source and historical target knowledge could largely improve the classification performance on the evolving target domain; (2) static transfer learning baselines might produce worse classification performance than TargetERM, thus leading to the occurrence of negative transfer when data distribution between

source and current target tasks are largely shifted for T6-T11. Table 5 shows the continuous transfer learning results on Office-Home. It confirms the effectiveness of our proposed CONTE algorithm.

Table 4: Transfer learning accuracy from MNIST (source) to continuously evolving USPS (target)

Target Domain	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
SourceOnly	0.8196	0.7778	0.6946	0.5745	0.3921	0.2272	0.1579	0.0907	0.0613	0.0429	0.0289
TargetERM	0.8012	0.7474	0.6951	0.6557	0.6253	0.6412	0.7205	0.7384	0.7693	0.7828	0.8381
CORAL	0.8570	0.8211	0.7897	0.7195	0.7240	0.6288	0.6323	0.6831	0.6313	0.6139	0.6632
DANN	0.9088	0.8774	0.8411	0.8037	0.7633	0.7389	0.7260	0.6413	0.6986	0.7688	0.7997
ADDA	0.9098	0.8859	0.8540	0.8012	0.7210	0.5835	0.4509	0.4434	0.4245	0.4410	0.4808
WDGRL	0.9133	0.8485	0.8510	0.8067	0.7793	0.7195	0.7559	0.7369	0.8127	0.8052	0.8062
DIFA	0.8680	0.8361	0.8122	0.7683	0.7140	0.6163	0.4295	0.3687	0.4559	0.3627	0.4425
CONTE <sub>∞</sub>	<b>0.9482</b>	<b>0.9382</b>	0.9178	0.9048	0.8839	0.8739	0.8037	0.8640	0.8804	0.9093	0.8585
CONTE	<b>0.9482</b>	0.9367	<b>0.9357</b>	<b>0.9502</b>	<b>0.9606</b>	<b>0.9586</b>	<b>0.9567</b>	<b>0.9636</b>	<b>0.9681</b>	<b>0.9711</b>	<b>0.9706</b>

Table 5: Transfer learning accuracy on Office-Home

	Art → Real World					Clipart → Product				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
SourceOnly	0.7220	0.3947	0.3135	0.2650	0.3512	0.5944	0.1866	0.1342	0.1074	0.1550
TargetERM	0.5643	0.3297	0.3010	0.2582	0.3299	0.6033	0.3805	0.4232	0.3061	0.3791
DAN	0.7341	0.4901	0.4193	0.3686	0.4597	0.7186	0.4201	0.3921	0.3352	0.4113
CORAL	0.7321	0.5072	0.4215	0.3809	0.4577	0.7169	0.4350	0.3764	0.3189	0.3893
DANN	0.7359	0.5092	0.4155	0.3850	0.4686	0.7063	0.4440	0.3694	0.3343	0.4303
MDD	0.7435	0.5056	0.4331	0.3874	0.4686	0.7264	0.4765	0.3886	0.3514	0.4294
CONTE <sub>∞</sub>	0.7560	0.5273	0.4575	0.4080	0.4850	<b>0.7411</b>	0.5017	0.4436	0.3634	0.4595
CONTE	<b>0.7560</b>	<b>0.6046</b>	<b>0.5447</b>	<b>0.5097</b>	<b>0.5459</b>	<b>0.7411</b>	<b>0.5747</b>	<b>0.5318</b>	<b>0.5009</b>	<b>0.5422</b>

**Effect of limited label information in the target domain:** We evaluate the effect of limited label information in the target domain on mitigating the negative transfer in the static transfer learning problem. When no label information is available in the target domain, it would be difficult to characterize and avoid the negative transfer. Figure 8 shows the classification performance of transfer learning algorithms from SVHN (source) to MNIST (target) where “w/” indicates “with limited label information in the target domain” (semi-supervised transfer learning) and “w/o” indicates “without any label information in the target domain” (unsupervised transfer learning). It can be seen that without any target label information, negative transfer is more likely to occur for transfer learning algorithms. It demonstrates that limited label information in the target domain is necessary to characterize the negative transfer.

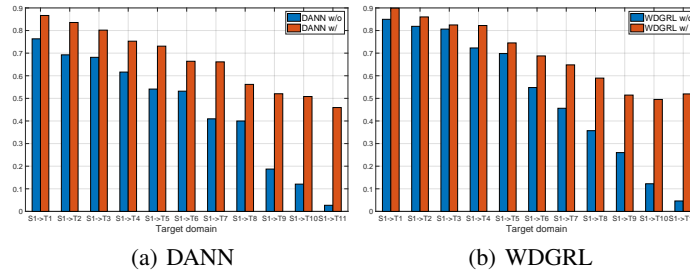


Figure 8: Transfer learning accuracy with or without limited label information in the target domain

**Effect of  $\mathcal{C}$ -divergence:** We empirically compare the proposed  $\mathcal{C}$ -divergence with unsupervised domain divergence in (Ganin et al., 2016) on the synthetic data set (shown in Figure 5). To be more specific, we implement a simple domain-adversarial neural network (Ganin et al., 2016) with either unsupervised domain divergence or our  $\mathcal{C}$ -divergence, and consider the following three algorithms. DANN<sub>un</sub>: proposed in (Ganin et al., 2016) with unsupervised domain divergence (no labeled target examples are available); DANN<sub>semi</sub>: a variant of DANN<sub>un</sub> with unsupervised domain divergence, but with limited labeled target examples for minimizing the classification error; DANN<sub>C</sub><sub>semi</sub>: a

variant of DANN\_un with our proposed  $\mathcal{C}$ -divergence and limited labeled target examples could help both minimize the classification error and label-informed distribution alignment. Figure 9 shows the transfer learning performance from the source (S1) to the target T4, T5 and T6, respectively. With limited target examples, DANN\_semi could largely avoid the negative transfer compared to DANN\_un. That confirms the effect of limited label target information for transfer learning. One intuitive explanation is that T5 and T6 (see Figure 5 for Target domain #5 and #6) are more likely to be aligned incorrectly with the source domain when no label information in the target domain is available. Limited target label information helps mitigate the occurrence of negative transfer in this case. Moreover, our proposed  $\mathcal{C}$ -divergence could help improve the transfer learning performance and avoid the negative transfer by encouraging the alignment of label-informed data distribution.

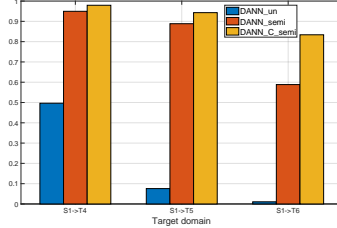
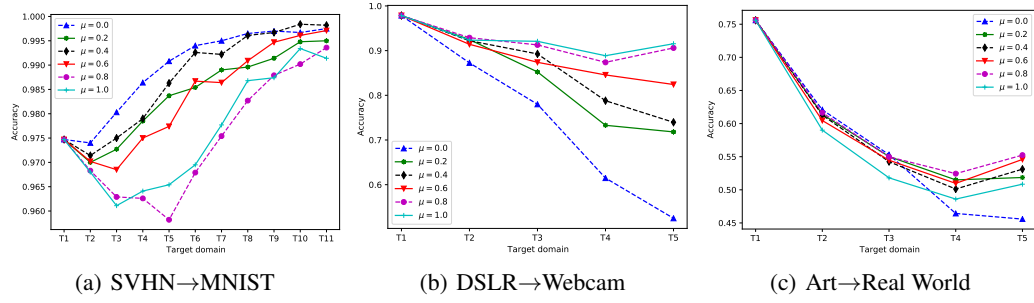
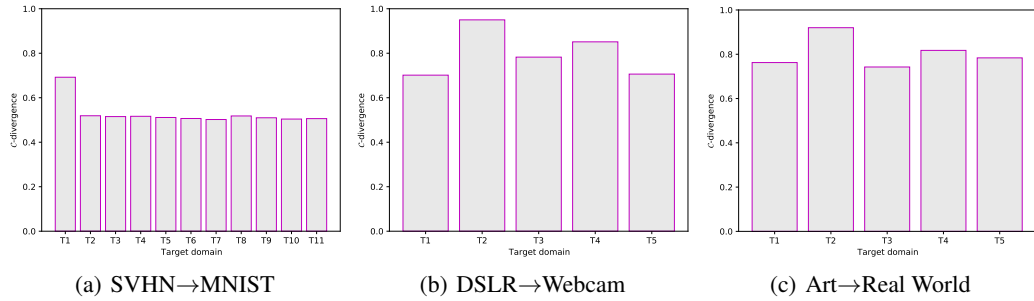


Figure 9: Effect of  $\mathcal{C}$ -divergence

**Effect of hyperparameter  $\mu$ :** We empirically investigate the effect of hyperparameter  $\mu$  in our proposed framework. Figure 10 shows the classification results of our algorithm with different values of  $\mu$  on digital (SVHN→MNIST), Office-31 (DSLRL→Webcam) and Office-Home (Art→Real World) data sets. We have the following observations: (1) Generally, the classification performance on digital (SVHN→MNIST) becomes better over time, whereas it becomes worse over time on Office-31 (DSLRL→Webcam) and Office-Home (Art→Real World); (2) Our algorithm obtains the best performance with  $\mu = 0.0$ ,  $\mu = 1.0$  and  $\mu = 0.8$  on digital, Office-31 and Office-Home data sets, respectively. Furthermore, the optimal value  $\mu = 0.0$  on digital data implies that the target classification performance of  $\mathcal{D}_{t+1}$  is more likely to rely on the most recent target data  $\mathcal{D}_t$ . On the other hand, large  $\mu$  indicates that it largely requires all the source and historical target knowledge to improve the target classification performance of  $\mathcal{D}_{t+1}$  on Office-31 and Office-Home. One explanation is that when the target data distribution shifts smoothly, the closest (most recent) target domain data could provide the most useful relevant information when learning the current target predictive function, otherwise, it is more likely to gather all the source and historical target domain knowledge to help improve the current target predictive function. In Figure 11, we visualize the estimated  $\mathcal{C}$ -divergence<sup>10</sup> within two consecutive time stamps in the target domain on those data sets (e.g., the 'T3' in the x-axis represents the  $\mathcal{C}$ -divergence between T2 and T3). It confirms that the  $\mathcal{C}$ -divergence increases very smoothly from T1 to T11 on digital (SVHN→MNIST), but it increases significantly from T1 to T2 on both Office-31 and Office-Home.

<sup>10</sup>In this case, (1) we show the label-informed domain classification accuracy ( $\geq 0.5$ ) because higher domain accuracy implies larger domain discrepancy across domains; (2) we use the ground truth of target examples to estimate the  $\mathcal{C}$ -divergence in order to accurately identify how the target distribution evolves.

Figure 10: Effect of  $\mu$  on digital, Office-31 and Office-HomeFigure 11: Estimate of  $\mathcal{C}$ -divergence on digital, Office-31 and Office-Home