Learning Unforgotten Domain-Invariant Representations for Online Unsupervised Domain Adaptation

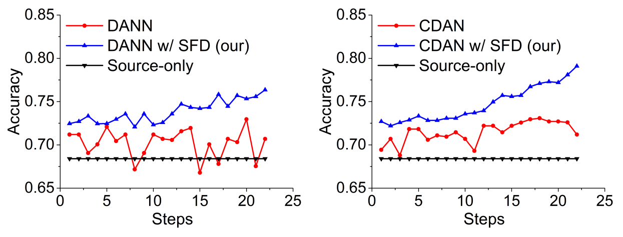
Cheng Feng[1] , **Chaoliang Zhong**[1] , **Jie Wang**[1] , **Ying Zhang**[1] , **Jun Sun**[1] and **Yasuto Yokota**[2]
[1]Fujitsu R&D Center, Co., LTD
[2]Fujitsu LTD
{fengcheng, clzhong, wwangjie, zhangying, sunjun, yokota.yasuto}@fujitsu.com

Abstract

Existing unsupervised domain adaptation (UDA) studies focus on transferring knowledge in an offline manner. However, many tasks involve online requirements, especially in real-time systems. In this paper, we discuss Online UDA (OUDA) which assumes that the target samples are arriving sequentially as a small batch. OUDA tasks are challenging for prior UDA methods since online training suffers from catastrophic forgetting which leads to poor generalization. Intuitively, a good memory is a crucial factor in the success of OUDA. We formalize this intuition theoretically with a generalization bound where the OUDA target error can be bounded by the source error, the domain discrepancy distance, and a novel metric on forgetting in continuous online learning. Our theory illustrates the tradeoffs inherent in learning and remembering representations for OUDA. To minimize the proposed forgetting metric, we propose a novel source feature distillation (SFD) method which utilizes the source-only model as a teacher to guide the online training. In the experiment, we modify three UDA algorithms, i.e., DANN, CDAN, and MCC, and evaluate their performance on OUDA tasks with real-world datasets. By applying SFD, the performance of all baselines is significantly improved.

1 Introduction

Unsupervised domain adaptation (UDA) aims to transfer the knowledge from a labeled source domain to an unlabeled target domain. Due to the ability of deep neural networks, UDA studies [Ganin *et al.*, 2016; Long *et al.*, 2017; Zhang *et al.*, 2019] have achieved good performance in a wide range of applications by learning domain-invariant representations. However, current UDA methods are in an offline manner which heavily relies on sample-rich target domains and adequate training time while real-world applications may involve online requirements, especially in real-time systems. In this paper, we discuss Online UDA (OUDA) which assumes that the target samples are arriving sequentially as a small batch rather than a static sample-rich target domain.



(a) DANN in OUDA (b) CDAN in OUDA

Figure 1: Catastrophic forgetting in OUDA. The experiments are conducted on Office-31, A-W task. Y-axis represents the accuracy on the whole target domain while X-axis represents the steps in online training. For each step, the algorithms obtain 36 unlabeled target samples and the training epoch is set as 20. Both DANN and CDAN suffer from catastrophic forgetting while SFD (a novel method for reducing forgetting) can significantly improve generalization with little extra time cost, i.e., about 0.15 seconds per epoch.

OUDA [Moon *et al.*, 2020] is composed of continuous learning steps. For each step, OUDA algorithm inherits the model generated in the previous step and updates the model with a small batch of unlabeled target data currently arrived. A crucial challenge in OUDA is to link new knowledge to old, i.e., learning while resisting forgetting. Although UDA has been extensively explored, few studies focus on OUDA. Existing OUDA studies [Moon *et al.*, 2020; Hajifar and Sun, 2021] are majorly driven by applications while lacking theoretical guarantees. Existing UDA theories [Ben-David *et al.*, 2007; Blitzer *et al.*, 2007; Zhao *et al.*, 2019] are also insufficient to cover OUDA which suffers from catastrophic forgetting. There exists a strong need to understand why and how to resist forgetting in OUDA tasks.

To measure the forgetting across different steps, we propose a novel metric, i.e., generator discrepancy. Then a novel generalization bound has been proved where the OUDA target error can be bounded by the proposed generator discrepancy and other measurable items based on the data currently collected. Our bound shows insights on providing theoretical guarantees for OUDA without the access of the data collected in previous steps while illustrating the tradeoffs inherent in learning and remembering representations. For example, in an extreme case when the generators in all steps share the same parameters, then we can ensure a zero generator discrepancy. However, in this case, generators cannot obtain any new knowledge since the parameters are fixed. In another hand, a large generator discrepancy can also lead to poor generalization since the OUDA target error is also bounded by the generator discrepancy. We conduct experiments to verify this conclusion as shown in Figure 1 where both DANN [Ganin *et al.*, 2016] and CDAN [Long *et al.*, 2017] apply no constraints on the generator discrepancy. Both methods suffer from poor generalization or catastrophic forgetting.

To minimize the proposed generator discrepancy, a Source Feature Distillation (SFD) method has been proposed which utilizes the source-only model as a teacher to guide OUDA algorithms. In the experiments, we choose two classical UDA methods, i.e., DANN [Ganin *et al.*, 2016] and CDAN [Long *et al.*, 2017] and one state-of-the-art (SOTA) UDA method [Jin *et al.*, 2020] as the baselines. The experiments are conducted with real-world datasets, i.e., Office-31, Office-home and ImageCLEF-DA. It is observed that all baselines suffer from catastrophic forgetting in OUDA while SFD can significantly reduce forgetting with little extra cost as shown in Figure 1. The contributions of this paper are as follows:

1. We propose a novel metric for measuring forgetting in OUDA and show a novel target error upper bound for OUDA tasks with the proposed forgetting metric.
2. Based on the theoretical results, we propose a novel SFD method that utilizes the source-only model as a teacher to guide the algorithms in OUDA tasks.
3. To verify the effectiveness of the theoretical results and the proposed SFD, we conduct experiments with several SOTA UDA algorithms on real-world OUDA tasks.

2 Realated Work

The goal for domain adaptation [Pan and Yang, 2009] is to generalize a learner across different domains of different distributions. Ben et al. [Ben-David *et al.*, 2007] and Blitzer et al. [Blitzer *et al.*, 2007] introduce a seminal theoretical model where the target error can be bounded by source risk and a divergence measure, known as the $\mathcal{H}$-divergence. Based on these theoretical results, adversarial domain adaptation methods [Ganin *et al.*, 2016; Long *et al.*, 2017; Tzeng *et al.*, 2017; Mao *et al.*, 2017] learn domain-invariant representations by applying a domain discriminator to predict if the feature is from the source or the target. Then the feature extractor is trained both to maximize the discriminator and minimize the classification loss in the source domain. However, existing UDA studies [Ben-David *et al.*, 2007; Blitzer *et al.*, 2007; Shu *et al.*, 2018; Zhao *et al.*, 2019; Tachet des Combes *et al.*, 2020] are in an offline manner which is insufficient to cover OUDA which suffers from catastrophic forgetting.

Online algorithms [Kirkpatrick *et al.*, 2017; McMahan *et al.*, 2013] assume that the target data are arriving sequentially as a small batch. As the algorithms have no access to the target arrived in previous steps, online algorithms suffer from catastrophic forgetting. EWC [Kirkpatrick *et al.*, 2017] remembers old tasks by selectively slowing down learning on the weights important for those tasks. However, existing online studies [McMahan *et al.*, 2013] or continual learning [Rolnick *et al.*, 2019; Kirkpatrick *et al.*, 2017; Lopez-Paz and Ranzato, 2017] are in a supervised manner which cannot cover OUDA.

Although both UDA and online learning have been extensively explored, few studies focus on OUDA. Existing OUDA studies [Mancini *et al.*, 2018; Moon *et al.*, 2020; Hajifar and Sun, 2021] are majorly application-driven. [Moon *et al.*, 2020] proposes an intuitive multi-step framework that computes the mean-target subspace by the geometrical interpretation of the euclidean space. [Mancini *et al.*, 2018] focuses on OUDA robotic kitting in unconstrained scenarios while [Hajifar and Sun, 2021] studies the continuous cross-subject liver viability evaluation. The theoretical analysis for OUDA is still vacant.

3 Method

In this section, we formalize the problem of OUDA and describe our method for addressing the problem. Theoretical guarantees for the proposed method are shown in Sec. 4.

3.1 Problem Definition for OUDA

In this paper, we focus on the problems concerning k-class classification in OUDA. Basically, we follow the standard notations in this field [Ben-David *et al.*, 2007; Blitzer *et al.*, 2007]. Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output space. Let $\mathcal{D}_S$ (resp. $\mathcal{D}_T$) be the source domain (resp. target domain) with a marginal data distribution $\mathcal{P}_S(x)$ (resp. $\mathcal{P}_T(x)$) and a marginal label distribution $\mathcal{P}_S(y)$ (resp. $\mathcal{P}_T(y)$). The error of a classifier h with a feature generator g in $\mathcal{D}_S$ is defined as:

$$\epsilon_S(h \circ g) \triangleq \int_{\mathcal{X}} \mathcal{L}(h(g(x)), f_S(g(x)))d\mathcal{P}_S(x) \tag{1}$$

where $\mathcal{L}$ is a loss function $\mathcal{L} : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_+$ and $\mathcal{L}$ satisfies the triangle inequality. f_S is the label function in $\mathcal{D}_S$. In the same way, we define the target error $\epsilon_T(h \circ g)$.

In OUDA, $\mathcal{D}_T$ is composed of a sequential target domains. For each step t, the algorithm collects a small batch of samples, i.e., $\mathcal{D}_T^t$. Assume that both the labeled source samples drown from $\mathcal{D}_S$ and the unlabeled target samples drown from $\mathcal{D}_T^t$ are i.i.d.. We define g^t (resp. h^t) as the feature generator (resp. classifier) learned in step t. Then we define $\epsilon_T^i(h^t \circ g^t)$ as the error of g^t and h^t in the i-th target domain, i.e., $\mathcal{D}_T^i$ where $i \leq t$. For each step t, the goal of OUDA is to minimize ϵ_T^{t-all} where

$$\epsilon_T^{t-all} \triangleq \sum_{i=1}^{t} \epsilon_T^i(h^t \circ g^t) \tag{2}$$

It's noteworthy that $\forall i < t$, $\epsilon_T^i(h^t \circ g^t)$ is an immeasurable item since OUDA algorithms have no access of the target data in previous steps, i.e., $\mathcal{D}_T^i, \forall i < t$.

3.2 Domain-Invariant Representation

To perform adaptation in an online manner, we leverage the idea of learning domain-invariant representations with domain adversarial training [Ganin *et al.*, 2016; Long *et al.*, 2017]. Domain-adversarial training methods are based on a three-player game that includes a feature generator g, a classifier h trained on source labels, and a discriminator d which

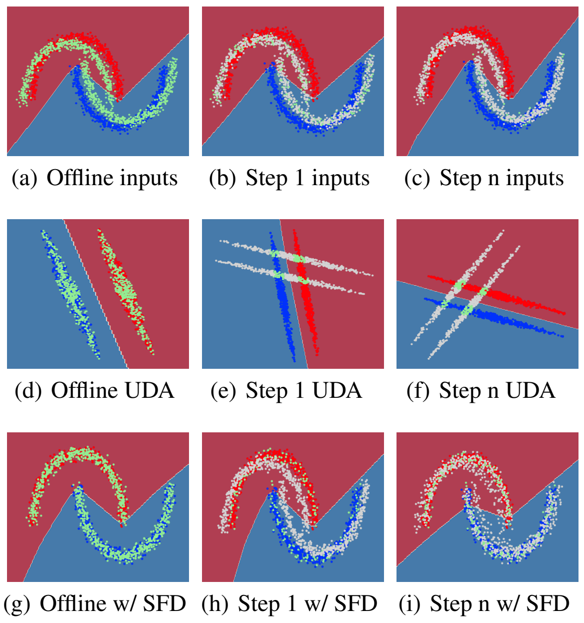


Figure 2: Motivation of learning stable source features in OUDA. Red and blue points are source samples with two classes. Green points are currently collected target samples while gray points are unavailable target samples, i.e., uncollected or collected in previous steps. Figures (a-c) show the inputs in OUDA where the target data is generated by rotating the source samples. Figures (d-e) show the features generated by offline UDA methods. Figures (g-i) show the features generated by offline UDA methods combined with SFD.

aims to infer the domain index of the features. In the three-player game, g aims to learn discriminative features for h while fooling d to prevent it from inferring the domain index. Formally, domain-adversarial training methods perform a minimax optimization with a total loss:

$$\mathcal{L}_{UDA} = \mathcal{L}_c - \lambda \mathcal{L}_d \tag{3}$$

where we have

$$\begin{aligned} \mathcal{L}_c &\triangleq \mathbb{E}_{x\sim\mathcal{D}_S}[\mathcal{L}_c(h(g(x)), y)] \\ \mathcal{L}_d &\triangleq \mathbb{E}_{x\sim\mathcal{D}_S \bigcup \mathcal{D}_T}[\mathcal{L}_d(g(x), y_d)] \end{aligned} \tag{4}$$

where y_d is domain label, e.g., 0 for source and 1 for target.

In OUDA, algorithms have to learn representations from a sample-rich source domain and sequential sample-poor target domains as shown in Figure 2 (a-c). By playing the three-player game with the whole target inputs, UDA methods could learn domain-invariant representations with good generalization as shown in Figure 2 (d). However, in OUDA, small $\mathcal{L}_c$ and large $\mathcal{L}_d$ are not sufficient to get such representations as shown in Figure 2 (e-f). Another critical challenge in OUDA is catastrophic forgetting which makes the situation worse when learning representations from Figure 2 (e) to Figure 2 (f). By applying a constraint on stable source representations as shown in 2 (h-i), we can gradually learn unforgotten domain-invariant target representations.

3.3 Learning Unforgotten Representations

In this paper, to get unforgotten representations in OUDA tasks, we apply a constraint on stable source features, i.e., $\mathcal{L}_{sfd}$. Theoretical explanation for the effectiveness of $\mathcal{L}_{sfd}$ is provided in Sec. 4. Then the total loss for OUDA,

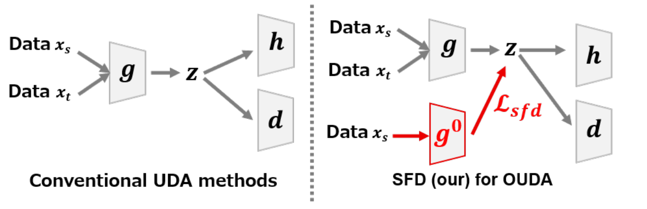


Figure 3: Framework of SFD. z is a feature space generated by a feature generator g. h and d are the source classifier and domain discriminator. g^0 is the source-only feature generator.

$$\mathcal{L}_{OUDA} = \mathcal{L}_{UDA} + \alpha \mathcal{L}_{sfd} \tag{5}$$

where α is a hyper-parameter and we have,

$$\mathcal{L}_{sfd} \triangleq \sum_{i=1}^{t-1} \mathbb{E}_{x\sim\mathcal{D}_S}[\mathcal{L}(g^t(x), g^i(x))] \tag{6}$$

where g^t is the feature generator to be learned in current t-th step and g^i is the previous feature generator in the i-th step. Computing $\mathcal{L}_{sfd}$ is at a high cost since it's related to the feature generators in all previous steps. To simplify the calculation, we show a upper bound for $\mathcal{L}_{sfd}$. Let g^0 be the source-only model. As $\mathcal{L}$ satisfies the triangle inequality,

$$\begin{aligned} \mathcal{L}_{sfd} &\triangleq \sum_{i=1}^{t-1} \mathbb{E}_{x\sim\mathcal{D}_S}[\mathcal{L}(g^t(x), g^i(x))] \\ &\leq \sum_{i=1}^{t-1} \mathbb{E}_{x\sim\mathcal{D}_S}[\mathcal{L}(g^t(x), g^0(x)) + \mathcal{L}(g^i(x), g^0(x))] \\ &= (t-1)\mathbb{E}_{x\sim\mathcal{D}_S}[\mathcal{L}(g^t(x), g^0(x))] + \delta \end{aligned} \tag{7}$$

where $\delta = \sum_{i=1}^{t-1} \mathbb{E}_{x\sim\mathcal{D}_S}[\mathcal{L}(g^i(x), g^0(x))]$.

It's noteworthy that δ is a small value since it has been minimized in previous steps and the only concern is the first item which corresponds to a distillation loss between g^0 and g^t in the source feature space, i.e., the proposed SFD. Let $\mathcal{L}$ be an L2 distance function, then we have:

$$\mathcal{L}_{sfd} \leq (t-1)\mathbb{E}_{x\sim\mathcal{D}_S}[\left\|g^t(x) - g^0(x)\right\|_2^2] + \delta \tag{8}$$

Then the minimization of $\mathcal{L}_{sfd}$ equals to the minimization of $\mathbb{E}_{x\sim\mathcal{D}_S}[\left\|g^t(x) - g^0(x)\right\|_2^2]$. As shown in Figure 3, our method can be applied to any UDA method which learns domain-invariant representations, i.e., a feature generator which induces similar distributions across two domains. In the experiment part, we verify the effectiveness of our method with both domain-adversarial training methods [Ganin *et al.*, 2016; Long *et al.*, 2017] and non-domain-adversarial training method, MCC [Jin *et al.*, 2020]. The experimental results in Sec. 5 show that our method can significantly reduce forgetting which contributes to the improvement of the performance in all baselines.

4 Theoretical Results for OUDA

In this section, we show the theoretical results for OUDA.

4.1 Preliminaries

Lemma 1. *Let $\mathcal{H}$ be a hypothesis space in a input space $\mathcal{X}$ with any distribution $\mathcal{D}$. Assume that $\mathcal{L}$ satisfies the triangle inequality. Then for any $h, h^{'}, h^{''} \in \mathcal{H}$, we have (Proof shown in Appendix A.1)*

$$\epsilon_{\mathcal{D}}(h, h^{'}) \leq \epsilon_{\mathcal{D}}(h, h^{''}) + \epsilon_{\mathcal{D}}(h^{''}, h^{'}) \tag{9}$$

Definition 1. *(Discrepancy Distance $\mathcal{H}\triangle\mathcal{H}$):Let $\mathcal{H}$ be a hypothesis space in a feature space $\mathcal{Z}$. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two distributions on $\mathcal{Z}$. Then the discrepancy distance between $\mathcal{P}_1$ and $\mathcal{P}_2$ is:*

$$d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_1, \mathcal{P}_2\right) \triangleq 2 \sup_{h,h^{'} \in \mathcal{H}} \left| \epsilon_{\mathcal{P}_1}(h, h^{'}) - \epsilon_{\mathcal{P}_2}(h, h^{'}) \right| \tag{10}$$

Remark. Based on **Lemma 1** and $\mathcal{H}\triangle\mathcal{H}$, [Ben-David *et al.*, 2007] and [Blitzer *et al.*, 2007] proved the following generalization bound on the target error risk.

Theorem 1. *[Ben-David* et al.*, 2007; Blitzer* et al.*, 2007] Let $\mathcal{H}$ be a hypothesis space. Then $\forall h \in \mathcal{H}$,*

$$\epsilon_T(h \circ g) \leq \epsilon_S(h \circ g) + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_S(g(x)), \mathcal{P}_T(g(x))\right) + \lambda^* \tag{11}$$

where $\lambda^* = \min_{h^* \in \mathcal{H}} \epsilon_S(h^* \circ g) + \epsilon_T(h^* \circ g)$

Remark. The first item is the source error while the second item is the domain discrepancy distance across two domains. The third item λ^* is assumed a small const.

4.2 A Generalization Bound for OUDA

In each step t, the goal of OUDA is to minimize ϵ_T^{t-all} where

$$\epsilon_T^{t-all} \triangleq \sum_{i=1}^{t} \epsilon_T^i(h^t \circ g^t) \tag{12}$$

Definition 2. *(Generator Discrepancy $\mathcal{H}\triangle\mathcal{H}$): Let $\mathcal{X}$ (resp. $\mathcal{H}$) be a input (resp. hypothesises) space. Let g^1 and g^2 be two generators,*

$$g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^2\right) \triangleq \sup_{\mathcal{P} \in \mathcal{X}} d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}(g^1(x)), \mathcal{P}(g^2(x))\right) \tag{13}$$

Remark. $g_{\mathcal{H}\triangle\mathcal{H}}$ is first proposed in this paper for measuring forgetting across OUDA steps. It's noteworthy that $g_{\mathcal{H}\triangle\mathcal{H}}$ is measured by a common input distribution $\mathcal{P}$ with two generators while $d_{\mathcal{H}\triangle\mathcal{H}}$ in **Theorem 1** is measured by two input distributions with a common generator.

Theorem 2. *(Upper bound for OUDA): Let $\mathcal{H}$ be a hypothesis space. Let $\mathcal{D}_T^t$ be the t-th target domain with a marginal data distribution of $\mathcal{P}_T^t(x)$. Then $\forall g^t, h^t$ (Proof shown in Appendix A.2),*

$$\begin{aligned} \epsilon_T^{t-all} \leq & \underbrace{t\epsilon_S(h^t \circ g^t)}_{\text{Source Error}} + \underbrace{\sum_{i=1}^{t} \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^i(x))\right)}_{\text{Domain Discrepancy}} \\ & + \underbrace{\sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i)}_{\text{Generator Discrepancy}} + \underbrace{\sum_{i=1}^{t} \lambda_t^*(i)}_{\text{Shared Error}} \end{aligned} \tag{14}$$

where $\lambda_t^*(i) = \min_{h^* \in \mathcal{H}} \epsilon_T^i(h^* \circ g^t) + \epsilon_S(h^* \circ g^t)$

Remark. It's noteworthy that although ϵ_T^{t-all} includes immeasurable items, i.e., $\forall i < t$, $\epsilon_T^i(h^t \circ g^t)$, all items in **Theorem 2** are measurable without the access of the data collected in previous steps. For the t-th step, the ***Source Error*** in **Theorem 2** can be minimized by the source labels. For the ***Domain Discrepancy***, it's noteworthy that $\forall i < t$, $d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^i(x))\right)$ has been minimized in previous i-th step since such it's about the i-th generator and the i-th target domain. Then the only concern is the measurable domain discrepancy distance between the source domain $\mathcal{D}_S$ and the target domain currently arrived $\mathcal{D}_T^t$, i.e., $d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_S(g^t(x)), \mathcal{P}_T^t(g^t(x))\right)$. For the ***Generator Discrepancy***, it requires the current t-th generator g^t sharing small gaps with all previous generators $g^i, i \leq t$. The proposed SFD method is to reduce $\sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i)$, details please refer to Sec. 4.3. For the ***Shared Error***, $\lambda_t^*(i)$ is assumed a small const, $\forall i \leq t$.

4.3 Minimizing Generator Discrepancy with SFD

To minimize $g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i)$, $\forall i \leq t$, a proper way is to minimize the feature movements between g^t and g^i with the same dataset drown from a distribution as wide as possible. It's noteworthy that the proposed generator discrepancy is not binding with certain distributions. As the target samples are used for obtaining new knowledge, we apply a constraint on stable source features, i.e., $\mathcal{L}_{sfd}$ to minimize generator discrepancy. However, there is still a gap between the ***Generator Discrepancy***, i.e., $\sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i)$ and the simplified $\mathcal{L}_{sfd}$ which minimizes $\mathbb{E}_{x \sim \mathcal{D}_S}[\left\|g^t(x) - g^0(x)\right\|_2^2]$.

Lemma 2. *Let $\mathcal{H}$ be a hypothesis space in a feature space $\mathcal{Z}$. Then $\forall \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3 \in \mathcal{Z}$, we have*

$$d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_1, \mathcal{P}_2\right) \leq d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_1, \mathcal{P}_3\right) + d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_3, \mathcal{P}_2\right) \tag{15}$$

Lemma 3. *Let $\mathcal{X}$ (resp. $\mathcal{H}$) be a input (resp. hypothesises) space. Then $\forall g^1, g^2, g^3$ (Proof shown in Appendix A.3),*

$$g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^2\right) \leq g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^3\right) + g_{\mathcal{H}\triangle\mathcal{H}}\left(g^3, g^2\right) \tag{16}$$

According to **Lemma 3**, we have

$$\sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}\left(g^t, g^i\right) \leq t g_{\mathcal{H}\triangle\mathcal{H}}\left(g^t, g^0\right) + \sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}\left(g^i, g^0\right) \tag{17}$$

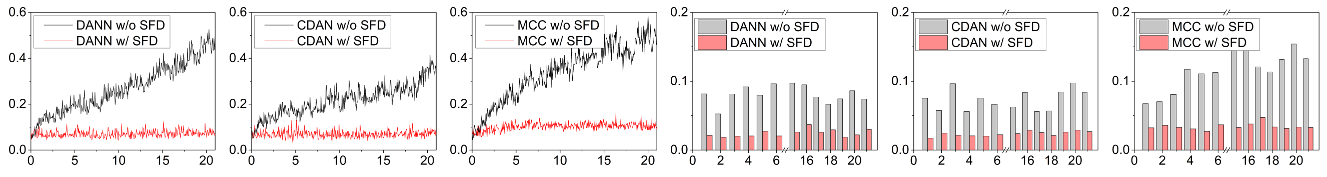
By minimizing the simplified $\mathcal{L}_{sfd}$, we can achieve the reduction of $g_{\mathcal{H}\triangle\mathcal{H}}\left(g^t, g^0\right)$ which also provides guarantees on the reduction of $\sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i)$ as $\forall i < t$, $g_{\mathcal{H}\triangle\mathcal{H}}\left(g^i, g^0\right)$ has been reduced in the previous i-th step.

4.4 Summary

By introducing the generator discrepancy $g_{\mathcal{H}\triangle\mathcal{H}}$, our theory provides theoretical explanations on why catastrophic forgetting leads to poor generalization in OUDA tasks. The proposed **Theorem 2** also shows insights on providing theoretical guarantees for OUDA only by utilizing measurable items with the target data currently collected. Furthermore, our theory illustrates the tradeoffs inherent in learning and remembering representations for OUDA tasks since reducing the generator discrepancy $g_{\mathcal{H}\triangle\mathcal{H}}$ is at a cost of weakening the learning ability. We believe these insights are helpful in guiding the future design of OUDA algorithms.

Method	**Office-Home**												
	A-C	A-P	A-R	C-A	C-P	C-R	P-A	P-C	P-R	R-A	R-C	R-P	Avg
SO [He *et al.*, 2016]	40.2	62.4	71.2	42.5	56.4	57.3	42.6	30.6	67.6	61.2	38.8	74.0	53.7
Metric	Accuracy (%) of last step												
DANN[Ganin *et al.*, 2016]	34.1	54.9	63.2	40.2	52.1	54.1	37.5	32.9	62.6	53.8	40.5	68.6	49.5
DANN w/ **SFD (our)**	43.6	63.1	72.3	47.6	58.4	60.1	45.7	39.6	69.2	62.2	49.1	76.0	57.2
CDAN[Long *et al.*, 2017]	39.5	57.8	66.6	42.9	57.6	58.3	42.1	35.0	65.7	54.3	43.5	73.9	53.1
CDAN w/ **SFD (our)**	**44.3**	65.4	**73.2**	50.1	**62.3**	62.8	47.3	**41.6**	72.1	63.4	**50.8**	**78.1**	59.3
MCC[Jin *et al.*, 2020]	40.6	62.5	67.0	47.6	58.7	61.4	43.9	34.6	66.0	57.6	42.0	73.2	54.6
MCC w/ **SFD (our)**	43.7	**67.8**	71.3	**53.4**	62.0	**67.5**	**51.9**	38.3	**72.6**	**66.5**	48.5	77.3	**60.1**
Metric	Average accuracy (%) of all steps												
DANN[Ganin *et al.*, 2016]	38.0	57.1	66.1	43.1	53.3	56.1	41.4	32.2	64.3	56.7	42.0	71.6	51.8
DANN w/ **SFD (our)**	42.1	62.7	70.9	46.7	56.8	59.4	44.5	36.4	67.8	61.9	47.4	75.4	56.0
CDAN[Long *et al.*, 2017]	39.5	58.4	67.8	43.9	56.9	57.6	43.4	34.2	66.1	57.6	43.3	73.0	53.5
CDAN w/ **SFD (our)**	**42.8**	62.7	**71.5**	47.6	**60.5**	61.5	46.1	**37.5**	69.3	62.1	**47.9**	76.1	57.1
MCC[Jin *et al.*, 2020]	40.3	63.0	67.3	47.7	60.2	60.3	46.2	34.3	66.2	58.1	43.3	73.2	55.0
MCC w/ **SFD (our)**	42.5	**66.0**	70.8	**52.1**	61.5	**65.4**	**50.5**	36.5	**70.7**	**64.4**	46.9	**76.2**	**58.6**

Table 1: Accuracy (%) on OUDA tasks with Office-Home dataset



(a) Source-DANN (b) Source-CDAN (c) Source-MCC (d) Target-DANN (e) Target-CDAN (f) Target-MCC

Figure 4: Forgetting analysis on A-W task (X-axis indicates the steps and Y-axis indicates the L2 discrepancy distance). Larger Y means stronger forgetting. Figures (a,b,c) show the forgetting comparison in the source domain between the current generator and the source-only generator. Figures (d,e,f) show the forgetting comparison in the target domain between the current generator and the lastest previous generator.

5 Experiment

In this section, we conduct experiments on a wide range of real-world datasets. We modify three offline UDA algorithms, i.e., DANN [Ganin *et al.*, 2016], CDAN [Long *et al.*, 2017], and MCC [Jin *et al.*, 2020], and evaluate their performance on OUDA tasks. To verify the effectiveness of the proposed SFD, we apply SFD to all baselines. The source code is available in https://github.com/FujitsuResearch/source-feature-distillation.

5.1 Datasets

As the most prevalent benchmark dataset in DA tasks, we conduct experiments on Office-Home [Venkateswara *et al.*, 2017], Office-31 [Saenko *et al.*, 2010] and ImageCLEF-DA [Long *et al.*, 2017].

Office-31 [Saenko *et al.*, 2010] is one of the most prevalent benchmark datasets in DA which has 31 classes with 4,652 images. Office-31 includes three domains, i.e., Amazon (A), Dslr (D), and Webcam (W).

Office-Home [Venkateswara *et al.*, 2017] is a harder task compared with Office-31 which has 65 classes with over 15,500 images including four domains, i.e., Art (A), Clipart (C), Product (P), and Real World (R).

ImageCLEF-DA [Long *et al.*, 2017] is a dataset organized by selecting 12 common classes shared by three public datasets, i.e., Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P).

5.2 Setup

Settings and Metrics. We start OUDA tasks with a source-only (SO) model which trains from scratch with ResNet-50 [He *et al.*, 2016] implemented by Pytorch. The whole target domain is divided into a sequential of sub-domains with a batch size of 36 by randomly selecting. We follow the settings in the literature for online training [Kirkpatrick *et al.*, 2017; McMahan *et al.*, 2013] where the algorithms have no access to the target data arrived in previous steps. The training epoch for each step is set as 20. We adopt mini-batch SGD with a momentum of 0.9 and the learning rate annealing strategy as [Ganin *et al.*, 2016]. For each step, we use the accuracy of the whole target dataset to verify the generalization of the updated model. We both compare the last step accuracy while the average accuracy of overall steps. All tasks are repeated three times and the mean results are used for comparison.

Baselines. We verify the effectiveness of SFD with both domain-adversarial training methods [Ganin *et al.*, 2016; Long *et al.*, 2017] and non-domain-adversarial training method, MCC [Jin *et al.*, 2020]. All methods are based on domain-invariant representations which learn a single feature generator to achieve good performance in both source and target domains. To achieve SFD, we use an extra source-only model to calculate $\mathcal{L}_{sfd}$ for all baselines. For all tasks, we use the same hyper-parameter where $\alpha = 1$, Parameter sensitivity analysis is also applied in Sec. 5.3.

Method	Office-31							ImageCLEF-DA						
	A-W	D-W	W-D	A-D	D-A	W-A	Avg	I-P	P-I	I-C	C-I	C-P	P-C	Avg
SO [He *et al.*, 2016]	66.1	91.3	97.8	72.7	60.3	60.4	74.8	74.3	84.2	91.8	72.3	57.8	88.3	78.1
Metric	Accuracy (%) of last step													
DANN[Ganin *et al.*, 2016]	66.5	93.9	99.1	73.2	55.9	56.2	74.1	71.5	79.3	88.6	73.0	61.3	82.1	76.0
DANN w/ **SFD (our)**	76.3	96.4	**100**	76.9	61.5	62.4	78.9	75.5	85.8	93.1	78.7	**68.8**	88.6	81.8
CDAN[Long *et al.*, 2017]	71.1	94.3	99.5	74.2	61.2	60.2	76.8	71.8	81.1	91.0	70.1	62.0	85.1	76.9
CDAN w/ **SFD (our)**	79.1	97.1	**100**	77.5	65.2	**67.3**	81.0	**76.0**	85.5	93.6	77.5	65.5	90.8	81.5
MCC[Jin *et al.*, 2020]	76.6	96.3	99.3	81.5	63.2	63.4	80.1	71.6	82.8	92.0	85.5	62.3	86.7	80.2
MCC w/ **SFD (our)**	**84.2**	**97.7**	**100**	**85.9**	**67.1**	66.5	**83.6**	75.2	**88.0**	**95.7**	**89.0**	67.8	**93.7**	**84.9**
Metric	Average accuracy (%) of all steps													
DANN[Ganin *et al.*, 2016]	70.2	93.1	99.4	74.7	59.9	58.7	76.0	72.1	79.4	90.0	74.3	62.9	79.4	76.4
DANN w/ **SFD (our)**	73.9	95.5	**99.6**	75.9	60.8	60.6	77.7	74.6	83.5	91.6	76.4	64.0	83.0	78.9
CDAN[Long *et al.*, 2017]	71.5	94.1	98.9	75.1	61.9	59.2	76.8	73.3	80.4	90.3	73.2	62.8	80.4	76.7
CDAN w/ **SFD (our)**	74.7	95.7	**99.6**	76.3	64.7	63.4	79.1	**75.4**	85.1	92.3	76.7	63.9	83.7	79.5
MCC[Jin *et al.*, 2020]	78.3	96.6	98.7	80.2	63.4	62.6	80.0	72.1	84.2	93.4	84.7	64.0	84.2	80.4
MCC w/ **SFD (our)**	**82.2**	**96.9**	99.2	**81.8**	**65.5**	**65.7**	**81.9**	74.5	**87.1**	**94.6**	**87.3**	**65.8**	**87.1**	**82.7**

Table 2: Accuracy (%) on OUDA tasks with Office-31 and ImageCLEF-DA datasets

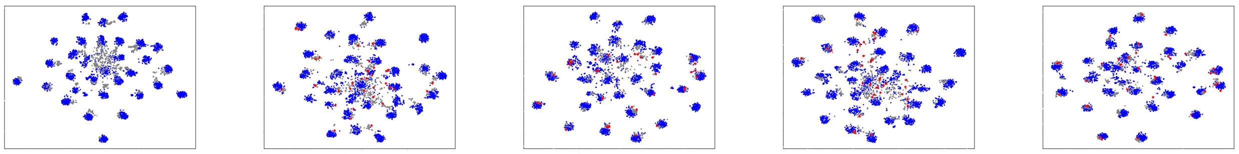

(a) Step 0: SO (b) Step 10: w/o SFD (c) Step 10: w/ SFD (d) Step 21: w/o SFD (e) Step 21: w/ SFD

Figure 5: T-SNE visualization of features on Office-31 A-W task with DANN. Figures (a-e) represent domain alignment in OUDA: blue denotes the source features, red denotes the target features arrived in current step, and gray denotes the other target features.

5.3 Results

Overall Results. According to the data listed in Table 1 and Table 2, all baselines in OUDA suffer from catastrophic forgetting in OUDA tasks which leads to poor performance in most of the tasks. Although all baselines are effective in offline UDA training, the performance in OUDA is even worse than the source-only model in most of the tasks. Data listed in Table 1 and Table 2 shows that SFD can improve the performance of all baselines by 5.5% on the accuracy of the last step while 2.7 % on the average accuracy of all steps. These results have verified the effectiveness of the proposed SFD. However, there still exists a gap between offline training and online training. For example, in the Office-Home task, CDAN [Long *et al.*, 2017] combined with SFD can achieve an accuracy of 59.3% in the last step while according to the results listed in [Long *et al.*, 2017], CDAN can achieve an accuracy of 63.8% in offline training. This is caused by the tradeoffs inherent in learning and remembering representations in OUDA tasks.

Forgetting Analysis. Based on a common input distribution, we use the feature discrepancy generated by two generators to describe forgetting, for example, the forgetting between g^1 and g^2 over a distribution $\mathcal{P}$ can be described as a L2 discrepancy distance, i.e., $\mathbb{E}_{x\sim\mathcal{P}}[\left\|g^1(x)-g^2(x)\right\|_2^2]$. Figure 4 (a,b,c) show the results of forgetting analysis in the source domain between the current generator and the source-only generator, i.e., $\mathcal{L}_{sfd}$. The proposed SFD method has a good performance on the convergence of $\mathcal{L}_{sfd}$ in all baselines. As shown in figure 4 (d,e,f), all baselines suffer from catastrophic forgetting in the target domain, i.e., a large target feature discrepancy even in adjacent steps while the proposed SFD can significantly reduce such forgetting by about 70%.

T-SNE Visualization. Figure 5 shows the T-SNE[Maaten and Hinton, 2008] visualization with DANN [Ganin *et al.*, 2016] on task Office-31 A-W. Although DANN has aligned the red points (available target features) and blue points (source features) in a single step, the gray points (unavailable or unseen target features) suffer from catastrophic forgetting which leads to a similar feature distribution with the source-only model on the whole target domain. The proposed SFD can significantly reduce such forgetting where the whole target can be gradually aligned to the source domain.

Parameter Sensitivity Analysis. The parameter α in Eq. 5 is to control the weights of $\mathcal{L}_{sfd}$. A larger α means a harder constraint on stable source features, i.e., a smaller $\mathcal{L}_{sfd}$. According to Sec. 4.3, a smaller $\mathcal{L}_{sfd}$ is beneficial to a smaller $\sum_i^t g_{\mathcal{H}\triangle\mathcal{H}}(g^t,g^i)$. However, a smaller $\sum_i^t g_{\mathcal{H}\triangle\mathcal{H}}(g^t,g^i)$ also means a weaker ability to obtain new knowledge. When $\sum_i^t g_{\mathcal{H}\triangle\mathcal{H}}(g^t,g^i)=0$, the algorithm has no ability to obtain any new knowledge. As the **Theorem 2** is bounded by both $\sum_i^t g_{\mathcal{H}\triangle\mathcal{H}}(g^t,g^i)$ and $d_{\mathcal{H}\triangle\mathcal{H}}\left(\mathcal{P}_S(g^t(x)),\mathcal{P}_T^t(g^t(x))\right)$), there exists a trade-off between the two items. Data listed in Table 3 shows the results of parameter sensitivity analysis

Method	**Office-31 A-W**					
	0	0.1	0.5	1	5	10
Metric	Accuracy (%) of last step					
DANN w/ **SFD**	66.5	75.7	76.1	76.3	74.7	73.9
CDAN w/ **SFD**	71.1	77.8	79.8	79.1	76.6	75.8
MCC w/ **SFD**	76.6	81.6	85.1	84.2	87.4	83.1
Metric	Average accuracy (%) of all steps					
DANN w/ **SFD**	70.2	73.2	73.6	73.9	72.8	72.3
CDAN w/ **SFD**	71.5	74.3	74.4	74.7	73.8	72.9
MCC w/ **SFD**	78.3	80.6	81.8	82.2	83.1	81.3

Table 3: Parameter sensitivity analysis for α

where α is at a wide range form 0.1 to 10. The results show that the proposed SFD has stable performance at a wide range of α while we can also observe the trade-offs on minimizing $\sum_i^t g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i)$. When $\alpha = 10$, the performance is worse than other settings except $\alpha = 0$, the removal of the SFD.

6 Conclusion

Existing unsupervised domain adaptation (UDA) studies focus on transferring knowledge in an offline manner. However, many tasks involve online requirements, especially in real-time systems. In this paper, we discuss online UDA (OUDA) which suffers from catastrophic forgetting with the drawback of poor generalization. We propose a novel generalization bound for OUDA which illustrates the tradeoffs inherent in learning and remembering representations. To minimize the proposed bound, we provide a novel SFD for OUDA which utilizes the source-only model as a teacher to guide OUDA algorithms. The experimental results show that SFD significantly improves the generalization of all baselines in online iterative procedures.

A Appendix

In this section, we show the proofs.

A.1 Proof of Lemma 1

Let $\mathcal{H}$ be a hypothesis space in a input space $\mathcal{X}$ with any distribution $\mathcal{D}$. We denote the loss function $\mathcal{L} : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_+$ and assume that $\mathcal{L}$ satisfies the triangle inequality. Then for any $h, h^{'}, h^{''} \in \mathcal{H}$, we have

$$\epsilon_{\mathcal{D}}(h, h^{'}) \leq \epsilon_{\mathcal{D}}(h, h^{''}) + \epsilon_{\mathcal{D}}(h^{''}, h^{'}) \tag{18}$$

Proof.

$$\begin{aligned}\epsilon_{\mathcal{D}}(h, h^{'}) =& \int_{\mathcal{X}} \mathcal{L}(h(x), h^{'}(x)) d\mathcal{P}_{\mathcal{D}}(x) \\ \leq& \int_{\mathcal{X}} \mathcal{L}(h(x), h^{''}(x)) + \mathcal{L}(h^{'}(x), h^{''}(x)) d\mathcal{P}_{\mathcal{D}}(x) \\ \leq& \int_{\mathcal{X}} \mathcal{L}(h(x), h^{''}(x)) d\mathcal{P}_{\mathcal{D}}(x) \\ &+ \int_{\mathcal{X}} \mathcal{L}(h^{'}(x), h^{''}(x)) d\mathcal{P}_{\mathcal{D}}(x) \\ =& \epsilon_{\mathcal{D}}(h, h^{''}) + \epsilon_{\mathcal{D}}(h^{''}, h^{'})\end{aligned} \tag{19}$$

Then we finish the proof. □

A.2 Proof of Theorem 2

Let $\mathcal{H}$ be a hypothesis space. Let $\mathcal{D}_T^t$ be the t-th target domain with a marginal data distribution of $\mathcal{P}_T^t(x)$. Then $\forall g^t, h^t$,

$$\begin{aligned}\epsilon_T^{t-all} \leq& t\epsilon_S(h^t \circ g^t) + \sum_{i=1}^{t} \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^i(x))\right) \\ &+ \sum_{i=1}^{t} g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i) + \sum_{i=1}^{t} \lambda_t^*(i)\end{aligned} \tag{20}$$

where $\lambda_t^*(i) = \min_{h^* \in \mathcal{H}} \epsilon_T^i(h^* \circ g^t) + \epsilon_S(h^* \circ g^t)$

Proof. $\forall i \leq t$, let f_S and f_T be the label functions in two domains. According to **Lemma 1**,

$$\begin{aligned}\epsilon_T^i(h^t \circ g^t) \leq& \epsilon_T^i(h^t \circ g^t, h^* \circ g^t) + \epsilon_T^i(h^* \circ g^t, f_T \circ g^t) \\ \leq& \epsilon_T^i(h^t \circ g^t, h^* \circ g^t) - \epsilon_S(h^t \circ g^t, h^* \circ g^t) \\ &+ \epsilon_S(h^t \circ g^t, h^* \circ g^t) + \epsilon_T^i(h^* \circ g^t, f_T \circ g^t) \\ \leq& \epsilon_S(h^t \circ g^t, f_S \circ g^t) \\ &+ \left|\epsilon_T^i(h^t \circ g^t, h^* \circ g^t) - \epsilon_S(h^t \circ g^t, h^* \circ g^t)\right| \\ &+ \epsilon_S(h^* \circ g^t, f_S \circ g^t) + \epsilon_T^i(h^* \circ g^t, f_T \circ g^t) \\ \leq& \epsilon_S(h^t \circ g^t) + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^t(x)), \mathcal{P}_T^i(g^t(x))\right) \\ &+ \min_{h^* \in \mathcal{H}} \epsilon_T^i(h^* \circ g^t) + \epsilon_S(h^* \circ g^t)\end{aligned} \tag{21}$$

For the second item, $\Delta = \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^t(x)), \mathcal{P}_T^i(g^t(x))\right)$,

$$\begin{aligned}2\Delta \leq& d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^t(x)), \mathcal{P}_S(g^i(x)))\right) \\ &+ d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^t(x)))\right) \\ \leq& g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i) + d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^i(x)))\right) \\ &+ d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_T^i(g^t(x)), \mathcal{P}_T^i(g^i(x)))\right) \\ \leq& 2g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i) + d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^i(x)))\right)\end{aligned} \tag{22}$$

Let $\lambda_t^*(i) = \min_{h^* \in \mathcal{H}} \epsilon_T^i(h^* \circ g^t) + \epsilon_S(h^* \circ g^t)$. Then,

$$\begin{aligned}\epsilon_T^i(h^t \circ g^t) \leq& \epsilon_S(h^t \circ g^t) + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}_S(g^i(x)), \mathcal{P}_T^i(g^i(x))\right) \\ &+ g_{\mathcal{H}\triangle\mathcal{H}}(g^t, g^i) + \lambda_t^*(i)\end{aligned} \tag{23}$$

Apply Eq. 23 to ϵ_T^{t-all}, we finish the proof. □

A.3 Proof of Lemma 3

Let $\mathcal{X}$ (resp. $\mathcal{H}$) be a input (resp. hypothesises) space. Then $\forall g^1, g^2, g^3$, we have

$$g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^2\right) \leq g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^3\right) + g_{\mathcal{H}\triangle\mathcal{H}}\left(g^3, g^2\right) \tag{24}$$

Proof. According to **Lemma 2**, we have

$$\begin{aligned}g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^2\right) =& \sup_{\mathcal{P} \in \mathcal{X}} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}(g^1(x)), \mathcal{P}(g^2(x))\right) \\ \leq& \sup_{\mathcal{P} \in \mathcal{X}} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}(g^1(x)), \mathcal{P}(g^3(x))\right) \\ &+ \sup_{\mathcal{P} \in \mathcal{X}} d_{\mathcal{H}\triangle\mathcal{H}} \left(\mathcal{P}(g^3(x)), \mathcal{P}(g^2(x))\right) \\ =& g_{\mathcal{H}\triangle\mathcal{H}}\left(g^1, g^3\right) + g_{\mathcal{H}\triangle\mathcal{H}}\left(g^3, g^2\right)\end{aligned} \tag{25}$$

Then we finish the proof. □

References

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[Blitzer *et al.*, 2007] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. *Advances in neural information processing systems*, 20, 2007.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[Hajifar and Sun, 2021] Sahand Hajifar and Hongyue Sun. Online domain adaptation for continuous cross-subject liver viability evaluation based on irregular thermal data. *IISE Transactions*, pages 1–12, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Jin *et al.*, 2020] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[Long *et al.*, 2017] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Mancini *et al.*, 2018] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the wild through online domain adaptation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1103–1109. IEEE, 2018.

[Mao *et al.*, 2017] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[McMahan *et al.*, 2013] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230, 2013.

[Moon *et al.*, 2020] JH Moon, Debasmit Das, and CS George Lee. Multi-step online unsupervised domain adaptation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 41172–41576. IEEE, 2020.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[Rolnick *et al.*, 2019] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Saenko *et al.*, 2010] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[Shu *et al.*, 2018] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[Tachet des Combes *et al.*, 2020] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33, 2020.

[Tzeng *et al.*, 2017] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[Venkateswara *et al.*, 2017] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

[Zhang *et al.*, 2019] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

[Zhao *et al.*, 2019] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.