Continual Domain Adversarial Adaptation via Double-Head Discriminators

Yan Shen **Zhanghexuan Ji** **Chunwei Ma** **Mingchen Gao**
Department of Computer Science and Engineering, University at Buffalo

Abstract

Domain adversarial adaptation in a continual setting poses significant challenges due to the limitations of accessing previous source domain data. Despite extensive research in continual learning, adversarial adaptation cannot be effectively accomplished using only a small number of stored source domain data, a standard setting in memory replay approaches. This limitation arises from the erroneous empirical estimation of $\mathcal{H}$-divergence with few source domain samples. To tackle this problem, we propose a double-head discriminator algorithm by introducing an additional source-only domain discriminator trained solely on the source learning phase. We prove that by introducing a pre-trained source-only domain discriminator, the empirical estimation error of $\mathcal{H}$-divergence related adversarial loss is reduced from the source domain side. Further experiments on existing domain adaptation benchmarks show that our proposed algorithm achieves more than 2% improvement on all categories of target domain adaptation tasks while significantly mitigating the forgetting of the source domain.

1 INTRODUCTION

Unsupervised Domain adaptation (UDA) refers to the process of transferring knowledge from a labeled source domain to an unlabeled target domain (Ben-David et al., 2010; Zhao et al., 2019), taking into account the presence of *domain shifts* between the source and target domains. One line of UDA work to bridge the domain gap focuses on learning domain invariant feature representations by adversarial adaptations (Ganin et al., 2016; Zhao et al., 2018; Long et al., 2017; Zhang et al., 2019). Classic adversarial domain adaptation applies in offline settings where both the source and target domain data can be accessed, assuming they follow an i.i.d. distribution. However, domain data is accessed sequentially in continual learning (CL). The sequential nature of CL complicates the direct application of these conventional approaches.

Intuitively, one can expect that the gap between offline and online learning would be partly bridged if a small portion of the previous domain data is stored and subsequently accessible. This 'divide-and-conquer' idea has brought up to a setting known as *memory replay continual learning* where the learner stores a small portion of previous tasks in memory and replays them with the new mini-batch data. However, different from memory replay CL in supervised task (Belouadah and Popescu, 2019; Zhao et al., 2020; Hou et al., 2019; Castro et al., 2018; Wu et al., 2019; Liu et al., 2021), adversarial adaptation requires estimation of an extra domain discrepancy term, as the $\mathcal{H}$-divergence, in addition to the supervised task risk on the previous source domain. Prior theoretical results show that empirically estimating $\mathcal{H}$-divergence using only a few source samples results in a significant error gap from the source side (Ben-David et al., 2010). Consequently, the model adversarially trained on a small number of stored source samples, would exhibit poorer performance in target adaptation.

In light of the above unique challenge in adversarial adaptation under CL settings, to construct a low-error empirical estimation of domain discrepancy with a small number of source samples, we propose our **double-head discriminator algorithm**. We train two domain discriminators on domain data of different phases. One is trained in the source learning phase as *source-only domain discriminator*. The other one is adversarially trained in the target adaptation phase with a task model. And we employ the ensemble of two domain discriminators to achieve a more accurate estimation of the empirical error with $\mathcal{H}$-divergence. In particular, the source-only domain discriminator is

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

trained exclusively with source domain data in one-class learning approaches. It serves as a score-based function to assess the level of in-distribution within the source domain. In the target adaption phase, the source-only domain discriminator is frozen. The ensembles of two domain discriminator's digits are used as $\mathcal{H}$-divergence signal to learn a domain generalized task model.

Our contributions are summarised as follows.

(i) We propose a double-head discriminator algorithm tailored for adversarial adaptation in a CL setting. Different from existing works on continual UDA, our algorithm learns a domain-generalized task model with better performance on target domain tasks while mitigating the issue of catastrophic forgetting on the tasks of previous source domains. Our proposed algorithm is effective, requiring only a few source domain samples stored in the replay memory buffer.

(ii) We theoretically analyze our proposed algorithm. Firstly, we show that the population form of two discriminator's ensemble digits does construct a $\mathcal{H}$-divergence to bound on the generalization error between the source and target domain's population risk. Next, we demonstrate that in empirical form, the ensemble of two discriminators reduces the error of empirical estimation on $\mathcal{H}$-divergence from the source domain side. Finally, we analyze the equilibrium of our adversarial loss on how source only domain discriminator regulates the source and target domain's distributions.

(iii) Empirically, we show that our algorithm consistently performs better on continual adaptation to target domain tasks while significantly mitigating the issue of catastrophic forgetting on previous source domain tasks.

2 RELATED WORKS

Unsupervised Domain Adaptation For UDA methods, besides adversarial domain adaptation (Ganin et al., 2016; Zhao et al., 2018; Long et al., 2017; Zhang et al., 2019; Saito et al., 2018) that learns feature representations invariant between source and target domain, self-training (ST) and knowledge distillation (KD) are also widely adopted for UDA (De Lange et al., 2021). ST (Arazo et al., 2020; Pham et al., 2021) trains on the supervised task on the pseudo-labels that are iteratively assigned to unlabeled target domain data (Lee et al., 2013; Yarowsky, 1995; Nigam and Ghani, 2000). As ST gradually shifts the decision boundary in feature space from the one separating the class-conditional distributions of the previous source domain to the one separating the class-conditional distributions of the new target domain (Kumar et al., 2020; Wang et al., 2022b), ST adapts the model from a source domain specific model to target domain special model. The model's discriminative ability on the source domain data will be lost after continual adaptation. KD (Liang et al., 2022; WU SJ, 2019; Dhar et al., 2019; Douillard et al., 2020; Zhu et al., 2021) partially maintains the class-conditional distributions of the previous source domain in its newly learned class-conditional distributions of the new target domain. However, there is a trade-off between the network's ability to adapt to new target domain data and to adhere to the previous source domain data. Other works (Ding et al., 2022; Fleuret et al., 2021; Dey et al., 2022; Kundu et al., 2020; Li et al., 2020; Tian et al., 2021; Wang et al., 2022a; Xia et al., 2021; Yang et al., 2021; Yeh et al., 2021; Gong et al., 2022; Niu et al., 2022; Ma et al., 2022) also address the problem of UDA. However, these works require either freezing on the task model trained on the source domain, caching the prototypical features of the source domain, or demanding specific engineering on the task model structure, which limits its application in the general setting of CL. Notably, federated UDA (Shen et al., 2023; Peng et al., 2019) proposes a simplified version of CL where all domain datasets are simultaneously accessible in a spatially isolated case.

Domain Incremental Learning The main goal for domain incremental learning is to consistently learn information on a new domain, without forgetting the knowledge of previous domains. The first category of methods is by incrementally adding new task heads to fit on new domains (Rusu et al., 2016; Zhou et al., 2012; Cortes et al., 2017; Yoon et al., 2018; Ji et al., 2023). The second category of methods is using memory replay methods to store the data of previous domains (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2018; Dokania et al., 2019; Riemer et al., 2018; Hayes et al., 2020; Prabhu et al., 2020). The third category of methods is to add regularization terms to constraint task's objectives to avoid forgetting (Li and Hoiem, 2017; Kirkpatrick et al., 2017; Zenke et al., 2017; Fini et al., 2020; Volpi et al., 2021; Rostami, 2021). Apart from these three categories of methods, contrastive learning (Tang et al., 2021), meta learning (Volpi et al., 2021; Sankaranarayanan and Balaji, 2023; Qin et al., 2023; Wang et al., 2022e) and domain randomization (Volpi et al., 2021; Wang et al., 2022d,c, 2023) are also implemented for DIL for model robustness and generalizations on varied domain distributions.

Causes of Catastrophic Forgetting Generally speaking, the catastrophic forgetting on the previous domain comes from two sources: (A) the feature misalignment on new domain with previous domains in the feature embedding space (McCloskey and Cohen, 1989; Goodfellow et al., 2013; Kemker et al., 2018; Xian et al.,

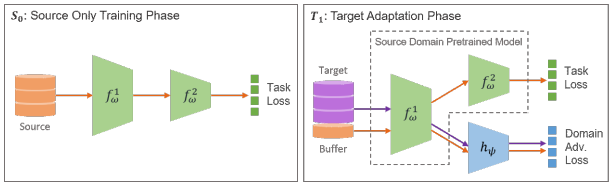


Figure 1: A continual adversarial domain adaptation model. Only the source risk of the client's local source data is accessible in source only training phase. A small set of buffered source domain data and target domain data is adversarial trained in target adaptation phase.

2021), occurs when excessive parameter adjustments in the feature extractor disrupt the representation of previously encountered classes, resulting in compromised feature extraction and task prediction accuracy. (B) the shift of decision boundaries of the task model when adapting to new domain data (Dong et al., 2022; Wang et al., 2022b; Kumar et al., 2020), emerges when the classifier becomes overly specialized to the class distributions of current domains, introducing biases that hinder its ability to distinguish between new and old class boundaries.

3 PRELIMINARY

Continual Unsupervised Domain Adaptation In Continual UDA, the data comes as a stream S_0, T_1 and a unified model is trained on current data locally without revisiting previous data. Let $P = \{S_0, T_1\}$ be the data stream, in which $S_0 = \{(\boldsymbol{x}_i^s, y_i^s)\}$ contains labeled examples in source domain, $T_1 = \{(\boldsymbol{x}_i^t)\}$ contains unlabeled examples in target domain, where $\boldsymbol{x}_i^s, \boldsymbol{x}_i^t \in \mathbb{D}, y_i^s \in \mathcal{C}$. Specifically, the continual domain adaptation algorithm begins with training a task model f_w on labeled examples from source domain S. For the successive phase, the data comes as unlabeled examples on target domains T. We name the two phases as source training phase S_0 and target adaptation phase T_1. The unified task model $f_\omega = f_\omega^2 \circ f_\omega^1$ consists of a *feature extractor* f_ω^1 and a *label predictor* f_ω^2. The feature extractor is a deep neural network $\boldsymbol{z} = f_\omega^1(\boldsymbol{x}), \boldsymbol{z} \in \mathbb{F}$ that maps the data to feature space. Continual domain adaptation aims to learn a feature extractor f_ω^1 that generates domain invariant feature representations. The label predictor is another network $\boldsymbol{y} = f_\omega^2(\boldsymbol{z}), \boldsymbol{y} \in \mathbb{R}^{|\mathcal{C}|}$ maps from feature space to task digits space. Both feature extractor and label predictor are trained continuously at both S_0 and T_1 phases. In addition to the task model, the trained model also involves another *domain discriminator* network $d = h_\psi(\boldsymbol{z}), d \in \mathbb{R}$ that tries to determine whether the extracted features belong to the source or target domain.

Target Domain Adaptation with Few Stored Source Samples is Challenging Memory replay in continual learning methods refers to storing a small number of samples from previous domains and replaying them alongside the current data stream in mini-batches while learning new domain data. We denote memory buffer as $\mathcal{M}$ that stores a small portion of previously accessed domain data. With the incorporation of replay memory buffer $\mathcal{M}$, it optimizes on a empirical task loss of the joint distribution of the current data stream and the replay memory $\mathcal{M}$ (Chaudhry et al., 2019). In traditional supervised CL settings, the empirical task loss on $\mathcal{M}$ is trained purely for memorizing the old task. Uniquely in adversarial adaptation, the new task objective of target adaptation on T_1 takes the general form of an empirical domain adversarial loss on the joint distribution of target domain data in T_1 and stored source domain data in $\mathcal{M}$ (illustrated in Fig(1)), as follows:

$$\min_{\omega} \max_{\psi} \mathbb{E}_{(\mathbf{x}_i^s, y_i^s)\sim\mathcal{M}}[\ell(f_\omega(\mathbf{x}_i^s), y_i^s) - \nu\mathbb{E}_{(\mathbf{x}_i^s, y_i^s)\sim\mathcal{M}} D_\psi^s(\mathbf{x}_i^s)] - \nu\mathbb{E}_{\mathbf{x}_i^t\sim T_1} D_\psi^t(\mathbf{x}_i^t) \tag{1}$$

As a part of new adversarial adaptation task on target domain at T_1, the min-max objective $\mathbb{E}_{(\mathbf{x}_i^s, y_i^s)\sim\mathcal{M}} D_\psi^s(\mathbf{x}_i^s) + \mathbb{E}_{\mathbf{x}_i^t\sim T_1} D_\psi^t(\mathbf{x}_i^t)$ is related to an empirical estimation of the $\mathcal{H}$-divergence. For a more detailed introduction, we refer interested readers to Appendix (A). However, according to Theorem 1 given by Ben-David et al. (Ben-David et al., 2010), using few samples of stored source domains data to construct an empirical version of $\mathcal{H}$-divergence, denoted as $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$, can result in significant errors when estimating the population $\mathcal{H}$-divergence.

Theorem 1. *Let $\mathcal{F}$ be a hypothesis space with VC dimensions d, if S' are samples of size m from S and T' are samples of size n from T respectively and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S', T')$ is the empirical $\mathcal{H}$-divergence between samples, then for any $\delta \in (0, 1)$, with probability at lease $1 - \delta$*

$$d_{\mathcal{H}\Delta\mathcal{H}}(S, T) \leq \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S', T') + 2\sqrt{\frac{d\log 2m + \log(2/\delta)}{m}} + 2\sqrt{\frac{d\log 2n + \log(2/\delta)}{2n}} \tag{2}$$

Due to the erroneous estimation of the objective function, the adversarial adaptation task on the target domain at T_1 is expected to exhibit poor performance.

4 METHODOLOGY

4.1 Double head Domain Discriminator For Continual UDA

To compensate for the erroneous empirical estimation of $\mathcal{H}$-divergence originating from a small number of source domain samples, our natural idea is to introduce an additional domain discriminator trained on the full set of source domain data instead of a tiny set in the memory buffer. In the specific problem setting of continual UDA, the auxiliary domain discriminator is trained on S_0 phase and then frozen during T_1 phase. Since only the source domain data is accessible in the S_0 phase, the auxiliary domain discriminator we introduced in S_0 phase is *source-only domain discriminator*. Extending the general loss function of $\mathcal{H}$-divergence to single-side (source) domain loss leads to the following form

$$\begin{aligned}\hat{d}_{\mathcal{H}\Delta\mathcal{H}} \triangleq sup_{\psi}[&\mathbb{E}_{\mathbf{x}_i^s \in S_0} D(\sigma(h_{\psi,s}(f_\omega^1(\mathbf{x}_i^s))))\\ &- \mathbb{E}_{\mathbf{x}_i^s \notin S_0} D(\sigma(h_{\psi,s}(f_\omega^1(\mathbf{x}_i^t))))]\end{aligned} \tag{3}$$

The above training objective $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ has a similar problem formulation of one-class learning. Specifically, the training data $\mathbf{x}_i \in S$ is treated as a one-class distribution. And a score based function $\sigma(h_{\psi,s} \circ f_\omega^1)(\mathbf{x}) \in [0,1]$ is trained to determine how possible that a data instance $\mathbf{x}$ lies within the distribution of training dataset S_0 (source domain). However, one-class learning doesn't learn a boundary as distinguishable as a multi-class classification model. An ideal one-class score function should exhibit positive correlations on its score with data points that belong to the in-distribution and have higher densities.

A deep one-class learning problem is a class of challenging tasks still under active research. we will describe our solution in the specific case of source-only domain classifier in Section (4.2).

In the remaining part of this section, we will describe how we utilize the two complementary domain discriminators jointly to learn a domain generalized task model in the target adaptation phase T_1. The task model f_ω and source-only domain discriminator $h_{\psi,s}$ are firstly trained on the source domain task. Then the pre-trained source-only domain discriminator $h_{\psi,s}$ is frozen in the successive T_1 phase. We introduced another target adaptation discriminator $h_{\psi,t}$ that is adversarial trained with feature generator f_ω^1 during T_1 phase. The target adaptation discriminator is trained discriminatively using the features from the source domain memory buffer $\mathcal{M}$ and the target domain data in T_1 with the commonly used cross-entropy loss:

$$\begin{aligned}&D_{\psi,t}^s(\mathbf{x}_i^s) = -\log(\sigma(h_{\psi,t}(f_\omega^1(\mathbf{x}_i^s)))),\\ &D_{\psi,t}^t(\mathbf{x}_i^t) = -\log(1-\sigma(h_{\psi,t}(f_\omega^1(\mathbf{x}_i^t))))\\ &\min_{\psi_t}[\mathbb{E}_{\mathbf{x}_i^s \sim \mathcal{M}} D_{\psi,t}^s(\mathbf{x}_i^s) + \mathbb{E}_{\mathbf{x}_i^t \sim T_1} D_{\psi,t}^t(\mathbf{x}_i^t)]\end{aligned} \tag{4}$$

To learn domain-independent feature representations, the feature extractor f_ω^1 is trained adversarially with the target domain discriminator $h_{\psi,t}$. The estimated $\mathcal{H}$-divergence from the domain discriminator is used as a signal to guide the learning of domain-invariant feature representations. Instead of solely relying on the target domain discriminator $h_{\psi,t}$ that is trained with only a small number of samples of source domain data in $\mathcal{M}$, we utilize the ensembles of source and target domain discriminator outputs to obtain a lower empirical estimation of the $\mathcal{H}$-divergence between the distributions of the source and target domains. The adversarial loss function for learning feature extractor f_ω^1 with respect to $\mathcal{H}$-divergence is given by:

$$\begin{aligned}D_\psi^s(\mathbf{x}_i^s) &= -\log(\sigma(h_{\psi,s}(f_\omega^1(\mathbf{x}_i^s)) + h_{\psi,t}(f_\omega^1(\mathbf{x}_i^s)))\\ D_\psi^t(\mathbf{x}_i^t) &= -\log(1-\sigma(h_{\psi,s}(f_\omega^1(\mathbf{x}_i^t)) + h_{\psi,t}(f_\omega^1(\mathbf{x}_i^t))))\end{aligned} \tag{5}$$

With the previously mentioned loss function for $\mathcal{H}$-divergence, the joint learning objective for the task model $f_\omega(\cdot)$ during the target adaptation phase T_1 can be expressed as follows:

$$\begin{aligned}&\min_{\omega} \mathbb{E}_{(\mathbf{x}_i^s, y_i^s)\sim\mathcal{M}}[\ell(f_\omega(\mathbf{x}_i^s), y_i^s) - \nu D_\psi^s((\mathbf{x}_i^s))]\\ &-\nu\mathbb{E}_{\mathbf{x}_i^t \sim T_1} D_\psi^t((\mathbf{x}_i^t))\end{aligned} \tag{6}$$

The entire diagram for Continual UDA with our double-head discriminator algorithm is illustrated in Fig. 2.

4.2 Example for Single Domain Discriminator Learning: Margin Disparity Discrepancy

A straightforward way for one-class learning of source-only domain discriminator $h_{\psi,s}$ in (3) is optimizing on commonly used cross-entropy function on a single class

$$\min_{\psi_s} \mathbb{E}_{\mathbf{x}_i^s \sim S}[-\log(\sigma(h_{\psi,s}(f_\omega^1(\mathbf{x}_i^s))))] \tag{7}$$

However, directly training on the above objective function would limit the trained source-only domain discriminator's ability as a score-based function on the in-distribution of the source domain. One reason is the uncontrollable digit outputs. The other reason is the biased features towards the highest neuron activations.

One way to address the above limitations of one-class learning is by adding an $\mathcal{H}$-Regularization loss as in

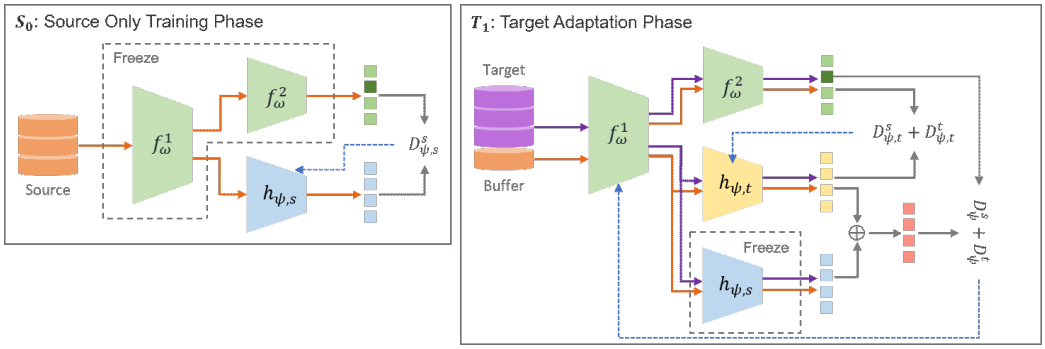


Figure 2: The flowchart of our proposed double-head discriminator algorithm. The solid line is the forward path. And the dashed line is the backward training path. After the task model is trained in source domain, an additional source-only domain discriminator $h_{\psi,s}$ is trained by freezing the task model f_ω. In the target adaptation phase, $h_{\psi,t}$ is adversarially trained with f^1_ω on domain adversarial loss, where the ensembles of domain discriminator $h_{\psi,s}$ and $h_{\psi,t}$'s digit is used as domain adversarial signal to learn domain invariant features for f^1_ω
.

(Hu et al., 2020). This approach is called HRN and applies to general settings of positive, unlabeled learning and continual learning. We will introduce and discuss the HRN method for continual UDA in our appendix. However, by utilizing the specific problem structure in UDA, a more effective score-based function for domain discrepancy is utilizing the margins between classification spaces as proposed in Margin Disparity Discrepancy (MDD)(Zhang et al., 2019). Instead of using a binary domain discriminator of scalar outputs $h_\psi(\cdot) : \mathbb{F} \to \mathbb{R}$, MDD introduces a multi-class domain discriminator of vector outputs $h_\psi(\cdot) : \mathbb{F} \to \mathbb{R}^{|\mathcal{C}|}$. The *margin disparity* from the hypothesis of task model f_ω to $h_\psi \circ f^1_\omega$ is used as the score-based function to measure whether a data instance $\mathbf{x}$ lies within the source domain distribution

Definition 4.1 (Margin Disparity Discrepancy (Zhang et al., 2019))**.** The margin disparity discrepancy is defined as a $\mathcal{H}$-divergence between source and target domains.

$$d^\rho_{f,\mathcal{H}}(S,T) \triangleq \sup_{f' \in \mathcal{H}} (disp^\rho_S(f', f) - disp^\rho_T(f', f)) \quad (8)$$

where $disp^\rho_D(f', f)$ is defined as the margin disparity between f and f' in domain D

$$disp^\rho_D(f, f') \triangleq \mathbb{E}_{\mathbf{x}_i, \sim D} \Phi^\rho(\rho_{f'}(\mathbf{x}_i, h_f(\mathbf{x}_i))) \quad (9)$$

where ρ_f, $h_f(\mathbf{x}_i)$ and Φ^ρ is defined as

$$\rho_f(\mathbf{x}_i, c) \triangleq f(\mathbf{x}_i, c) - \max_{c' \neq c} f(\mathbf{x}_i, c') \quad (10)$$

$$h_f(\mathbf{x}_i) \triangleq \arg\max_c f(\mathbf{x}_i, c) \quad (11)$$

$$\Phi^\rho(x) = \begin{cases} 1 & \text{if } x < 0 \\ 1 - x/\rho & \text{if } 0 \leq x \leq \rho \\ 0 & \text{if } x > \rho \end{cases} \quad (12)$$

Again, with the commonly used cross-entropy loss in training objectives, $\mathcal{H}$-divergence of $d^\rho_{f,\mathcal{H}}(S,T)$ is approximated as

$$\begin{aligned} d^\rho_{f,\mathcal{H}}(S,T) \approx \mathbb{E}_S \log(\text{softmax}(h_\psi(f^1_\omega(\mathbf{x}^s_i)), h_f(\mathbf{x}^s_i))) \\ - \mathbb{E}_T \log(1 - \text{softmax}(h_\psi(f^1_\omega(\mathbf{x}^t_i)), h_f(\mathbf{x}^s_i))) \end{aligned} \quad (13)$$

The MDD-induced training objective for source-only domain discriminator ψ_s in Equation (3) results in

$$\min_{\psi_s} \mathbb{E}_{\mathbf{x}^s_i \sim S_0} -\log(\text{softmax}(h_{\psi,s}(f^1_\omega(\mathbf{x}^s_i)), \arg\max_c f(\mathbf{x}^s_i))) \quad (14)$$

The MDD form of adversarial loss for feature extractor f^1_ω from the ensembles of source and target domain discriminator, as expressed in Equation (5), is given by:

$$\begin{aligned} D^s_\psi(\mathbf{x}^s_i) = -\log(\text{softmax}(h_{\psi,s}(f^1_\omega(\mathbf{x}^s_i)) + h_{\psi,t}(f^1_\omega(\mathbf{x}^s_i)) \\ , \arg\max_c f(\mathbf{x}^s_i))) \\ D^t_\psi(\mathbf{x}^t_i) = -\log(1 - \text{softmax}(h_{\psi,s}(f^1_\omega(\mathbf{x}^t_i)) + h_{\psi,t}(f^1_\omega(\mathbf{x}^t_i)) \\ , \arg\max_c f(\mathbf{x}^t_i))) \end{aligned} \quad (15)$$

The full description of our double-head domain discriminator algorithm for continual UDA is shown in Algorithm 1 of our appendix.

5 THEORETICAL ANALYSIS

In this section, we relate the source-only domain discriminator $h_{\psi,s}(\cdot)$, which is trained on source domain data and frozen during T_1, to a fixed hypothesis f_0. Thus, we study its effect on target adaptation in T_1.

First, we show that in the population form, our domain adversarial function from the ensembles of two discriminator's digit constructs a $\mathcal{H}$-divergence as the generalization upper bound between source and target domain task's population risks.

Theorem 2. *For a hypothesis class $\mathcal{F}$ and a fixed $f_0 \in \mathcal{F}$ where for every $f \in \mathcal{F}$, $f - f_0$ is also in $\mathcal{F}$, then we have the following property holds*

$$err_T(f) \leq err_S^{(\rho)}(f) + d_{f,f_0,\mathcal{F}}^{(\rho)}(S,T) + \lambda \tag{16}$$

where $err_S^{(\rho)}(f)$, $d_{f,f_0,\mathcal{F}}^{(\rho)}(S,T)$ and λ is defined as

$$\begin{aligned}
&err_S^{(\rho)}(f) = \mathbb{E}_{(x_i,y_i)\sim S}\Phi_\rho \circ \rho_f(x_i,y_i)\\
&d_{f,f_0,\mathcal{F}}^{(\rho)}(S,T) = \sup_{f'\in\mathcal{F}}\{\mathbb{E}_{x_i\sim T}\Phi_\rho \circ \rho_{f'+f_0}(x_i,h_f(x_i))\\
&-\mathbb{E}_{x_i\sim S}\Phi_\rho \circ \rho_{f'+f_0}(x_i,h_f(x_i))\}\\
&\lambda = \min_{f^\star\in\mathcal{F}} err_S^{(\rho)}(f^\star) + err_T^{(\rho)}(f^\star),
\end{aligned} \tag{17}$$

Remark The upper bound above has a similar form to the learning bound proposed by (Zhang et al., 2019). From the perspective of population loss, our domain loss function from the ensembles of two discriminator's digits is equivalent to that of the traditional MDD version where source-only domain discriminator f_0 is not introduced.

Next, we bound on the gap between empirical estimations of domain adversarial loss and its populated version. We first introduce Rademacher complexity as the richness of mapping from an arbitrary input space $\mathcal{X} \in \mathbb{D} \to \mathbb{R}$. The following states the formal definitions of the empirical and average Rademacher complexity.

Definition 5.1. (Rademacher Complexity) Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{X} \in \mathbb{D} \to \mathbb{R}$. And $\hat{D} = \{(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n)\}$ is a fixed sample of size n drawn from distribution $\mathcal{D}$ over $\mathbb{D}$. Then the empirical Rademacher complexity w.r.t sample $\hat{D}$ is defined as

$$\hat{\Re}_{n,\hat{D}}(\mathcal{G}) = \mathbb{E}_\delta \sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n}\delta_i g(\mathbf{x}_i) \tag{18}$$

where δ_i's independent uniform random variables taking values $\{+1, -1\}$. The random variables δ_i are called Rademacher variables.

The Rademacher complexity of $\mathcal{G}$ is the expectation of the empirical Rademacher complexity over all samples of size n drawn according to $\mathcal{D}$:

$$\Re_{n,\mathcal{D}}(\mathcal{G}) := \mathbb{E}_{\hat{D}\sim\mathcal{D}}[\hat{\Re}_{n,\hat{D}}(\mathcal{G})] \tag{19}$$

In the following, we define $\mathcal{G}_s$ as a family of source domain discrepancy loss function associated to $\mathcal{F}$ mapping from $\mathcal{X} \in \mathbb{D} \to \mathbb{R}$, $\mathcal{G}_t$ as a family of target domain discrepancy loss function associated to $\mathcal{F}$ mapping from $\mathcal{X}$ to $\mathbb{R}$:

$$\begin{aligned}
\mathcal{G}_s &= \{g_s : x \to \log(\frac{e^{\rho_{f'}(\mathbf{x},h_f)}}{1+e^{\rho_{f'}(\mathbf{x},h_f)}}) : f, f' \in \mathcal{F}\}\\
\mathcal{G}_t &= \{g_t : x \to \log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x},h_f)}}) : f, f' \in \mathcal{F}\}
\end{aligned} \tag{20}$$

With the Rademacher complexity defined above, we would proceed to show that our $\mathcal{H}$-divergence based domain adversarial loss could be empirically estimated through finite samples of source domain data and target domain data.

Theorem 3. *Let $f_0 \in \mathcal{F}$ be a fixed hypothesis that maps from $\mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ which satisfies $\rho_{f_0}(\mathbf{x}^s, h_f) \geq \epsilon_s$ for source domain data $\mathbf{x}^s \in S$ and $\rho_{f_0}(\mathbf{x}^t, h_f) \leq \epsilon_t$ for target domain data $\mathbf{x}^t \in T$. $\mathbf{x}_i^s$ is an i.i.d sample of size m drawn from the source distribution S and $\mathbf{x}_t^s$ is an i.i.d sample of size n drawn from the target distribution T. Given the same settings as Definition 5.1. For any $\delta > 0$, with the probability at least $1-2\delta$, we have the following generalization error bound for $\mathcal{H}$-divergence based adversarial loss function*

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}^s\in S}[\log(\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f_0}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f'}(\mathbf{x}^s,h_f)}})]\\
&+\mathbb{E}_{\mathbf{x}^t\in T}[\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)+\rho_{f_0}(\mathbf{x}^t,h_f)}})]\\
&\leq \frac{1}{m}\sum_{i=1}^{m}\log(\frac{e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f_0}(\mathbf{x}_i^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f'}(\mathbf{x}_i^s,h_f)}})\\
&+\frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}_i^t,h_f)+\rho_{f'}(\mathbf{x}_i^t,h_f)}})\\
&+\max\{\frac{2}{(e^{\epsilon_s}-1)\lambda_s^+ + 1}, \frac{2}{(e^{\epsilon_s}-1)\lambda_s^- + 1}\}\Re_{m,\mathcal{D}_s}(\mathcal{G}_s)\\
&+\max\{\frac{2e^{\epsilon_t}}{(1-\lambda_t^+)e^{\epsilon_t}+\lambda_t^+}, \frac{2e^{\epsilon_t}}{(1-\lambda_t^-)e^{\epsilon_t}+\lambda_t^-}\}\Re_{n,\mathcal{D}_t}(\mathcal{G}_t)\\
&+\sqrt{\frac{\log\frac{1}{\delta}}{2m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}
\end{aligned} \tag{21}$$

where λ_s^+, λ_s^-, λ_t^+ and λ_t^- is defined as

$$\begin{aligned}
&\lambda_s^- = \min\{\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)}}\}, \lambda_s^+ = \max\{\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)}}\}\\
&\lambda_t^- = \min\{\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)}}\}, \lambda_t^+ = \max\{\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)}}\},\\
&\forall \mathbf{x}^s \in S, \mathbf{x}^t \in T
\end{aligned} \tag{22}$$

Remark This theorem justifies that the populated domain adversarial loss with respect to $\mathcal{H}$-divergence could be approximated by the empirical one computed from finite source and target domain samples. By introducing source-only domain discriminator related f_0, the

empirical estimation error from the source and target domain could be non-uniform. The error term in the source domain is controlled by ϵ_s. And the error term in the target domain is controlled by ϵ_t. A large ϵ_s would achieve a lower generalization error from the source domain side. Conversely, over-training on f_0 would cause a large ϵ_t, resulting in a larger generalization error from the target domain side. Therefore, our theorem shows a trade-off between generalization error from the source and target domain side. Note that in the continual UDA case, there are only a small number of stored source domain samples for target domain adaptation. The source domain empirical error $\Re_{m,\mathcal{D}_s}(\mathcal{G}_s)$ becomes the major source of empirical estimation error. Thus it is justifiable to introduce a f_0 with a larger ϵ_s to exchange source domain error with target domain error. Our theorem also emphasizes the importance of training a better one-class score-based function $\rho_{f_0}(\mathbf{x}^s, h_f)$ with a higher score for in-distribution data on source domains than outliers.

Finally, we analyze the equilibrium of our adversarial loss w.r.t. generator and discriminators. We would show how our introduced source-only domain discriminator's score $\sigma_{h_f} \circ f'_s$ controls the magnitudes of consistency between source and target domain's distributions.

Proposition 1. *Consider the following optimization problem we have defined*

$$\begin{aligned}&\max_{f'} \mathbb{E}_{\hat{S}} \log(\sigma_{h_f} \circ f') + \mathbb{E}_{\hat{T}} \log(1 - \sigma_{h_f} \circ f') \\ &\min_{\hat{S},\hat{T}} \mathbb{E}_{\hat{S}} \log(\frac{1}{2}\sigma_{h_f} \circ f' + \frac{1}{2}\sigma_{h_f} \circ f_0) \\ &+ \mathbb{E}_{\hat{T}} \log(1 - \frac{1}{2}\sigma_{h_f} \circ f' - \frac{1}{2}\sigma_{h_f} \circ f_0)\end{aligned} \tag{23}$$

Assume that there is no restriction on the choice of f'. By fixing f_0, we have the following result.

The minimization problem w.r.t S and T is equivalent to minimization on the sum of two terms L_1 and L_2

$$\begin{aligned}L_1 = \; & 4KL(\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S} || \frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}) \\ + \; & 4KL(\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S} || \frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}) \\ + \; & 4KL(\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T} || \frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}) \\ + \; & 4KL(\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S} || \frac{3}{4}\hat{S} + \frac{1}{4}\hat{T})\end{aligned} \tag{24}$$

is a symmetric distribution divergence between $\hat{S}$ and $\hat{T}$ and has global minimum at $\hat{S} = \hat{T}$

$$\begin{aligned}L_2 = & \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \\ & \frac{1}{4 - (\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}))^2/(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2} d\mathbf{x}\end{aligned} \tag{25}$$

is a re-weighted bounds on the total variations between $\hat{p}_s(\mathbf{x})$ and $\hat{q}_t(\mathbf{x})$

Remark Recall that $\sigma_{h_f} \circ f_0(\mathbf{x})$ is the output score of the source-only domain discriminator for the possibilities of $\mathbf{x}$ belonging to the source domain. Assuming that for in-distribution area of source domain $\mathbf{x}$, $\sigma_{h_f} \circ f_0(\mathbf{x}) > \epsilon$, where $\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}) > 0$. Otherwise for out-of-distribution area $\sigma_{h_f} \circ, f_0(\mathbf{x}) < \epsilon$, where $\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}) > 0$. L_2 would be further approximated as $\tilde{L}_2$

$$\begin{aligned}&\tilde{L}_2 = 2 \int_{\mathbf{x}} (\sigma_{h_f} \circ f_0(\mathbf{x}) - \epsilon)(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \\ &\frac{1}{4 - (\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}))^2/(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2} d\mathbf{x} \\ &\|\tilde{L}_2 - L_2\| \leq \frac{1}{12}\|1 - 2\epsilon\|\end{aligned} \tag{26}$$

Furthermore, since $\sigma_{h_f} \circ f_0(\mathbf{x})$ is learned on the entire source domain dataset as a source-based function that relates to the source domain's distribution density, it re-weights the empirical distribution $\hat{p}_s(\mathbf{x})$ based on a small number of samples from the source domain stored in $\mathcal{M}$.

6 EXPERIMENTS

To evaluate the effectiveness of the double-head discriminator algorithm for Continual UDA. We describe the benchmark datasets and other experiment settings in Section C of our appendix. Then we perform the ablation study in Section 6.1. Next, we compare ours with various existing methods in 6.2.

6.1 Ablation Study

Effect of Different Memory Size To investigate the effect of different memory sizes on the model performance, we evaluate the task of continual adaptation to MNISTM, USPS, and SVHN with memory sizes of 8, 16, 32, 64, and 128 on each class of source domain (MNIST). The $+\infty$ shows the cases of offline adaptation where all source and target data would be accessed in an i.i.d. way. We show our result in Fig (3). With the increasing memory buffer size, the performance would slightly increase. However, our algorithm entails minimal performance loss from the smaller memory buffer size.

The Benefit of Source only Domain Discriminator To investigate the necessary of introducing source-only domain discriminator $h_{\psi,s}$ in phase T_1, we evaluate the contribution of $h_{\psi,s}$'s digits on learning the task model f_ω to adapt on the target domain. Specifically, we use $h_{\psi,s}(f^1_\omega(\mathbf{x}_i)) + \gamma h_{\psi,t}((f^1_\omega(\mathbf{x}_i)))$ in Equation (15) as the domain discriminator's signal to

Methods	Office-home Target Domain Adaptations											
	$Ar \to Cl$	$Ar \to Pr$	$Ar \to Re$	$Cl \to Ar$	$Cl \to Pr$	$Cl \to Re$	$Pr \to Ar$	$Pr \to Cl$	$Pr \to Re$	$Re \to Ar$	$Re \to Cl$	$Re \to Pr$
NLL-OT(Asano et al., 2019)	49.1	71.7	77.3	60.2	68.7	73.1	57.0	46.5	76.8	67.0	52.3	79.5
NLL-KL(Zhang et al., 2021)	49.0	71.5	77.1	59.0	68.7	72.9	56.4	46.9	76.6	66.2	52.3	79.1
HD-SHOT(Liang et al., 2020)	48.6	72.8	77.0	60.7	70.0	73.2	56.6	47.0	76.7	67.5	52.6	80.2
SD-SHOT(Liang et al., 2020)	50.1	75.0	78.8	63.2	72.9	76.4	60.0	48.0	79.4	69.2	54.2	81.6
DINE(Liang et al., 2022)	52.2	78.4	81.3	65.3	76.6	78.7	62.7	49.6	82.2	69.8	55.8	84.2
Ours	53.8	78.8	81.9	66.4	77.8	77.9	63.0	52.9	83.2	72.0	59.4	84.9
Ours+KD	**54.8**	**81.1**	**84.0**	**67.5**	**79.0**	**80.5**	**65.1**	**53.8**	**84.5**	**73.2**	**60.0**	**86.7**
Ours+SL	54.0	79.2	82.4	66.8	78.3	79.0	63.7	53.2	83.2	72.8	59.4	85.8
i.i.d.-adv	54.9	79.0	82.8	67.0	78.7	78.1	63.6	54.2	83.8	72.9	60.8	85.8

Table 1: Comparison of Target Domain Adaptation Performance on Office-home.

Methods	Office-home Source Domain Forgetting											
	$Ar \to Cl$	$Ar \to Pr$	$Ar \to Re$	$Cl \to Ar$	$Cl \to Pr$	$Cl \to Re$	$Pr \to Ar$	$Pr \to Cl$	$Pr \to Re$	$Re \to Ar$	$Re \to Cl$	$Re \to Pr$
NLL-OT(Asano et al., 2019)	10.91	7.64	7.31	12.73	13.18	11.13	7.29	7.72	6.19	7.07	7.28	5.35
NLL-KL(Zhang et al., 2021)	10.93	7.66	7.34	13.01	13.05	10.98	7.27	7.50	6.03	6.97	7.26	5.46
HD-SHOT(Liang et al., 2020)	11.10	9.69	8.06	14.99	15.02	12.06	7.57	7.86	6.58	7.22	7.92	6.02
SD-SHOT(Liang et al., 2020)	11.21	8.93	7.89	15.24	15.55	12.25	7.75	7.93	6.72	7.22	8.13	6.05
DINE(Liang et al., 2022)	9.67	6.66	6.26	9.29	10.02	9.76	6.13	5.92	5.82	6.19	6.05	4.93
Ours	**4.52**	**3.95**	**3.53**	**5.12**	**4.83**	**4.69**	**1.93**	**2.05**	**1.89**	**2.12**	**3.13**	**1.43**

Table 2: Comparison of Source Domain Forgetting Performance on Office-home.

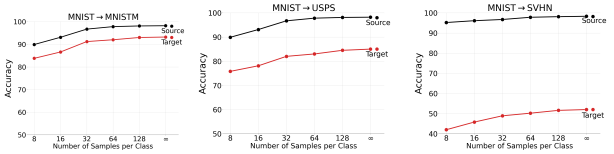


Figure 3: Effect of different memory size on model performance

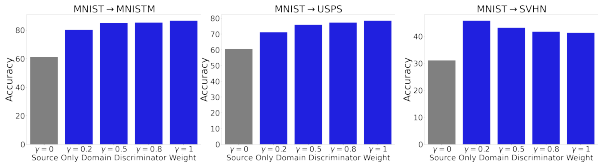


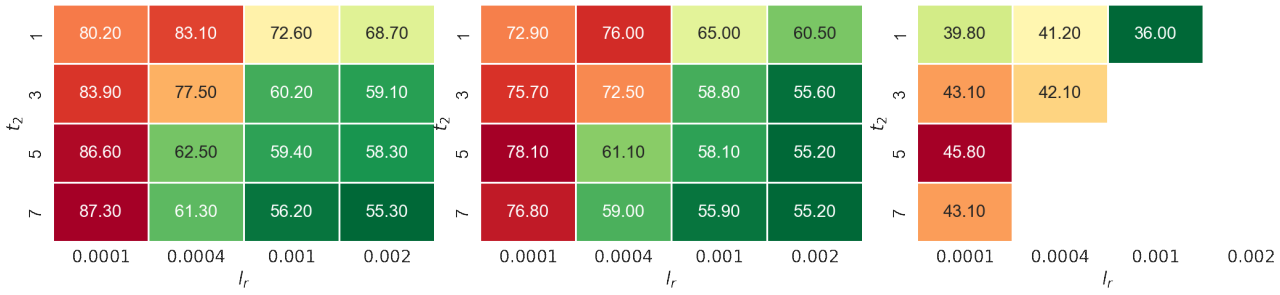
Figure 4: Effect of source only domain discriminator's contribution on target adaptation performance

adapt f_ω. We shift γ from 0 to 1 where $h_{\psi,s}$ is gradually mixed with $h_{\psi,t}$. The result in Fig (4) showed that the performance is significantly lower in $\gamma = 0$ where the source-only domain discriminator that is trained in the source phase is not used for adaptation in phase T_1. In the particular case of $\gamma = 0$, it follows the standard setting of domain adversarial training that one domain discriminator is adversarial trained with task predictor on domain adversarial loss. When γ shifted from 0 to 1, the source-only domain discriminator $h_{\psi,s}$'s signal is gradually incorporated into $h_{\psi,t}$ with a weight determined by γ. Our result emphasizes the importance of introducing an additional pre-trained domain discriminator on the S_0 phase. The choice of γ to ensemble domain predictions from $h_{\psi,t}$ and $h_{\psi,s}$ adopts a wide range from 0 to 1. In MNIST to SVHN task, $\gamma = 0.2$ has a better result because the data variations of SVHN are much larger than MNIST, and a smaller γ would have less empirical error from the target domain side as we have analyzed on Theorem 3.

Effect of Learning Rate and Epoch on Source-only Domain Discriminator Training source only domain discriminator has resemblance of one-class learning on a score-based function, the learning rate and epoch plays a important role to assure sensitivity to in-distribution data while prevent over saturation. In this section, we investigate the effect of learning rate and epoch on source-only domain discriminator. We plot the combinatorial case on the learning rate of 0.0001, 0.0004, 0.001, and 0.002 and epochs of 1, 3, 5, and 7 training epochs as a heatmap that are shown in Fig (5). We observe that pre-training a source-only domain discriminator in S_0 phase with a smaller learning rate and moderate number of training epochs would lead to better performance of target adaption in T_1 phase. This accords with the observation of the learning rate and epoch's effect on the performance of one-class learning in (Hu et al., 2020). For a stable and optimized performance, we choose the learning epoch source of 5 with a learning rate of 0.0001 in the rest of our experiment.

6.2 Comparison to Existing Continual UDA

Baseline We compare our proposed method with two strong baselines, Knowledge Distillation (KD) and Self-Learning (ST), which are commonly used semi-



(a) MNIST→MNISTM (b) MNIST→USPS (c) MNIST→SVHN

Figure 5: Effect of Source-only Domain Discriminator's learning rate l_r and training epochs t_2 on target adaptation performance.

supervised learning (SSL) techniques for continual UDA. KD transfers knowledge from the source to the target domain by distilling the source domain teach model's pseudolikelihoods assigned to unlabeled target domain (Hinton et al., 2015). A representative work on KD is DINE (Liang et al., 2022). ST trains the model on source labeled data first, then iteratively assigns pseudo-labels to the unlabeled target domain and trains on the most confident predictions (Nigam and Ghani, 2000). Variants of ST include NLL-OT (Asano et al., 2019) and NLL-KL (Zhang et al., 2021). SHOT (Liang et al., 2020) combines ST with K-Means by assigning pseudo-label from its distance to cluster centroid.

Results with office-31 are presented in Table (3, 4), and those for office-home are shown in Table (1, 2). Nearly all categories of the results in our proposed method show improvement in the target domain adaptation task upon the baseline method. Additionally, our method effectively addresses the issue of catastrophic forgetting on the source domain by employing adversarial adaptation to learn a domain-generalized model. However, our proposed method sometimes only sees minor improvement over baseline or even falls short in rare cases. We believe that this is because of the sub-optimal optimization behavior of adversarial training, which involves minimaximization on saddle point. One way to further improve the performance of our proposed method is to follow with a final stage of SSL fine-tuning. As SSL performance improves with decreasing domain discrepancy (Ben-David et al., 2010; Zhao et al., 2019; Kumar et al., 2020), our proposed method can be used as a pre-processing step for SSL. This can potentially lead to improved SSL performance. There is no surprise that Ours+KD achieve over 2% performance increase among all categories of baseline methods including purely adversarial adaptation in i.i.d. settings as the result of a final stage fine-tuning. Because the combined use of SSL and adversarial domain adaptation would achieve better results than each individual one. As a result of final stage fune-tuning

	Office-31 Target Domain Adaptations					
Methods	$A \to W$	$D \to W$	$W \to D$	$A \to D$	$D \to A$	$W \to A$
NLL-OT(Asano et al., 2019)	85.5	95.1	98.7	88.8	64.6	66.7
NLL-KL(Zhang et al., 2021)	86.8	94.8	98.7	89.4	65.1	67.1
HD-SHOT(Liang et al., 2020)	83.1	95.1	98.1	86.5	66.1	68.9
SD-SHOT(Liang et al., 2020)	83.7	95.3	97.1	89.2	67.9	71.1
DINE(Liang et al., 2022)	86.8	96.2	98.6	91.6	72.2	73.3
Ours	92.6	97.3	99.2	92.0	73.9	73.8
Ours+KD	**93.8**	**98.4**	**100.0**	**93.8**	**74.0**	**75.6**
Ours+SL	93.2	97.7	100.0	92.5	73.9	74.4
i.i.d.-adv	94.5	98.4	100.0	93.5	74.6	74.2

Table 3: Office-31 Target Domain Adaptation

	Office-31 Source Domain Forgetting					
Methods	$A \to W$	$D \to W$	$W \to D$	$A \to D$	$D \to A$	$W \to A$
NLL-OT(Asano et al., 2019)	4.53	3.14	2.73	4.30	6.17	5.11
NLL-KL(Zhang et al., 2021)	4.37	2.99	2.48	4.02	5.94	4.99
HD-SHOT(Liang et al., 2020)	5.12	4.01	3.98	4.87	7.80	5.56
SD-SHOT(Liang et al., 2020)	5.31	4.54	4.03	4.85	7.88	5.72
DINE(Liang et al., 2022)	3.81	2.16	1.50	3.32	5.08	3.98
Ours	**1.97**	**1.03**	**0.98**	**1.55**	**3.72**	**2.96**

Table 4: Office-31 Source Domain Forgetting

with KD after our proposed adversarial domain adaptation, we would achieve over 2% performance increase among all categories of baseline methods on Ours+KD in CL including purely adversarial adaptation in i.i.d. settings.

7 CONCLUSION

We have proposed a double-head discriminator algorithm for continual adversarial domain adaptation. With our introduced source-only domain discriminator, the empirical estimation error of the $\mathcal{H}$-divergence in domain adversarial loss is reduced from the source domain side. Extensive experiments have shown that our proposed algorithm has consistently outperformed the existing baseline. For future work, we would focus on a memory-free continual adversarial UDA algorithm.

Acknowledgements

The work is supported by NSF CAREER award IIS-2239537.

References

E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

Y. M. Asano, C. Rupprecht, and A. Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

E. Belouadah and A. Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 583–592, 2019.

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

F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018.

A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. Adanet: Adaptive structural learning of artificial neural networks. In *International conference on machine learning*, pages 874–883. PMLR, 2017.

M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

S. Dey, V. Arora, and S. N. Tripathi. Leveraging unsupervised data and domain adaptation for deep regression in low-cost sensor calibration. *arXiv preprint arXiv:2210.00521*, 2022.

P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019.

N. Ding, Y. Xu, Y. Tang, C. Xu, Y. Wang, and D. Tao. Source-free domain adaptation via distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7212–7222, 2022.

P. Dokania, P. Torr, and M. Ranzato. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

J. Dong, S. Zhou, B. Wang, and H. Zhao. Algorithms and theory for supervised gradual domain adaptation. *Transactions on Machine Learning Research*, 2022.

A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020.

E. Fini, S. Lathuiliere, E. Sangineto, M. Nabi, and E. Ricci. Online continual learning under extreme memory constraints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 720–735. Springer, 2020.

F. Fleuret et al. Uncertainty reduction for model adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9613–9623, 2021.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

T. Gong, J. Jeong, T. Kim, Y. Kim, J. Shin, and S.-J. Lee. Robust continual test-time adaptation: Instance-aware bn and prediction-balanced memory. *arXiv preprint arXiv:2208.05117*, 2022.

I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

T. L. Hayes, K. Kafle, R. Shrestha, M. Acharya, and C. Kanan. Remind your neural network to prevent catastrophic forgetting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 466–483. Springer, 2020.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference*

on Computer Vision and Pattern Recognition, pages 831–839, 2019.

W. Hu, M. Wang, Q. Qin, J. Ma, and B. Liu. Hrn: A holistic approach to one class learning. *Advances in neural information processing systems*, 33:19111–19124, 2020.

Z. Ji, D. Guo, P. Wang, K. Yan, L. Lu, M. Xu, J. Zhou, Q. Wang, J. Ge, M. Gao, et al. Continual segment: towards a single, unified and accessible continual segmentation model of 143 whole-body organs in ct scans. *arXiv preprint arXiv:2302.00162*, 2023.

R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526, 2017.

A. Kumar, T. Ma, and P. Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020.

J. N. Kundu, N. Venkat, R. V. Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9641–9650, 2020.

Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

J. Liang, D. Hu, J. Feng, and R. He. Dine: Domain adaptation from single and multiple black-box predictors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8003–8013, 2022.

Y. Liu, B. Schiele, and Q. Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021.

M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.

D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

C. Ma, Z. Ji, Z. Huang, Y. Shen, M. Gao, and J. Xu. Progressive voronoi diagram subdivision enables accurate data-free class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2022.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93, 2000.

S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

X. Peng, Z. Huang, Y. Zhu, and K. Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2019.

H. Pham, Z. Dai, Q. Xie, and Q. V. Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021.

A. Prabhu, P. H. Torr, and P. K. Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer, 2020.

X. Qin, X. Song, and S. Jiang. Bi-level meta-learning for few-shot domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15900–15910, 2023.

M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2018.

M. Rostami. Lifelong domain adaptation via consolidated internal distribution. *Advances in neural information processing systems*, 34:11172–11183, 2021.

A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

S. Sankaranarayanan and Y. Balaji. Meta learning for domain generalization. In *Meta-Learning with Medical Imaging and Health Informatics Applications*, pages 75–86. Elsevier, 2023.

Y. Shen, J. Du, H. Zhao, Z. Ji, C. Ma, and M. Gao. Fedmm: A communication efficient solver for federated adversarial domain adaptation. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1808–1816, 2023.

S. Tang, P. Su, D. Chen, and W. Ouyang. Gradient regularized contrastive learning for continual domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2665–2673, 2021.

J. Tian, J. Zhang, W. Li, and D. Xu. Vdm-da: Virtual domain modeling for source data-free domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3749–3760, 2021.

R. Volpi, D. Larlus, and G. Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4443–4453, 2021.

F. Wang, Z. Han, Y. Gong, and Y. Yin. Exploring domain-invariant parameters for source free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2022a.

H. Wang, B. Li, and H. Zhao. Understanding gradual domain adaptation: Improved analysis, optimal path and beyond. In *International Conference on Machine Learning*, pages 22784–22801. PMLR, 2022b.

Z. Wang, L. Shen, T. Duan, D. Zhan, L. Fang, and M. Gao. Learning to learn and remember super long multi-domain task sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7982–7992, 2022c.

Z. Wang, L. Shen, L. Fang, Q. Suo, T. Duan, and M. Gao. Improving task-free continual learning by distributionally robust memory evolution. In *International Conference on Machine Learning*, pages 22985–22998. PMLR, 2022d.

Z. Wang, X. Wang, L. Shen, Q. Suo, K. Song, D. Yu, Y. Shen, and M. Gao. Meta-learning without data via wasserstein distributionally-robust model fusion. In *Uncertainty in Artificial Intelligence*, pages 2045–2055. PMLR, 2022e.

Z. Wang, L. Shen, T. Duan, Q. Suo, L. Fang, W. Liu, and M. Gao. Distributionally robust memory evolution with generalized divergence for continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

D. M. WU SJ. The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China*, pages 833–844, 2019.

H. Xia, H. Zhao, and Z. Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9010–9019, 2021.

R. Xian, H. Ji, and H. Zhao. Cross-lingual transfer with class-weighted language-invariant representations. In *International Conference on Learning Representations*, 2021.

S. Yang, Y. Wang, J. Van De Weijer, L. Herranz, and S. Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

H.-W. Yeh, B. Yang, P. C. Yuen, and T. Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 474–483, 2021.

J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

H. Zhang, Y. Zhang, K. Jia, and L. Zhang. Unsupervised domain adaptation of black-box source models. *arXiv preprint arXiv:2101.02839*, 2021.

Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020.

H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

G. Zhou, K. Sohn, and H. Lee. Online incremental feature learning with denoising autoencoders. In *Artificial intelligence and statistics*, pages 1453–1461. PMLR, 2012.

F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

1 OVERVIEW OF ADVERSARIAL DOMAIN ADAPTATION

Let T and S be the source and target distributions, respectively. In a general formulation, the upper bound of the target prediction error is given by Ben-David et.al, (Ben-David et al., 2010)

Theorem 1. *Let $\mathcal{F}$ be the hypothesis space. For any classifier $f \in \mathcal{F}$, err_S denotes the population loss of a classifier $f \in \mathcal{F}$ under the source distribution S, i.e., $err_{\mathcal{S}}(f) \triangleq \mathbb{E}_{(\mathbf{x}_i,\mathbf{y}_i)\sim S}[\ell(f(\mathbf{x}_i), y_i)]$ And $err_T(f)$ parallel notates for the target domain error. respectively. Then for any classifier $f \in \mathcal{H}$,*

$$err_S(f) \leq err_T(f) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{T}, \mathcal{S}) + \min_{f^* \in \mathcal{F}} \{err_S(f^*) + err_T(f^*)\}, \tag{1}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{T}, \mathcal{S})$ is a discrepancy-based distance, known as the $\mathcal{H}$-divergence, and $\min_{f^* \in \mathcal{H}}\{err_{\mathcal{P}}(f^*) + err_{\mathcal{Q}}(f^*)\}$ is the optimal joint error, i.e., the sum of source and target domain's population loss of f in a hypothesis class $\mathcal{F}$.

For the unsupervised domain adaptation problem, it has been proven that minimizing the upper bound, which is the r.h.s in (1), leads to an architecture consisting of a *feature extractor* parameterized by ω, i.e., f^1_ω, a *label predictor*, parameterized also by ω i.e., f^2_ω ($f_\omega \triangleq f^2_\omega \circ f^1_\omega$), [1] and a *domain classifier* parameterized by ψ, i.e., h_ψ, as shown in Fig (1) (Ganin and Lempitsky, 2015; Zhao et al., 2018). The feature extractor generates the domain-independent feature representations, which are then fed into the domain classifier and label predictor. The domain classifier then tries to determine whether the extracted features belong to the source or target domain. Meanwhile, the label predictor predicts instance labels based on the extracted features of the labeled source-domain instances.

In Adversarial Domain Adaptation, an additional learning objective of $d_{\mathcal{H}\Delta\mathcal{H}}$ is introduced to encourage the extracted features to be both discriminative and invariant to changes between the source and target domains. By extending the $\mathcal{H}$-divergence to general loss function in (Mansour et al., 2009), r.h.s in (1) is equivalent as

$$\min_{\omega} \max_{\psi} \triangleq \mathbb{E}_{(\mathbf{x}^s_i, y_i)\sim S} \ell(f_\omega(\mathbf{x}^s_i), y_i) + \nu \mathbb{E}_{\mathbf{x}^s_i \sim S} D^s_\psi(\mathbf{x}^s_i) + \nu \mathbb{E}_{\mathbf{x}^t_i \sim T} D^t_\psi(\mathbf{x}^t_i) \tag{2}$$

where $D^s_\psi(\mathbf{x}^s_i) \triangleq D^s(h_\psi(f^1_\omega(\mathbf{x}^s_i))$ and $D^t_\psi(\mathbf{x}^t_i) \triangleq -D^t(h_\psi(f^1_\omega(\mathbf{x}^t_i))$

In the majority of domain adversarial problems, $d_{\mathcal{H}\Delta\mathcal{H}}$ is reformulated as the difference between the parameterized output of the domain classifier on the source domain and the target domain, given by $\mathbb{E}_{\mathbf{x}^s_i \sim S} D^s_\psi(\mathbf{x}^s_i) + \mathbb{E}_{\mathbf{x}^t_i \sim T} D^t_\psi(\mathbf{x}^t_i)$. This term is commonly referred to as the *domain adversarial loss*.

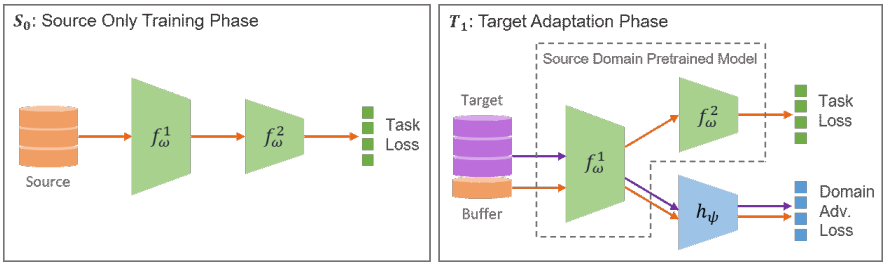


Figure 1: A continual adversarial domain adaptation model. Only the source risk of the client's local source data is accessible in source only training phase. A small set of buffered source domain data and target domain data is adversarial trained in target adaptation phase.

[1] The parameters of f^1 and f^2 are not the same. In this case, we abuse the notation to simplify the expression.

2 HOLISTIC REGULATED ONE-CLASS DOMAIN DISCRIMINATOR

Hu et.al, (Hu et al., 2020) proposed HRN, a simple but efficient deep one-class learning algorithm. And we adopt HRN as another training oracle for source only domain discriminator with scalar domain discriminators $h_{\psi,s} : \mathcal{Z} \in \mathbb{F} \to \mathbb{R}$ such as DANN (Ganin et al., 2016) and CDAN (Long et al., 2017).

$$\min_{\psi_s} \mathbb{E}_{\mathbf{x}_i^s \sim S}[-\log(\sigma(h_{\psi,s}((\mathbf{z}(\mathbf{x}_i^s)))))] + \lambda \|\nabla_{\mathbf{z}} h_{\psi,s}(\mathbf{z}(\mathbf{x}_i^s))\|_2^n \tag{3}$$

where $\mathbf{z}$ is the domain features that are fed as the input to domain discriminator. In general, the domain features that could be used for $\mathbf{z}$ include but not limited to the following cases:

- Domain-Adversarial Neural Networks (DANN) (Ganin et al., 2016), $\mathbf{z}$ is designed simply to be the domain invariant feature $f_\omega^1(\mathbf{x}_i^s)$
$$\mathbf{z} \triangleq f_\omega^1(\mathbf{x}_i^s) \tag{4}$$
- Conditional Domain Adaptation Network (CDAN) (Long et al., 2017), $\mathbf{z}$ is from the cross-product space of $f_\omega^1(\mathbf{x}_i^s)$ and $f_\omega(\mathbf{x}_i^s)$
$$\mathbf{z} \triangleq f_\omega^1(\mathbf{x}_i^s) \otimes f_\omega^1(\mathbf{x}_i^s) \tag{5}$$

Apart from the commonly adopted NLL for classification, HRN adds an additional regularization term on the n's power of scalar domain discriminator $h_{\psi,s}(\cdot)$'s gradient norm.And n is the exponential term which is used with λ to control the strength of regularization.The full description of our double-head domain discriminator algorithm for scalar domain discriminator is shown in Algorithm 2.

3 EXPERIMENT SETUP

MNISTM (Ganin et al., 2016) is a dataset that demonstrates domain adaptation by combining MNIST with randomly colored image patches from the BSD500 dataset.

USPS (Hull, 1994) is a digit dataset automatically scanned from envelopes by the U.S. Postal Service containing pixel grayscale samples. The images are centered, normalized. And a broad range of font styles are shown in the dataset.

SVHN has RGB images of printed digits clipped from photographs of house number plates. The trimmed photos are centered on the digit of interest while surrounding digits and other distractions are retained. Photos of house numbers in various countries was used to create the SVHN dataset.

Office-31 (Saenko et al., 2010) is a typical domain adaptation dataset made up of three distinct domains with 31 categories in each domain. There are 4,652 images in total from 31 classes.

Office-home (Venkateswara et al., 2017) is a typical domain adaptation dataset made up of four distinct domains with 65 categories in each domain. There are total 15,500 images in total from 65 classes

Implementation Details On MNISTM, USPS and SVHN, we use a three-layer convolutional network as the invariant feature extractor, and the network models are trained from random initialization on server. On Office-31 and Office-home, we use the pre-trained ResNet50 (He et al., 2016) on ImageNet (Russakovsky et al., 2015) as the feature extractor. Both the task classifier and the domain classifier are two-layer fully-connected neural networks. The domain classifier's parameter are trained from random initialization in all settings. In Office-31 and Office-home datasets, we set the memory buffer size as 10 samples per class. We uniformly use supervised training in source domain data for 15 epochs in S_0 phase. Following supervised supervised training, we train freeze our task model and train source only domain classifier for 5 epochs in all remaining experiment except for ablation study. Learning rate of task model and domain discriminator is fixed with 0.001 on Adam optimizer. The source only domain discriminator is trained with Adam optimizer of learning rate 0.0001 for 5 epochs in all remaining experiment except for ablation study.

4 ADDITIONAL EXPERIMENT RESULT FOR HRN

Table 4: Experiment on testing with holistic-regulated training on source-only domain discriminator. The source-only domain discriminator $h_{\psi,s}$ of scalar output is trained on holistic regulated one-class loss in Equation (3).

Algorithm 1 Double Head Discriminator Algorithm

Initialization: Task Model $f_\omega \triangleq f^2_\omega \circ f^1_\omega$
Source Only multi-class Domain Classifier $h_{\psi,s}$

Phase 1 – Source Only training phase

procedure TASK MODEL TRAINING PHASE
  for $t \in \{1, \dots, t_1\}$ **do**
    for $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_K, y_K)\} \overset{K}{\sim} S_0$ **do**
      $L = \frac{1}{K} \sum_{i=i}^{K} \ell(f_\omega(\mathbf{x}_i), y_i)$
      $\omega \rightarrow SGD(L, \omega)$ ▷ Train Task Model on Source Domain
    end for
  end for
end procedure
procedure SOURCE ONLY DOMAIN CLASSIFIER TRAINING PHASE
  for $t \in \{1, \dots, t_2\}$ **do**
    for $(\mathbf{x}_1, \dots, \mathbf{x}_K) \overset{K}{\sim} S_0$ **do**
      $d'(\mathbf{x}) \rightarrow \arg\max_c f_\omega(\mathbf{x}_i, c) \quad \forall \mathbf{x} \in \{\mathbf{x}_1 \dots \mathbf{x}_K\}$ ▷ Get Pseudo Domain Label from Task Model
      $D = \frac{1}{K} \sum_{i=i}^{K} -\log(\text{softmax}(h_{\psi,s}(f^1_\omega(\mathbf{x}_i)), d'(\mathbf{x}_i)))$
      $\psi_s \rightarrow SGD(D, \psi_s)$ ▷ Train on Source Only Domain Classifier
    end for
  end for
end procedure

Phase 2 – Sample on Source Domain Replay Memory

procedure MEMORY SAMPLE PHASE
  $\mathcal{M} \rightarrow \{\}$
  for $c \in \{1, \dots, C\}$ **do**
    Sample $\{(\mathbf{x}_1, c), \dots, (\mathbf{x}_N, c)\} \overset{N}{\sim} S_0$
    $\mathcal{M}.append(\{(\mathbf{x}_1, c), \dots, (\mathbf{x}_N, c)\})$ ▷ Store N data per class on source domain on Replay Memory
  end for
end procedure

Phase 3 – Unlabeled Target Adaptation Phase with Memory Reply

Initialization:Target Adaptation Phase multi-class Domain Classifier $h_{\psi,t}$
procedure TARGET PHASE
  for $t \in \{1, \dots, t_3\}$ **do**
    for $\{(\mathbf{x}^s_1, y^s_1), \dots, (\mathbf{x}^s_K, y^s_K)\} \overset{K}{\sim} \mathcal{M}$, $(\mathbf{x}^t_1 \dots, \mathbf{x}^t_K) \overset{K}{\sim} T_1$ **do**
      $L = \frac{1}{K} \sum_{i=1}^{K} \ell(f_\omega(\mathbf{x}^s_i), y^s_i)$
      $\mathbf{d}'(\mathbf{x}) \rightarrow \arg\max_c f_\omega(\mathbf{x}, c) \quad \forall \mathbf{x} \in \{\mathbf{x}^s_1 \dots \mathbf{x}^s_K, \mathbf{x}^t_1 \dots \mathbf{x}^t_K\}$
      $D_{\psi,t} = \frac{1}{K} \sum_{i=1}^{K} -\log(\text{softmax}(h_{\psi,t}(f^1_\omega(\mathbf{x}^s_i)), \mathbf{d}'(\mathbf{x}^s_i))) - \log(1 - \text{softmax}(h_{\psi,t}(f^1_\omega(\mathbf{x}^t_i)), \mathbf{d}'(\mathbf{x}^t_i)))$
      $D_\psi = \frac{1}{K} \sum_{i=1}^{K} -\log(\text{softmax}(h_{\psi,s}(f^1_\omega(\mathbf{x}^s_i)) + h_{\psi,t}(f^1_\omega(\mathbf{x}^s_i)), \mathbf{d}'(\mathbf{x}^s_i))) - \log(1 - \text{softmax}(h_{\psi,s}(f^1_\omega(\mathbf{x}^t_i)) + h_{\psi,t}(f^1_\omega(\mathbf{x}^t_i)), \mathbf{d}'(\mathbf{x}^t_i)))$
      $\omega \rightarrow SGD(L - \beta D_\psi, \omega)$
      $\psi_t \rightarrow SGD(D_{\psi,t}, \psi_t)$
    end for
  end for
end procedure

Algorithm 2 Double Head Discriminator Algorithm For Scalar Domain Discriminator

Initialization: Task Model $f_\omega \triangleq f^2_\omega \circ f^1_\omega$
Source Only scalar Domain Classifier $h_{\psi,s}$

Phase 1 – Source Only training phase

procedure TASK MODEL TRAINING PHASE
  for $t \in \{1, \ldots, t_1\}$ **do**
    for $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_K, y_K)\} \overset{K}{\sim} S_0$ **do**
      $L = \frac{1}{K}\sum_{i=i}^{K} \ell(f_\omega(\mathbf{x}_i), y_i)$
      $\omega \rightarrow SGD(L, \omega)$ ▷ Train Task Model on Source Domain
    end for
  end for
end procedure
procedure SOURCE ONLY DOMAIN CLASSIFIER TRAINING PHASE
  for $t \in \{1, \ldots, t_2\}$ **do**
    for $(\mathbf{x}_1, \ldots, \mathbf{x}_K) \overset{K}{\sim} S_0$ **do**
      $D = \frac{1}{K}\sum_{i=i}^{K} -\log(\sigma(h_{\psi,s}(f^1_\omega(\mathbf{x}_i)),) + \lambda\|\nabla_{\mathbf{z}} h_{\psi,s}(\mathbf{z})|_{\mathbf{z}=f^1_\omega(\mathbf{x}_i)}\|^n_2$
      $\psi_s \rightarrow SGD(D, \psi_s)$ ▷ Train on Source Only Domain Classifier
    end for
  end for
end procedure

Phase 2 – Sample on Source Domain Replay Memory

procedure MEMORY SAMPLE PHASE
  $\mathcal{M} \rightarrow \{\}$
  for $c \in \{1, \ldots, C\}$ **do**
    Sample $\{(\mathbf{x}_1, c), \ldots, (\mathbf{x}_N, c)\} \overset{N}{\sim} S_0$
    $\mathcal{M}.append(\{(\mathbf{x}_1, c), \ldots, (\mathbf{x}_N, c)\})$ ▷ Store N data per class on source domain on Replay Memory
  end for
end procedure

Phase 3 – Unlabeled Target Adaptation Phase with Memory Reply

Initialization:Target Adaptation Phase scalar Domain Classifier $h_{\psi,t}$
procedure TARGET ADAPTATION PHASE
  for $t \in \{1, \ldots, t_3\}$ **do**
    for $\{(\mathbf{x}^s_1, y^s_1), \ldots, (\mathbf{x}^s_K, y^s_K)\} \overset{K}{\sim} \mathcal{M}$, $(\mathbf{x}^t_1 \ldots, \mathbf{x}^t_K) \overset{K}{\sim} T_1$ **do**
      $L = \frac{1}{K}\sum_{i=i}^{K} \ell(f_\omega(\mathbf{x}^s_i), y_i)$
      $D_{\psi,t} = \frac{1}{K}\sum_{i=i}^{K} -\log(\sigma(h_{\psi,t}(f^1_\omega(\mathbf{x}^s_i)))) - \log(\sigma(-h_{\psi,t}(f^1_\omega(\mathbf{x}^t_i))))$
      $D_{\psi} = \frac{1}{K}\sum_{i=i}^{K} -\log(\sigma(h_{\psi,s}(f^1_\omega(\mathbf{x}^s_i)) + h_{\psi,t}(f^1_\omega(\mathbf{x}^s_i)))) - \log(\sigma(-h_{\psi,s}(f^1_\omega(\mathbf{x}^t_i)) - h_{\psi,t}(f^1_\omega(\mathbf{x}^t_i))))$
      $\omega \rightarrow SGD(L - \beta D_\psi, \omega)$
      $\psi_t \rightarrow SGD(D_{\psi,t}, \psi_t)$
    end for
  end for
end procedure

Table 1: Holistic Regulated One-class Domain Discriminator.

Discriminator Used	DANN(T_1Only)	CDAN(T_1Only)	HRN-DANN	HRN-CDAN
MNIST → MNISTM	58.8	59.2	78.1	80.3
MNIST → USPS	60.6	62.3	69.1	73.4
MNIST → SVHN	32.1	35.7	37.5	40.8

The rest of domain adversarial training is the same as in Algorithm (2) using the ensembles of two discriminators digits as domain invariant signals for feature extractor, f^1_ω. $n = 6$ and $\lambda = 0.1$ is used as holistic regulated loss on $h_{\psi,s}$ for stable performance. In general, including a holistic regulated source-only domain discriminator has performance improvement over using single domain discriminator in T_1 only. However the HRN method of training a scalar domain discriminator is inferior than MDD included multi-class domain discriminator for continual UDA.

5 PROOF OF THEOREM 2

Lemma 1. *(Lemma C.1, (Zhang et al., 2019)) For any distribution D and any f, we have*

$$disp^{(\rho)}_D(f', f) = err^{(\rho)}_D(f') + err^{(\rho)}_D(f) \tag{6}$$

Theorem 2. *For a hypothesis class $\mathcal{F}$ and a fixed $f_0 \in \mathcal{F}$ where for every $f \in \mathcal{F}$, $f - f_0$ is also in $\mathcal{F}$, then we have the following property holds*

$$err_T(f) \le err^{(\rho)}_S(f) + d^{(\rho)}_{f,f_0,\mathcal{F}}(S,T) + \lambda \tag{7}$$

where $err^{(\rho)}_S(f)$, $d^{(\rho)}_{f,f_0,\mathcal{F}}(S,T)$ and λ is defined as

$$\begin{aligned} &err^{(\rho)}_S(f) = \mathbb{E}_{(x_i,y_i)\sim S}\Phi_\rho \circ \rho_f(x_i, y_i) \\ &d^{(\rho)}_{f,f_0,\mathcal{F}}(S,T) = \sup_{f'\in\mathcal{F}} \{\mathbb{E}_{x_i\sim T}\Phi_\rho \circ \rho_{f'+f_0}(x_i, h_f(x_i)) - \mathbb{E}_{x_i\sim S}\Phi_\rho \circ \rho_{f'+f_0}(x_i, h_f(x_i))\} \\ &\lambda = \min_{f^\star\in\mathcal{F}} err^{(\rho)}_S(f^\star) + err^{(\rho)}_T(f^\star), \end{aligned} \tag{8}$$

Proof. We first define $f^\star$ be the ideal joint hypothesis which minimizes the combined margin loss,

$$f^\star \triangleq \arg\min_{f\in\mathcal{F}} \{err^{(\rho)}_S(f) + err^{(\rho)}_T(f)\} \tag{9}$$

$$\begin{aligned} err_T(f) \le& \mathbb{E}_T \mathbb{1}[h_f \neq h_{f^\star}] + \mathbb{E}_T \mathbb{1}[h_{f^\star} \neq y] \\ \le& err^{(\rho)}_S(f) - err^{(\rho)}_S(f) + disp^{(\rho)}_T(f^\star, f) + err^{(\rho)}_T(f^\star) \end{aligned} \tag{10}$$

From the triangular inequality of margin discrepancy(Lemma C.1, (Zhang et al., 2019)), we have

$$\begin{aligned} err_T(f) \le& err^{(\rho)}_S(f) - err^{(\rho)}_S(f) + disp^{(\rho)}_T(f^\star, f) + err^{(\rho)}_T(f^\star) \\ \le& err^{(\rho)}_S(f) + err^{(\rho)}_S(f^\star) - disp^{(\rho)}_S(f^\star, f) + disp^{(\rho)}_T(f^\star, f) + err^{(\rho)}_T(f^\star) \end{aligned} \tag{11}$$

Let we define $f_1 \triangleq f^\star - f_0$. From the properties of hypothesis class $\mathcal{F}$, we have $f_1 \in \mathcal{F}$. By substituting the definition of f_1 into $disp^{(\rho)}_S(f^\star, f)$ and $disp^{(\rho)}_T(f^\star, f)$, we have

$$\begin{aligned} &disp^{(\rho)}_T(f^\star, f) - disp^{(\rho)}_S(f^\star, f) = \mathbb{E}_{x_i\sim T}\Phi_\rho \circ \rho_{f^\star}(x_i, h_f(x_i)) - \mathbb{E}_{x_i\sim S}\Phi_\rho \circ \rho_{f^\star}(x_i, h_f(x_i)) \\ &= \mathbb{E}_{x_i\sim T}\Phi_\rho \circ \rho_{f_1+f_0}(x_i, h_f(x_i)) - \mathbb{E}_{x_i\sim S}\Phi_\rho \circ \rho_{f_1+f_0}(x_i, h_f(x_i)) \\ &\le \sup_{f'\in\mathcal{F}} \{\mathbb{E}_{x_i\sim T}\Phi_\rho \circ \rho_{f'+f_0}(x_i, h_f(x_i)) - \mathbb{E}_{x_i\sim S}\Phi_\rho \circ \rho_{f'+f_0}(x_i, h_f(x_i))\} \\ &= d^{(\rho)}_{f,f_0,\mathcal{F}}(S,T) \end{aligned} \tag{12}$$

By substituting Eq (12) into Eq (11), we finally reach

$$err_T(f) \le err^{(\rho)}_S(f) + d^{(\rho)}_{f,f_0,\mathcal{F}}(S,T) + \lambda \tag{13}$$

□

6 PROOF OF THEOREM 3

Lemma 2. *(Theorem 3.3, (Mohri et al., 2018)) Let $\mathcal{G}$ be a family of functions mapping $\mathcal{X} \in \mathbb{D} \to \mathbb{R}$. Then for any $\delta > 0$, with probability at least $1-\delta$ over the draw of i.i.d samples from sample S of size m, each of the following holds for all $g \in \mathcal{G}$*

$$\mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\Re_m(\mathcal{G}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \tag{14}$$

Lemma 3. *(Talagrand's lemma, (Mohri et al., 2018)) Let $\Psi_i : \mathbb{R} \to \mathbb{R}$ be an l-Lipschitz. Then for any hypothesis set $\mathcal{G}$ of real valued functions, and for any sample D of size n, the following inequality holds:*

$$\mathbb{E}_\delta \sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \delta_i(\Psi_i \circ g)(\mathbf{x}_i) \leq l\hat{\Re}_{n,\hat{D}}(\mathcal{G}) \tag{15}$$

Theorem 3. *Let $f_0 \in \mathcal{F}$ be a fixed source function that maps from $\mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ that is trained on source domain only which satisfies $\rho_{f_0}(\mathbf{x}^s, h_f) \geq \epsilon_s$ for source domain data $\mathbf{x}^s \in S$ and $\rho_{f_0}(\mathbf{x}^t, h_f) \leq \epsilon_t$ for target domain data as outliers $\mathbf{x}^t \in T$. $\mathbf{x}_i^s$ is an i.i.d sample of size m drawn from the source distribution S and $\mathbf{x}_i^s$ is an i.i.d sample of size n drawn from the target distribution T. Given the same settings as Definition (4.1). For any $\delta > 0$, with the probability at least $1-2\delta$, we have the following generalization error bound for domain discrepancy loss function*

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}^s\in S}[\log(\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f_0}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f'}(\mathbf{x}^s,h_f)}})] + \mathbb{E}_{\mathbf{x}^t\in T}[\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)+\rho_{f_0}(\mathbf{x}^t,h_f)}})] \\
&\leq \frac{1}{m}\sum_{i=1}^{m}\log(\frac{e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f_0}(\mathbf{x}_i^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f'}(\mathbf{x}_i^s,h_f)}}) + \frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}_i^t,h_f)+\rho_{f'}(\mathbf{x}_i^t,h_f)}}) \\
&+ \max\{\frac{2}{(e^{\epsilon_s}-1)\lambda_s^+ + 1}, \frac{2}{(e^{\epsilon_s}-1)\lambda_s^- + 1}\}\Re_{m,\mathcal{D}_s}(\mathcal{G}_s) \\
&+ \max\{\frac{2e^{\epsilon_t}}{(1-\lambda_t^+)e^{\epsilon_t}+\lambda_t^+}, \frac{2e^{\epsilon_t}}{(1-\lambda_t^-)e^{\epsilon_t}+\lambda_t^-}\}\Re_{n,\mathcal{D}_t}(\mathcal{G}_t) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}
\end{aligned} \tag{16}$$

where λ_s and λ_t is defined as

$$\begin{aligned}
\lambda_s^- &= \min\{\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)}}\}, \lambda_s^+ = \max\{\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)}}\}, \forall \mathbf{x}^s \in S \\
\lambda_t^- &= \min\{\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)}}\}, \lambda_t^+ = \max\{\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)}}\}, \forall \mathbf{x}^t \in T
\end{aligned} \tag{17}$$

Proof. We first define z_i^s as

$$z_i^s \triangleq \log(\frac{e^{\rho_{f'}(\mathbf{x}_i,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}_i,h_f)}}) \tag{18}$$

From the above Equation, we have

$$e^{\rho_{f'}(\mathbf{x}_i,h_f)} = \frac{e^{z_i^s}}{1-e^{z_i^s}} \tag{19}$$

Then by substituting the above equation into $\log(\frac{e^{\rho_{f'}(\mathbf{x},h_f)+\rho_{f_0}(\mathbf{x}_i,h_f)}}{1+e^{\rho_{f'}(\mathbf{x},h_f)+\rho_{f_0}(\mathbf{x}_i,h_f)}})$, we have

$$\log(\frac{e^{\rho_{f'}(\mathbf{x},h_f)+\rho_{f_0}(\mathbf{x}_i,h_f)}}{1+e^{\rho_{f'}(\mathbf{x},h_f)+\rho_{f_0}(\mathbf{x}_i,h_f)}}) = \log(\frac{e^{z_i^s+\rho_{f_0}(\mathbf{x}_i,h_f)}}{e^{z_i^s+\rho_{f_0}(\mathbf{x}_i,h_f)}+1-e^{z_i^s}}) \tag{20}$$

Define the following transformation function

$$\Gamma_i(z_i^s) = \log(\frac{e^{z_i^s+\rho_{f_0}(\mathbf{x}_i,h_f)}}{e^{z_i^s+\rho_{f_0}(\mathbf{x}_i,h_f)}+1-e^{z_i^s}}) \tag{21}$$

By Lemma 2, with probability at least $1-\delta$, for any $g_s \in \mathcal{G}_s$.

$$\mathbb{E}_{\mathbf{x}^s \in S}[\log(\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f_0}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f'}(\mathbf{x}^s,h_f)}})] - \frac{1}{m}\sum_{i=1}^{m}\log(\frac{e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f_0}(\mathbf{x}_i^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f'}(\mathbf{x}_i^s,h_f)}}) \leq 2\Re_{m,\mathcal{D}_s}(\Gamma_i \circ \mathcal{G}_s) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \quad (22)$$

Next we take gradient on Γ_i

$$\Gamma_i'(z_i^s) = \frac{1}{e^{z_i^s+\rho_{f_0}(\mathbf{x}_i,h_f)}+1-e^{z_i^s}} \quad (23)$$

From the definition of $z_i^s, \lambda_s, \epsilon_s$, we have

$$0 \leq e^{z_i^s} \leq 1, \quad \rho_{f_0}(\mathbf{x}_i,h_f) \geq \epsilon_s \quad (24)$$

Then we can bound Γ_i' by

$$0 \leq \Gamma_i'(z_i^s) \leq \frac{1}{(e^{\epsilon_s}-1)e^{z_i^s}+1} \quad (25)$$

As $e^{z_i^s}$ takes value between $[\lambda_s^-, \lambda_s^+]$, using the properties of linear functions, we have

$$0 \leq \Gamma_i'(z_i^s) \leq l_\Gamma = \max\{\frac{1}{(e^{\epsilon_s}-1)\lambda_s^-+1}, \frac{1}{(e^{\epsilon_s}-1)\lambda_s^++1}\} \quad (26)$$

Therefore Γ_i is l_Γ-Lipschitz. By applying the Lemma 3 into inequality (22), we have the following inequality holds with probability at least $1-\delta$

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}^s \in S}[\log(\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f_0}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f'}(\mathbf{x}^s,h_f)}})] - \frac{1}{m}\sum_{i=1}^{m}\log(\frac{e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f_0}(\mathbf{x}_i^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f'}(\mathbf{x}_i^s,h_f)}}) \leq 2\Re_{m,\mathcal{D}_s}(\Gamma_i \circ \mathcal{G}_s) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \\ &\leq \max\{\frac{2}{(e^{\epsilon_s}-1)\lambda_s^++1}, \frac{2}{(e^{\epsilon_s}-1)\lambda_s^-+1}\}\Re_{m,\mathcal{D}_s}(\mathcal{G}_s) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\end{aligned} \quad (27)$$

Similarly we define z_i^t

$$z_i^t \triangleq \log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}_i,h_f)}}) \quad (28)$$

From the above Equation, we have

$$e^{\rho_{f'}(\mathbf{x}_i,h_f)} = e^{-z_i^t}-1 \quad (29)$$

Then by substituting the above equation into $\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x},h_f)+\rho_{f_0}(\mathbf{x}_i,h_f)}})$, we have

$$\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x},h_f)+\rho_{f_0}(\mathbf{x}_i,h_f)}}) = \log(\frac{1}{e^{z_i^t+\rho_{f_0}(\mathbf{x}_i,h_f)}+1-e^{z_i^t}}) \quad (30)$$

Similarly we define the following transformation function

$$\Psi_i(z_i^t) = \log(\frac{1}{e^{z_i^t+\rho_{f_0}(\mathbf{x}_i,h_f)}+1-e^{z_i^t}}) \quad (31)$$

By Lemma 2, with probability at least $1-\delta$, for any $g_t \in \mathcal{G}_t$.

$$\mathbb{E}_{\mathbf{x}^t \in T}[\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)+\rho_{f_0}(\mathbf{x}^t,h_f)}})] - \frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}_i^t,h_f)+\rho_{f'}(\mathbf{x}_i^t,h_f)}}) \leq 2\Re_{n,\mathcal{D}_t}(\Psi_i \circ \mathcal{G}_t) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}} \quad (32)$$

Next we take gradient on Ψ_i

$$\Psi_i'(z_i^t) = \frac{e^{\rho_{f_0}(\mathbf{x}_i,h_f)}}{e^{\rho_{f_0}(\mathbf{x}_i,h_f)}-e^{\rho_{f_0}(\mathbf{x}_i,h_f)+z_i^t}+e^{z_i^t}} \quad (33)$$

From the definition of $z_i^t, \lambda_t, \epsilon_t$, we have

$$0 \leq e^{z_i^t} \leq 1, \quad \rho_{f_0}(\mathbf{x}_i,h_f) \leq \epsilon_t \quad (34)$$

Then we can bound Ψ_i' by

$$0 \leq \Psi_i'(z_i^t) \leq \frac{e^{\epsilon_t}}{(1-e^{z_i^t})e^{\epsilon_t}+e^{z_i^t}} \tag{35}$$

As e^{ϵ_t} takes value between $[\lambda_t^-, \lambda_t^+]$, using properties of linear functions, we have

$$0 \leq \Psi_i'(z_i^t) \leq l_\Psi = \max\{\frac{e^{\epsilon_t}}{(1-\lambda_t^-)e^{\epsilon_t}+\lambda_t^-}, \frac{e^{\epsilon_t}}{(1-\lambda_t^+)e^{\epsilon_t}+\lambda_t^+}\} \tag{36}$$

Therefore Ψ_i is l_Ψ-Lipschitz. By applying the Lemma 3 into Inequality (32), we have the following inequality holds with probability at least $1-\delta$

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}^t \in T}[\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)+\rho_{f_0}(\mathbf{x}^t,h_f)}})] - \frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}_i^t,h_f)+\rho_{f'}(\mathbf{x}_i^t,h_f)}}) \leq 2\Re_{n,\mathcal{D}_t}(\Psi_i \circ \mathcal{G}_t) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}} \\
&\leq \max\{\frac{2e^{\epsilon_t}}{(1-\lambda_t^-)e^{\epsilon_t}+\lambda_t^-}, \frac{2e^{\epsilon_t}}{(1-\lambda_t^+)e^{\epsilon_t}+\lambda_t^+}\}\Re_{n,\mathcal{D}_t}(\mathcal{G}_t) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}
\end{aligned} \tag{37}$$

By summing up Equation (37) with (27), we have the following inequality holds with probability at least $1-2\delta$

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}^s \in S}[\log(\frac{e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f_0}(\mathbf{x}^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}^s,h_f)+\rho_{f'}(\mathbf{x}^s,h_f)}})] + \mathbb{E}_{\mathbf{x}^t \in T}[\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}^t,h_f)+\rho_{f_0}(\mathbf{x}^t,h_f)}})] \\
&\leq \frac{1}{m}\sum_{i=1}^{m}\log(\frac{e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f_0}(\mathbf{x}_i^s,h_f)}}{1+e^{\rho_{f'}(\mathbf{x}_i^s,h_f)+\rho_{f'}(\mathbf{x}_i^s,h_f)}}) + \frac{1}{n}\sum_{i=1}^{n}\log(\frac{1}{1+e^{\rho_{f'}(\mathbf{x}_i^t,h_f)+\rho_{f'}(\mathbf{x}_i^t,h_f)}}) \\
&+ \max\{\frac{2}{(e^{\epsilon_s}-1)\lambda_s^+ + 1}, \frac{2}{(e^{\epsilon_s}-1)\lambda_s^- + 1}\}\Re_{m,\mathcal{D}_s}(\mathcal{G}_s) \\
&+ \max\{\frac{2e^{\epsilon_t}}{(1-\lambda_t^+)e^{\epsilon_t}+\lambda_t^+}, \frac{2e^{\epsilon_t}}{(1-\lambda_t^-)e^{\epsilon_t}+\lambda_t^-}\}\Re_{n,\mathcal{D}_t}(\mathcal{G}_t) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}
\end{aligned} \tag{38}$$

which completes the proof □

7 PROOF OF PROPOSITION 1

Proposition 1. *Consider the following optimization problem we have defined*

$$\max_{f'} \mathbb{E}_{\hat{S}} \log(\sigma_{h_f} \circ f') + \mathbb{E}_{\hat{T}} \log(1 - \sigma_{h_f} \circ f') \tag{39}$$

$$\min_{\hat{S},\hat{T}} \mathbb{E}_{\hat{S}} \log(\frac{1}{2}\sigma_{h_f} \circ f' + \frac{1}{2}\sigma_{h_f} \circ f_0) + \mathbb{E}_{\hat{T}} \log(1 - \frac{1}{2}\sigma_{h_f} \circ f' - \frac{1}{2}\sigma_{h_f} \circ f_0) \tag{40}$$

Assume that there is no restriction on the choice of f'. By fixing f_0, we have the following two results.

1. *The optimal value of $\sigma_{h_f} \circ f'$ on data x is*

$$\frac{\hat{p}_s(\mathbf{x})}{\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})} \tag{41}$$

   where $\hat{p}_s(\mathbf{x})$ and $\hat{q}_t(\mathbf{x})$ are density functions of source and target domain empirical distributions

2. *The minimization problem w.r.t S and T is equivalent to minimization on the sum of two terms L_1 and L_2, where*

$$\begin{aligned}
L_1 = &4KL(\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}||\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}) + 4KL(\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}||\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}) \\
&+ 4KL(\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}||\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}) + 4KL(\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}||\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T})
\end{aligned} \tag{42}$$

is a symmetric distribution divergence between $\hat{S}$ and $\hat{T}$ and has global minimum of $\hat{S} = \hat{T}$

$$L_2 = \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \frac{1}{4 - (\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}))^2/(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2} d\mathbf{x} \tag{43}$$

is a re-weighted bounds on the total variations between $\hat{p}_s(\mathbf{x})$ and $\hat{q}_t(\mathbf{x})$

Proof. For maximization w.r.t target adaptation domain discriminator f', we have

$$\begin{aligned} &\mathbb{E}_{\hat{S}} \log(\sigma_{h_f} \circ f') + \mathbb{E}_{\hat{T}} \log(1 - \sigma_{h_f} \circ f') \\ =&\int_{\mathbf{x}} \hat{p}_s(\mathbf{x}) \log(\sigma_{h_f} \circ f') + \hat{q}_t(\mathbf{x}) \log(1 - \sigma_{h_f} \circ f') d\mathbf{x} \end{aligned} \tag{44}$$

As we relaxed the restriction on $\sigma_{h_f} \circ f'$, we could find that the maximization of $p(x) \log(\sigma_{h_f} \circ f_t) + q(x) \log(1 - \sigma_{h_f} \circ f')$ could be satisfied on every $x \in \mathbb{D}$ as $\sigma_{h_f} \circ f'$ reaches

$$\sigma_{h_f} \circ f'(\mathbf{x}) = \frac{\hat{p}_s(\mathbf{x})}{\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})} \tag{45}$$

The above optimal value of $\sigma_{h_f} \circ f'(\mathbf{x})$ could be derived from simple calculus.

Then we analyze the maximization bounds w.r.t $\hat{S}$ and $\hat{T}$ on the equilibrium condition of target adaptation domain discriminator f'. By substituting the equilibrium condition of (45) into (40)

$$\begin{aligned} D =&\mathbb{E}_{\hat{S}} \log(\frac{1}{2}\sigma_{h_f} \circ f' + \frac{1}{2}\sigma_{h_f} \circ f_0) + \mathbb{E}_{\hat{T}} \log(1 - \frac{1}{2}\sigma_{h_f} \circ f_0 - \frac{1}{2}\sigma_{h_f} \circ f') \\ =&\mathbb{E}_{\hat{S}} \log(\frac{\hat{S}}{2(\hat{S} + \hat{T})} + \frac{1}{2}\sigma_{h_f} \circ f_0) + \mathbb{E}_{\hat{T}} \log(\frac{1}{2} - \frac{1}{2}\sigma_{h_f} \circ f_0 + \frac{\hat{T}}{2(\hat{S} + \hat{T})}) \end{aligned} \tag{46}$$

Using first order Taylor expansion, we have

$$\begin{aligned} D = &\underbrace{\mathbb{E}_{\hat{S}} \log(\frac{1}{4} + \frac{\hat{S}}{2(\hat{S} + \hat{T})}) + \mathbb{E}_{\hat{T}} \log(\frac{1}{4} + \frac{\hat{T}}{2(\hat{S} + \hat{T})})}_{L_1} \\ &\underbrace{-\mathbb{E}_{\hat{S}} \frac{4(\hat{S} + \hat{T})}{3\hat{S} + \hat{T}}(\frac{1}{4} - \frac{1}{2}\sigma_{h_f} \circ f_0) + \mathbb{E}_{\hat{T}} \frac{4(\hat{S} + \hat{T})}{3\hat{T} + \hat{S}}(\frac{1}{4} - \frac{1}{2}\sigma_{h_f} \circ f_0)}_{L_2} \end{aligned} \tag{47}$$

As we depose the D into term L_1 and L_2, we could further write L_1 as

$$\begin{aligned} L_1 =& \mathbb{E}_{\hat{S}} \log(\frac{3\hat{S} + \hat{T}}{4(\hat{S} + \hat{T})}) + \mathbb{E}_{\hat{T}} \log(\frac{3\hat{T} + \hat{S}}{4(\hat{S} + \hat{T})}) \\ =& -4\mathbb{E}_{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}} \log(\frac{3\hat{S} + \hat{T}}{4(\hat{S} + \hat{T})}) + 4\mathbb{E}_{\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}} \log(\frac{3\hat{S} + \hat{T}}{4(\hat{S} + \hat{T})}) \\ &- 4\mathbb{E}_{\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}} \log(\frac{3\hat{T} + \hat{S}}{4(\hat{S} + \hat{T})})) + 4\mathbb{E}_{\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}} \log(\frac{3\hat{T} + \hat{S}}{4(\hat{S} + \hat{T})}) \\ =& -4\mathbb{E}_{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}} \log(\frac{1}{8}\frac{\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}}{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}}) + 4\mathbb{E}_{\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}} \log(\frac{1}{8}\frac{\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}}{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}}) \\ &- 4\mathbb{E}_{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}} \log(\frac{1}{8}\frac{\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}}{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}}) + 4\mathbb{E}_{\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}} \log(\frac{1}{8}\frac{\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}}{\frac{1}{2}\hat{S} + \frac{1}{2}\hat{T}}) \\ =& 4KL(\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}||\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}) + 4KL(\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}||\frac{3}{4}\hat{T} + \frac{1}{4}\hat{S}) \\ &+ 4KL(\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}||\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}) + 4KL(\frac{1}{2}\hat{T} + \frac{1}{2}\hat{S}||\frac{3}{4}\hat{S} + \frac{1}{4}\hat{T}) \end{aligned} \tag{48}$$

Next, the term L_2 could be treated as

$$\begin{aligned}
L_2 &= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x})) \frac{\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})}{3\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})} \hat{p}_s(\mathbf{x}) d\mathbf{x} - \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x})) \frac{\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})}{3\hat{q}_t(\mathbf{x}) + \hat{p}_s(\mathbf{x})} \hat{q}_t(\mathbf{x}) d\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))(-\frac{\hat{p}_s(\mathbf{x})}{3\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})} + \frac{\hat{q}_t(\mathbf{x})}{3\hat{q}_t(\mathbf{x}) + \hat{p}_s(\mathbf{x})}) d\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x})) \frac{\hat{q}_t(\mathbf{x})^2 - \hat{p}_s(\mathbf{x})^2}{(3\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))(3\hat{q}_t(\mathbf{x}) + \hat{p}_s(\mathbf{x}))} d\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \frac{(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2}{(3\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))(3\hat{q}_t(\mathbf{x}) + \hat{p}_s(\mathbf{x}))} d\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \frac{(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2}{3\hat{p}_s(\mathbf{x})^2 + 3\hat{q}_t(\mathbf{x})^2 + 10\hat{p}_s(\mathbf{x})\hat{q}_t(\mathbf{x})} d\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \frac{(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2}{4(\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2 - (\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}))^2} d\mathbf{x} \\
&= \int_{\mathbf{x}} (1 - 2\sigma_{h_f} \circ f_0(\mathbf{x}))(\hat{q}_t(\mathbf{x}) - \hat{p}_s(\mathbf{x})) \frac{1}{4 - (\hat{p}_s(\mathbf{x}) - \hat{q}_t(\mathbf{x}))^2 / (\hat{p}_s(\mathbf{x}) + \hat{q}_t(\mathbf{x}))^2} d\mathbf{x}
\end{aligned} \tag{49}$$

□

References

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

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

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

W. Hu, M. Wang, Q. Qin, J. Ma, and B. Liu. Hrn: A holistic approach to one class learning. *Advances in neural information processing systems*, 33:19111–19124, 2020.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.