Open-Set Heterogeneous Domain Adaptation: Theoretical Analysis and Algorithm

Thai-Hoang Pham[1,2], Yuanlong Wang[1,2], Changchang Yin[1,2], Xueru Zhang[1], Ping Zhang[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Department of Biomedical Informatics, The Ohio State University, USA
{pham.375,wang.16050,yin.731,zhang.12807,zhang.10631}@osu.edu

Abstract

Domain adaptation (DA) tackles the issue of distribution shift by learning a model from a source domain that generalizes to a target domain. However, most existing DA methods are designed for scenarios where the source and target domain data lie within the same feature space, which limits their applicability in real-world situations. Recently, heterogeneous DA (HeDA) methods have been introduced to address the challenges posed by heterogeneous feature space between source and target domains. Despite their successes, current HeDA techniques fall short when there is a mismatch in both feature and label spaces. To address this, this paper explores a new DA scenario called open-set HeDA (OSHeDA). In OSHeDA, the model must not only handle heterogeneity in feature space but also identify samples belonging to novel classes. To tackle this challenge, we first develop a novel theoretical framework that constructs learning bounds for prediction error on target domain. Guided by this framework, we propose a new DA method called Representation Learning for OSHeDA (RL-OSHeDA). This method is designed to simultaneously transfer knowledge between heterogeneous data sources and identify novel classes. Experiments across text, image, and clinical data demonstrate the effectiveness of our algorithm. Model implementation is available at https://github.com/pth1993/OSHeDA.

1 Introduction

Machine learning (ML) techniques have achieved unprecedented success over the past decades in numerous areas (LeCun, Bengio, and Hinton 2015). However, ML systems are often built on the assumption that training and testing data are independent and identically distributed, which is commonly violated in real-world applications where the environment changes during model deployment. Existing works have shown that the performance of ML models often deteriorates due to distribution shifts between training and testing data (Ben-David et al. 2010; Quiñonero-Candela et al. 2022). To learn a model robust under distribution shifts, domain adaptation (DA) (Ben-David et al. 2010; Mansour, Mohri, and Rostamizadeh 2009) has been proposed to transfer knowledge from a source domain that possesses abundant labeled data to a different but relevant target domain.

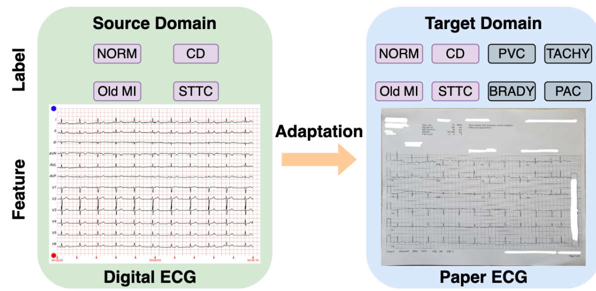


Figure 1: A motivating example about OSHeDA in the context of screening diseases using electrocardiogram (ECG) data. While digital ECGs comprise the majority of labeled data for training ML models for disease screening, physical or paper ECGs remain prevalent worldwide. Thus, the transfer of knowledge from digital ECG datasets is essential to support the training of ML models that analyze paper ECGs. Moreover, ML systems must effectively manage rare abnormalities (indicated with gray boxes), which may not be available in training data, to prevent misdiagnosis.

Existing DA methods, however, typically assume a homogeneous scenario where the source and target domains have the same feature and label spaces. Consequently, they may fail when the source and target domain data lie in different spaces. For example, heterogeneous feature space is common in biomedical domains in which medical terms undergoes continuous evolution, leading to the retirement of outdated terms (e.g., ICD-9 coding system) and the introduction of novel ones (e.g., ICD-10 coding system) (Grief et al. 2016). In such cases, acquiring training data that seamlessly aligns with target domain's feature space can be impractical or excessively costly. Heterogeneous domain adaptation (HeDA) methods have emerged to handle the heterogeneity observed in distinct feature spaces, which often vary significantly between domains (Li et al. 2020; Zhao et al. 2022).

Despite the significant successes achieved by these HeDA methods, they face a major limitation: current HeDA techniques can only address heterogeneity in feature space and are inadequate when there is a mismatch in both feature and label spaces. This limitation restricts the practical application of HeDA methods in many real-world scenarios because neglecting label mismatch, such as new classes emerging in the target domain, can lead to negative transfer effects from the source to the target domains (Liu et al. 2019).

To overcome this limitation, this study explores a new DA scenario called open-set heterogeneous domain adaptation (OSHeDA). In OSHeDA, ML methods must not only manage heterogeneity in feature space between source and target domains but also identify samples belonging to novel classes in the target domain. Figure 1 illustrates a real example from clinical applications for this novel learning scenario. In this instance, the adaptation process aims to transfer knowledge from digital electrocardiogram (ECG) to paper ECG formats (*heterogeneous*). Moreover, ML models for ECG-based diagnosis must also detect rare abnormalities that were not included in the training data (*open-set*).

To address the challenge of feature and label mismatch in OSHeDA, we first develop a novel theoretical analysis that constructs learning bounds for the prediction error of ML models on the target domain. Guided by this theoretical analysis, we then design a novel representation learning method named **R**epresentation **L**earning for **O**pen-**S**et **He**terogeneous **D**omain **A**daptation (RL-OSHeDA). This method is proposed to transfer knowledge between heterogeneous data sources and identify novel class simultaneously. Unlike existing HeDA methods, RL-OSHeDA transfer knowledge from source to target domains by aligning representations between source and target domains for known classes while also enforcing the representations of novel class in target domains to move apart from the known classes of the source and target domains. To effectively identify samples from novel class within unlabeled data, RL-OSHeDA optimizes a non-negative risk estimator for open-set and employs pseudo labeling to enrich the labeled data.

In summary, the contributions of our work are as follows:

- We conduct a theoretical analysis to establish learning bounds in the OSHeDA scenario. This analysis emphasizes the importance of minimizing the distance between source and target domains for known classes, while maximizing the separation from unknown classes. Moreover, we investigate the impact of pseudo-label and the non-negative risk estimator for open-set in OSHeDA.
- Motivated by the theoretical results, we propose a novel algorithm (RL-OSHeDA) based on representation learning to transfer knowledge from source to target domains.
- We conduct experiments on real data from clinical, computer vision, and natural language processing domains to validate the effectiveness of our method for OSHeDA.

2 Related Works

In this section, we summarize existing research from related areas including heterogeneous domain adaptation, open-set domain adaptation, and open-set semi-supervised learning.

Heterogeneous Domain Adaptation (HeDA). HeDA aims to transfer knowledge across domains with distinct feature spaces and data distributions. Depending on whether unlabeled target data are used in the adaptation process, HeDA approaches are categorized into three types: supervised, semi-supervised, and unsupervised methods. Supervised HeDA methods utilize ample labeled data from both source and target domains for adaptation (Hoffman et al. 2013; Li et al. 2013; Hoffman et al. 2014). In contrast, semi-supervised HeDA methods require only a small number of labeled target domain data and utilize unlabeled instances from the target domain to facilitate transfer (Yao et al. 2020; Li et al. 2020; Fang et al. 2022; Zhao et al. 2022; Yao et al. 2019). Finally, unsupervised HeDA methods operate without any labeled target data, relying solely on unlabeled instances and labeled source data to align cross-domain feature representations (Shen and Guo 2018; Li et al. 2018; Zou et al. 2018). However, successful transfer in unsupervised settings depends on specific assumptions about domain relationships (Liu, Zhang, and Lu 2020).

Open-Set Domain Adaptation (OSDA). OSDA represents a realistic and challenging scenario in DA where the target domain includes instances whose classes are not observed in the source domain, alongside a shift in feature distribution between the two domains. In contrast to the OSHeDA, OSDA assumes a homogeneous feature space between the source and target domains. Existing approaches for OSDA can be categorized into two main groups: adversarial learning and self-supervised learning. Adversarial learning methods employ adversarial networks to detect unknown samples and align the distributions of known samples between the source and target domains (Saito et al. 2018; Luo et al. 2020). On the other hand, self-supervised learning methods utilize techniques like data augmentation to distinguish between known and unknown instances in the target domain (Bucci, Loghmani, and Tommasi 2020; Li et al. 2021).

Open-Set Semi-supervised Learning (OS-SSL). OS-SSL is a SSL scenario that addresses novel classes within unlabeled data during training. Unlike OSDA, OS-SSL requires only a small amount of labeled data. However, it assumes that both labeled and unlabeled data of known classes are drawn from the same distribution, and this setting does not account for novel classes during inference. Methods designed for OS-SSL can be broadly categorized into two types based on how they detect novel classes: criterion-based approaches and detector-based approaches. Criterion-based approaches use heuristic rules to identify novel classes (Chen et al. 2020; Huang, Yang, and Gong 2022; Du et al. 2023; He et al. 2022). In contrast, detector-based approaches employ parameterized detectors to filter outliers (Yu et al. 2020; Huang et al. 2021; Wang et al. 2023; Saito, Kim, and Saenko 2021).

3 Problem Formulation

Notations. Let $\mathcal{X}^d$ and $\mathcal{Y}^d$ denote the feature and label spaces of a domain d associated with a distribution $P_d(X_d, Y_d) : \mathcal{X}^d \times \mathcal{Y}^d \rightarrow [0,1]$ and labeling function $h_d : \mathcal{X}^d \rightarrow \Delta\left(\mathcal{Y}^d\right)$ where X_d and Y_d are random variables that take values in $\mathcal{X}^d$ and $\mathcal{Y}^d$, and $\Delta\left(\mathcal{Y}^d\right)$ is a probability simplex over $\mathcal{Y}^d$. Consider a model $h : \mathcal{X}^d \rightarrow \Delta\left(\mathcal{Y}^d\right)$, then the *expected error* of h under domain d for some loss function $L : \Delta\left(\mathcal{Y}^d\right) \times \mathcal{Y}^d \rightarrow \mathbb{R}_+$ (e.g., 0-1, cross-entropy loss) can be defined as $\mathrm{E}\left(P_d, h\right) = \mathbb{E}_{P_d}\left[L\left(h\left(X_d\right), Y_d\right)\right].$

Open-Set Heterogeneous Domain Adaptation (OSHeDA) Setup. In DA, we consider $d \in \{s, t\}$ where s and t denote the source and target domains, respectively. Different from conventional DA setup where feature and label spaces remain the same between source and target domains, in OSHeDA, we have $\mathcal{X}^s \neq \mathcal{X}^t$ (*heterogeneous*) and $\mathcal{Y}^s \subset \mathcal{Y}^t$ (*open-set*). Because $\mathcal{Y}^s \subset \mathcal{Y}^t$, we use Y to denote the random variable of label in both source and target domains, and we have $P_s\left(Y \in \mathcal{Y}^t \setminus \mathcal{Y}^s\right) = 0$. Moreover, classes in the sets $\mathcal{Y}^t \setminus \mathcal{Y}^s$ are referred to as unknown in our setting. Given sets of samples $D_s = \{x_i^s, y_i^s\}_{i=1}^{n_s} \overset{i.i.d}{\sim} P_s\left(X_s, Y\right)$ (*source dataset*), $D_{t_l} = \{x_i^t, y_i^t\}_{i=1}^{n_{t_l}} \overset{i.i.d}{\sim} P_t\left(X_t, Y | Y \in \mathcal{Y}^s\right)$ (*labeled target dataset*), and $D_{t_u} = \{x_i^t\}_{i=1}^{n_{t_u}} \overset{i.i.d}{\sim} P_t\left(X_t, Y\right)$ (*unlabeled target dataset*), where n_s, n_{t_l}, n_{t_u} are size of datasets and $n_{t_l} \ll n_s, n_{t_u}$, the goal of OSHeDA is to learn a model $h : \mathcal{X}^t \to \Delta\left(\mathcal{Y}^t\right)$ from D_s, D_{t_l}, D_{t_u} such that the expected error on the target domain $\mathrm{E}\left(P_t, h\right)$ is small.

Representation learning. Representation learning is a common approach for transferring knowledge from a source to a target domain in DA (Zhao et al. 2019; Ganin et al. 2016; Albuquerque et al. 2019; Pham, Zhang, and Zhang 2023), and we will leverage this method in OSHeDA. Specifically, it maps the input spaces $\mathcal{X}^s$ and $\mathcal{X}^t$ of the source and target domains to a shared representation space $\mathcal{Z}$ using two representation mappings: $f_s : \mathcal{X}^s \to \mathcal{Z}$ and $f_t : \mathcal{X}^t \to \mathcal{Z}$. A shared classifier $h : \mathcal{Z} \to \Delta(\mathcal{Y}^t)$ can then be employed to make predictions from this representation space. Notably, h can be utilized for both domains because $\mathcal{Y}^s \subset \mathcal{Y}^t$.

4 Theoretical Analysis

In our analysis, we consider Jensen–Shannon (JS) divergence ($\mathcal{D}_{JS}$) as the statistical distance between two domains. While different distances (Ben-David et al. 2010) were used in domain adaptation literature, we adopt JS divergence because it is aligned with the training objective of adversarial learning (Goodfellow et al. 2014), a technique used in many representation learning-based domain adaptation works (Zhao et al. 2019; Ganin et al. 2016; Pham, Zhang, and Zhang 2023). Next, we present the main theorems, with detailed proofs provided in Appendix A.

4.1 Learning bounds for OSHeDA (infinite case)

To simplify notations used in our following analysis, we denote $P_{t,k}(\cdot) = P_t(\cdot | Y \in \mathcal{Y}^s)$ and $P_{t,u}(\cdot) = P_t(\cdot | Y \notin \mathcal{Y}^s)$ as the distributions of target domain conditioned on known and unknown classes, respectively. We also introduce two distributions P_s^u and P_t^u induced from P_s and P_t by the two mappings f_s^u and f_t^u such that $f_s^u(X^s, Y) = (X^s, unk)$ and $f_t^u(X^t, Y) = (X^t, unk)$ where unk denotes unknown class. In addition, we adopt an assumption commonly used in DA literature (Nguyen et al. 2021; Mansour, Mohri, and Rostamizadeh 2009; Cortes and Mohri 2014) as follows.

Assumption 1 (Bounded loss) *Assume loss function L defined on input space $\mathcal{X}$ and output space $\mathcal{Y}$ is upper bounded by a constant C, i.e., $\forall x \in \mathcal{X}, y \in \mathcal{Y}$, $h \in \mathcal{H}$, we have $L(h(x), y) \leq C$.*

We note that this assumption is indeed reasonable rather than stringent. For example, while Assumption 1 does not hold for the cross-entropy loss typically utilized in classification, we can adjust this loss to ensure that it satisfies Assumption 1 (Pham, Zhang, and Zhang 2024). Based on this assumption, we then provide an upper bound for prediction error on the target domain in OSHeDA as follows.

Theorem 1 *Given a loss function L satisfying Assumption 1, then for any $h \in \mathcal{H}, f_s \in \mathcal{F}_s, f_t \in \mathcal{F}_t$, we have:*

$$\begin{aligned}
\mathrm{E}\left(P_t, h \circ f_t\right) \leq\ & \underbrace{\lambda\, \mathrm{E}\left(P_s, h \circ f_s\right)}_{\textbf{\textit{source error}}} \\
& + \underbrace{\mathrm{E}\left(P_t^u, h \circ f_t\right) - \lambda\, \mathrm{E}\left(P_s^u, h \circ f_s\right)}_{\textbf{\textit{open-set difference}}} \\
& \underbrace{\begin{aligned} &+ \sqrt{2}\lambda C \Big(\left(\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}} \\ &+ \left(\mathcal{D}_{JS}\left(P_s(Z, Y) \parallel P_{t,k}(Z, Y)\right)\right)^{\frac{1}{2}} \Big) \end{aligned}}_{\textbf{\textit{domain distance}}}
\end{aligned}$$

where $\lambda = P_t(Y \in \mathcal{Y}^s)$, $\mathcal{H}$, $\mathcal{F}_s$, $\mathcal{F}_t$ are hypothesis classes for h, f_s, f_t, and $P_s(Z)$ and $P_{t,k}(Z)$ are the distributions induced from $P_s(X_s)$ and $P_{t,k}(X_t)$ by f_s and f_t, respectively.

Remark 1 *The upper bound in Theorem 1 shed a light on achieving good accuracy on target domain. Specifically, to minimize $\mathrm{E}\left(P_t, h \circ f_t\right)$, the model need to optimize three terms: **(i)** the source error $\mathrm{E}\left(P_s, h \circ f_s\right)$, **(ii)** the open-set difference $\mathrm{E}\left(P_t^u, h \circ f_t\right) - \lambda\, \mathrm{E}\left(P_s^u, h \circ f_s\right)$, and **(iii)** the distances of marginal and joint distributions between source domain and target domain conditioned on known labels $\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,k}(Z)\right)$ and $\mathcal{D}_{JS}\left(P_s(Z, Y) \parallel P_{t,k}(Z, Y)\right)$.*

We want to emphasize that minimizing the distance of the joint distribution between the source and target domains, $\mathcal{D}_{JS}\left(P_s(Z, Y) \parallel P_{t,k}(Z, Y)\right)$, requires knowledge of the label distribution in the target domain $P_{t,k}(Y)$. Therefore, access to labeled target data during training is essential to avoid negative transfer. Note that the concept of open-set difference is not exclusive to OSHeDA. This term also appears in existing works for OSDA (Fang et al. 2020) and positive-unlabeled learning (Kiryo et al. 2017) which are special cases of our setting. Thus, this demonstrates the consistency between our work and the existing literature. Next, we present a lower bound for OSHeDA.

Proposition 1 *Given a loss function L satisfying Assumption 1, then for any $h \in \mathcal{H}, f_s \in \mathcal{F}_s, f_t \in \mathcal{F}_t$, we have:*

$$\begin{aligned}
\mathrm{E}\left(P_t, h \circ f_t\right) \geq\ & \lambda\, \mathrm{E}\left(P_{t,k}, h \circ f_t\right) + (1-\lambda)\, \mathrm{E}\left(P_s^u, h \circ f_s\right) \\
& - \sqrt{2}(1-\lambda) C \left(\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,u}(Z)\right)\right)^{\frac{1}{2}}
\end{aligned}$$

where $P_{t,u}(Z)$ is distribution induced from $P_{t,u}(X_t)$ by f_t.

Remark 2 *Theorem 1 shows the necessity of reducing $\mathrm{E}\left(P_s, h \circ f_s\right)$ to achieve high accuracy on target domain. However, it may unavoidably increase $\mathrm{E}\left(P_s^u, h \circ f_s\right)$. This observation, combined with Proposition 1, suggests that to avoid the large lower bound for the target error $\mathrm{E}\left(P_t, h \circ f_t\right)$, we should increase the distance of the marginal distribution between the source domain and the unknown data in target domain, $\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,u}(Z)\right)$. In other words, we should segregate the representations of known classes from those of unknown class.*

4.2 Learning bound for OSHeDA (finite case)

The learning bounds discussed in Section 4.1 are only applicable for the setting when we have access to unlimited data from source and target domains. In such cases, minimizing JS divergence of data distribution between these domains is equivalent to achieving invariant representations through adversarial learning (Goodfellow et al. 2014). However, we only work with finite data in practice. Thus, we present the following result, which provides a guarantee for using adversarial learning to optimize JS divergence from finite data.

Proposition 2 (Adapted from Biau et al. (2020)) *The error in minimizing JS divergence of data distributions between source and target domains in representation space, using finite data, is up to* $\mathcal{O}\left(\left(1/\sqrt{n_s}+1/\sqrt{n_t}\right)\right)$.

where n_s and n_t are the size of source and target datasets.

Remark 3 *Proposition 2 states that the performance of minimizing JS divergence from finite data is proportional to the dataset size. Note that in OSHeDA, we only have access to limited label data from target domain which then results in significant error in estimating JS divergence only from labeled source and target data. In essence, this result underscores the need for the development of an effective approach to utilize unlabeled target data for estimating the JS divergence, which involves techniques like pseudo-labeling.*

Therefore, we apply pseudo-labeling on unlabeled data to enrich labeled target data. Let g be pseudo-label model and denote $\mathrm{N}(P_{t,k},g)=\mathbb{E}\left[\mathcal{D}_{JS}\left(P_{t,k}(g(Z))\parallel P_{t,k}(Y|Z)\right)\right]$ as the noise of g with respect to the target domain conditioned on known labels. Then, the impact of pseudo-labeled data can be illustrated in a new bound for OSHeDA as follows.

Theorem 2 *Given a loss function L satisfying Assumption 1, for any $0<\delta<1$, with probability at least $1-\delta$, the following holds for all $h\in\mathcal{H}, f_s\in\mathcal{F}_s, f_t\in\mathcal{F}_t$:*

$$\begin{aligned}&\mathrm{E}\left(P_t,h\circ f_t\right)\le\lambda\widehat{\mathrm{E}}\left(P_s,h\circ f_s\right)+\widehat{\mathrm{E}}\left(P^u_t,h\circ f_t\right)\\&-\lambda\widehat{\mathrm{E}}\left(P^u_s,h\circ f_s\right)+\sqrt{2}\lambda C\left(\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}}\right.\\&\left.+\left(\mathcal{D}_{JS}\left(P_s(Z,Y)\parallel P_{t,k}(Z,g(Z))\right)\right)^{\frac{1}{2}}+\left(\mathrm{N}(P_{t,k},g)\right)^{\frac{1}{2}}\right)\\&+\mathcal{O}\left(\lambda C\sqrt{\frac{d_s\log n_s+d_s\log|\mathcal{Y}^t|+\log\frac{1}{\delta}}{n_s}}\right.\\&\left.+C\sqrt{\frac{d_t\log n_t+d_t\log|\mathcal{Y}^t|+\log\frac{1}{\delta}}{n_t}}\right)\end{aligned}$$

where $\widehat{\mathrm{E}}\left(P_s,h\circ f_s\right)$, $\widehat{\mathrm{E}}\left(P^u_t,h\circ f_t\right)$, $\widehat{\mathrm{E}}\left(P^u_s,h\circ f_s\right)$ are empirical errors calculated on samples from distributions P_s, P^u_t, P^u_s, $n_t=n_{t_l}+n_{t_u}$, and d_s, d_t are Natarajan dimension (Natarajan 1989) of hypothesis classes $\mathcal{H}\circ\mathcal{F}_s$, $\mathcal{H}\circ\mathcal{F}_t$.

Theorem 2 shows that the error in the target domain depends on the quality of the pseudo-label model g, with higher-quality g being more effective at reducing noise. Additionally, the bound emphasizes the importance of aligning the joint distributions between the source and target domains in OSHeDA. This makes OSHeDA more challenging compared to homogeneous DA (HoDA), where source and target data lie on the same space. In contrast, HoDA methods can attain good performance under certain conditions by solely aligning the marginal distributions of representations between source and target domains. We will illustrate this contrast through the bound for HoDA in Section 4.3.

4.3 Learning bound for HoDA

Before constructing the learning bound for HoDA, we introduce an assumption about the representation Z as follows.

Assumption 2 (Sufficient representation) *Let $I_s(\cdot,\cdot)$ be the mutual information between two random variables in the source domain. We assume $I_s(Z,Y)=I_s(X_s,Y)$. In particular, $I_s(Z,Y)=\mathcal{D}_{KL}\left(P_s(Z,Y)\parallel P_s(Z)\otimes P_s(Y)\right)$ and $I_s(X_s,Y)=\mathcal{D}_{KL}\left(P_s(X_s,Y)\parallel P_s(X_s)\otimes P_s(Y)\right)$ where D_{KL} is KL divergence between two distributions.*

Note that Assumption 2 is reasonable because we have access to labeled data of source domain and the dimension of $\mathcal{Y}$ is often smaller than that of $\mathcal{Z}$. Based on this, we establish the learning bound in HoDA under the covariate shift below.

Proposition 3 *Suppose Assumptions 1 and 2 hold and the distribution shift between source and target domains is covariate shift (i.e., $P_s(X)\neq P_t(X), P_s(Y|X)=P_t(Y|X)$), then for any $h\in\mathcal{H}$ and $f\in\mathcal{F}$, we have:*

$$\mathrm{E}\left(P_t,h\circ f\right)\le\mathrm{E}\left(P_s,h\circ f\right)+\sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_t(Z)\right)\right)^{\frac{1}{2}}$$

In HoDA, due to the homogeneity of the input space, we can utilize a single representation mapping f for both the source and target domains. Note that the bound in Proposition 3 depends solely on the distance of the marginal distributions between the source and target domains, $\mathcal{D}_{JS}\left(P_s(Z)\parallel P_t(Z)\right)$, which can be effectively minimized even without access to labeled data in the target domain. Clearly, covariate shift assumption is only reasonable in HoDA, where the source and target data share the same feature and label spaces.

5 Methodology

Motivated by theoretical results presented in Section 4, we introduce RL-OSHeDA, a representation learning method specifically designed for OSHeDA. Our method aims to simultaneously optimize both the upper bound in Theorem 2 and the lower bound in Proposition 1. RL-OSHeDA features two distinct representation mappings, f_s and f_t, which map heterogeneous source and target feature spaces to a shared representation space, along with a classifier h that makes predictions based on these representations. Figure 2 presents the overall architecture of RL-OSHeDA, while pseudo code describing training process can be found in Appendix B.2.

5.1 Objective function

To improve predictive performance in OSHeDA, our method targets the following: (i) minimizing prediction errors on both source and labeled target data, (ii) minimizing the distances of marginal and label-conditioned representation distributions for known classes between source and target

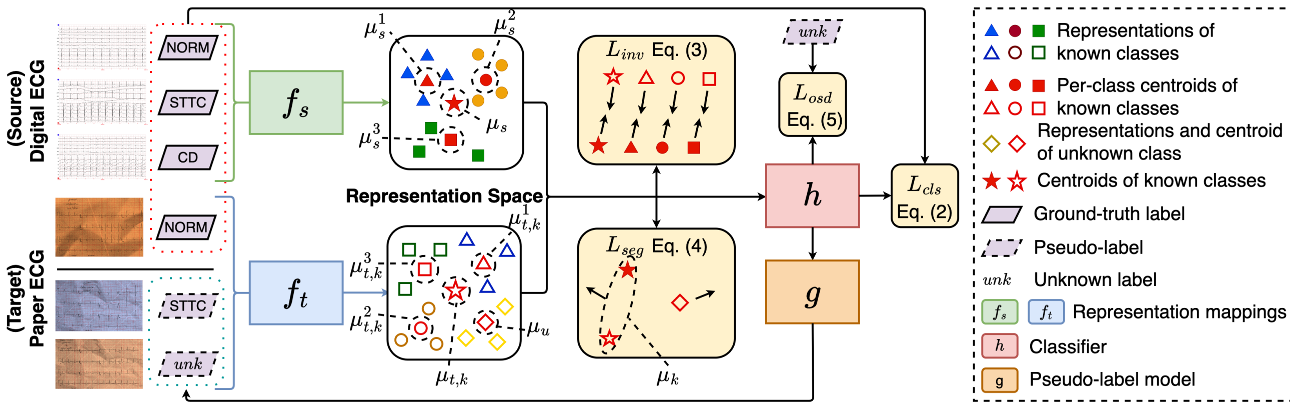


Figure 2: Overall architecture of RL-OSHeDA is illustrated with a motivating example from ECG-based diagnosis application. We leverage 2-stage learning process to update model parameters. In stage 1, model parameters are updated by optimizing L_{cls}. In stage 2, model parameters are updated by optimizing L_{cls}, L_{inv}, L_{seg}, and L_{osd} with the help from pseudo-label model g.

data, (iii) maximizing the distances of marginal representation distributions between known and unknown classes, and (iv) minimizing the open-set difference. Specifically, RL-OSHeDA optimizes the following objective function:

$$L = L_{cls} + L_{inv} - L_{seg} + L_{osd} \tag{1}$$

where L_{cls} is the classification error computed from source and labeled target datasets D_s and D_{t_l}, defined as follows:

$$L_{cls} = \frac{\lambda}{n_s}\sum_{i=1}^{n_s} \text{CE}\left(h\left(f_s\left(x_i^s\right)\right), y_i^s\right) + \frac{1}{n_{t_l}}\sum_{i=1}^{n_{t_l}} \text{CE}\left(h\left(f_t\left(x_i^t\right)\right), y_i^t\right) \tag{2}$$

Here CE is the cross-entropy loss.

L_{inv} denotes the distances of marginal and label-conditioned representation distributions for known classes between source and target datasets. Note that, we minimize the distance of the label-conditioned representation distribution $P(Z|Y)$, rather than the joint distribution $P(Z,Y)$, as noted by Pham, Zhang, and Zhang (2023). As shown in Proposition 2, L_{inv} can be defined based on JS divergence and minimized through adversarial learning. However, the number of discriminators required for this approach scales linearly with the number of classes, leading to instability in training when the dataset has a large number of classes. To address this issue, we implement L_{inv} using maximum mean discrepancy (MMD) defined as follows:

$$L_{inv} = \|\mu_s - \mu_{t,k}\|_2^2 + \sum_{m=1}^{|\mathcal{Y}^s|} \left\|\mu_s^m - \mu_{t,k}^m\right\|_2^2 \tag{3}$$

where μ_s (resp. $\mu_{t,k}$) is centroid of representations from source data (resp. target data belonging to known classes), and μ_s^m (resp. $\mu_{t,k}^m$) is centroid of representations from source data (resp. target data) belonging to known class m. Note that $\mu_{t,k}$ and $\mu_{t,k}^m$ are computed using both instances with ground-truth labels from labeled target data and those with high-quality pseudo-labels (see Section 5.2) from unlabeled target data to provide a more accurate estimation.

L_{seg} is the distances between marginal representation distributions of known and unknown classes. Similarly, we implement L_{seg} with MMD as follows:

$$L_{seg} = \|\mu_k - \mu_u\|_2^2 \tag{4}$$

where μ_k (resp. μ_u) are centroids of representations from both source and target datasets belonging to ground-truth and pseudo known (resp. unknown) classes.

L_{osd} represents the open-set difference, as detailed in Theorem 2. The optimal value for the open-set difference is 0. However, due to the flexibility of deep neural networks, this term can become excessively negative during training and adversely affect model performance. To address this issue, we implement L_{osd} as a non-negative risk estimator:

$$L_{osd} = \max\left(0, \frac{1}{n_t}\sum_{i=1}^{n_t} \text{CE}\left(h\left(f_t\left(x_i^t\right)\right), unk\right) - \frac{\lambda}{n_s}\sum_{i=1}^{n_s} \text{CE}\left(h\left(f_s\left(x_i^s\right)\right), unk\right)\right) \tag{5}$$

where $n_t = n_{t_l} + n_{t_u}$ is the size of the target dataset.

5.2 Pseudo-labeling using 2-stage learning

The accuracy of L_{inv} and L_{seg} highly depends on the quality of pseudo-labels. Traditionally, the pseudo-label model g is derived by modifying the classifier h (e.g., using hard labels calculated from h's outputs as pseudo-labels), which creates a coupling between g and h. Specifically, g is defined as $a \circ h$, where a is an operator applied to the output of h (e.g., $a := \arg\max$). When the distributions of the source and target domains are well-aligned, this coupling is harmless, as the optimal solution for g also aligns with that for h. However, at the beginning of the training process, when the distributions of the source and target domains are not aligned, g and h have completely different objective functions, resulting in a trade-off between them. To address this issue, we propose a 2-stage learning approach as follows:

- **Stage 1** ($epoch < T$): Update f_s, f_t, h using L_{cls}.
- **Stage 2** ($epoch \geq T$): Update f_s, f_t, h using L.

where T is a threshold indicating when to switch from stage 1 to stage 2. In stage 1, optimizing L_{cls} partially aligns the source and target domains, thereby reducing the trade-off between g and h during the optimization of L in stage 2. Additionally, rather than simply using the hard labels with

Table 1: Prediction performances (HOS, OS^*, UNK) of RL-OSHeDA and baselines for OSHeDA scenario on 7 datasets. We report average results over 10 random seeds for each dataset.

	CIFAR10 & ILSVRC2012			ImageCLEF-DA			Multilingual Reuters Collection			NUSWIDE & ImageNet		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
DS3L	61.49±0.74	59.04±1.00	64.40±1.06	58.62±2.04	52.87±2.70	66.74±2.77	59.35±0.94	52.92±1.27	67.57±1.24	67.61±1.65	66.17±2.22	69.20±2.30
KPG	57.27±0.50	54.73±0.00	60.30±1.11	40.68±0.94	34.60±0.00	50.79±2.91	11.27±0.07	8.59±0.00	17.04±0.96	55.18±1.17	52.60±0.00	58.10±2.45
OPDA	53.30±0.77	48.22±1.00	60.26±1.11	53.17±1.83	45.28±2.13	65.76±2.76	55.85±0.96	48.47±1.23	65.94±1.23	71.06±1.44	66.60±1.98	76.38±2.09
PL	42.75±0.52	37.12±0.49	52.31±1.10	39.20±1.62	31.93±1.66	54.34±2.91	42.85±0.81	34.56±1.00	57.86±1.29	42.43±0.26	34.05±0.00	61.15±2.10
SCT	59.61±0.75	57.35±1.00	62.33±1.08	58.76±2.05	53.09±2.71	66.71±2.76	61.17±0.94	**54.96±1.30**	69.00±1.21	70.42±1.49	68.00±2.20	73.10±1.99
SSAN	60.38±0.73	59.01±1.00	62.01±1.08	58.61±2.05	53.18±2.74	66.14±2.74	58.25±0.93	51.99±1.25	66.26±1.24	67.98±1.49	66.25±2.04	69.85±2.21
STN	61.59±0.72	58.80±0.98	64.87±1.05	56.25±2.06	49.80±2.69	65.84±2.76	59.21±0.96	52.91±1.31	67.24±1.23	67.75±1.23	64.80±1.42	71.08±2.16
SL	60.74±0.74	58.29±1.00	63.67±1.08	58.59±2.05	52.84±2.70	66.71±2.76	58.53±0.96	52.14±1.32	66.74±1.21	69.41±1.64	66.63±2.26	72.57±2.23
RL-OSHeDA	**72.33±0.70**	**67.88±0.98**	**77.81±0.97**	**63.98±2.04**	**56.63±2.72**	**74.80±2.51**	**65.39±0.91**	54.47±1.21	**81.97±0.96**	**80.01±1.30**	**74.65±2.01**	**86.35±0.81**

	Office & Caltech256			Wikipedia			PTB-XL			Average over datasets		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
DS3L	72.06±2.48	67.41±3.68	78.15±2.89	56.00±2.01	50.72±2.36	66.24±2.80	30.30±1.19	34.95±0.58	26.74±1.82	57.92±1.58	54.87±1.97	62.72±2.12
KPG	34.46±0.99	29.18±0.00	45.34±3.58	24.82±0.42	16.40±0.00	52.52±3.18	N/A	N/A	N/A	37.28±0.68	32.68±0.00	47.35±2.36
OPDA	65.23±2.58	57.70±3.43	76.23±3.03	52.66±1.92	45.94±1.81	65.42±3.12	31.47±1.22	36.35±0.63	27.74±1.86	54.67±1.53	49.79±1.74	62.53±2.17
PL	48.92±1.36	40.38±1.13	68.41±3.40	41.87±1.65	35.14±1.48	58.40±3.10	26.18±1.37	36.43±0.55	20.43±1.66	40.60±1.08	35.66±0.90	53.27±2.22
SCT	75.72±2.14	71.05±3.20	81.79±2.54	58.41±2.01	52.86±2.39	68.42±2.74	26.23±1.65	**46.48±1.71**	18.27±1.60	59.89±1.58	57.10±1.93	64.49±1.99
SSAN	72.95±2.36	67.99±3.37	79.67±2.88	58.37±1.76	52.76±1.95	68.36±2.80	25.16±1.47	40.40±0.65	18.27±1.54	57.39±1.54	55.94±1.86	61.51±2.07
STN	72.26±2.28	66.46±3.30	79.84±2.75	57.75±1.91	51.40±2.13	69.00±2.97	27.08±0.96	22.63±0.26	33.72±1.65	57.41±1.45	52.40±1.73	64.51±2.08
SL	72.14±2.54	67.72±3.75	77.89±2.96	57.10±1.97	51.60±2.19	67.04±2.76	25.74±1.55	44.50±0.76	18.11±1.52	57.46±1.64	56.24±2.00	61.82±2.08
RL-OSHeDA	**78.18±2.05**	**73.04±2.91**	**85.25±2.50**	**63.10±1.89**	**57.26±2.45**	**73.04±2.37**	**47.48±1.25**	44.30±1.39	**51.16±1.86**	**67.21±1.45**	**61.18±1.95**	**75.77±1.71**

the largest logits from h as the output of g, we propose generating pseudo-labels as follows:

- First, select pseudo-labels as $g(x^t) = a'(h(f_t(x^t)))$ where a' is $\arg\max$ operator applied to the logits of the known classes only.
- Then, select $1 - \lambda$ fraction of instances with the smallest maximum logits and assign pseudo-labels unk to them.

The motivation behind this design of the pseudo-label model g is that, at the beginning of stage 2, there is no supervision signal for training the parameters of h related to unknown class. Therefore, relying solely on logits to determine the unknown class is unreliable. Note that this strategy is only used to generate pseudo-labels during the training of stage 2. Once training is complete and h's parameters for unknown class are well-trained by optimizing L_{osd}, we can simply use $\arg\max$ across all classes to generate predictions.

6 Experiments

Next, we empirically evaluate the performance of our methods across clinical, computer vision, and natural language processing applications. We focus on the OsHeDA scenario, characterized by heterogeneity in the feature space between the source and target domains, with the label space of the target domain encompassing both known and unknown classes.

6.1 Experimental setup

Datasets. We conduct our experiments on 7 datasets including CIFAR10 (Krizhevsky 2009) & ILSVRC2012 (Russakovsky et al. 2015); Wikipedia (Rasiwasia et al. 2010); Multilingual Reuters Collection (Amini, Usunier, and Goutte 2009); NUSWIDE (Chua et al. 2009) & ImageNet (Deng et al. 2009); Office (Saenko et al. 2010) & Caltech256 (Griffin et al. 2007); ImageCLEF-DA (Griffin et al. 2007); PTB-XL (Wagner et al. 2020). These datasets results in 56 DA tasks. Detailed descriptions and statistics of these datasets are provided in Appendix C.1.

Baselines. We compare our method with several representative methods from **HeDA** (SSAN (Li et al. 2020), STN (Yao et al. 2019), SCT (Zhao et al. 2022), KPG (Gu et al. 2022)), **OSDA** (OPDA (Saito et al. 2018)), and **OS-SSL** (DS3L (Guo et al. 2020)) literature. For the HeDA methods, they are trained on both source and target data. In contrast, OSDA and OS-SSL methods are trained only on target data as they cannot handle heterogeneous feature spaces. During inference, HeDA and OS-SSL methods classify instances as unk using the same method as our pseudo-label model g (see Section 5.2). Additionally, we explore supervised learning (SL) and pseudo-labeling (PL) methods trained on target data. Among all baselines, only KPG is designed to handle OSHeDA by combining Gromov-Wasserstein distance and partial optimal transport (Xu et al. 2020). Since λ is a required input for most methods in our experiments, we utilize techniques from positive-unlabeled learning (Zeiberg, Jain, and Radivojac 2020) to estimate λ. Detailed architectures of our model and the baselines are in Appendix B.1.

Evaluation method. We utilize HOS, the harmonic mean of OS^* and UNK (Bucci, Loghmani, and Tommasi 2020). OS^* is the class-wise averaged accuracy of known classes, while UNK measures the accuracy for the unknown class. HOS is particularly suitable for OSHeDA because it emphasizes the ability to both correctly classify known classes and detect out-of-distribution instances simultaneously. In particular, this metric increases when the performance in both known and unknown classifications is high.

6.2 Experimental results

OSHeDA benchmark. The prediction performance (HOS) of RL-OSHeDA and the baselines is summarized in Table 1. RL-OSHeDA consistently outperforms all baselines across all datasets, demonstrating its effectiveness in simultaneously addressing heterogeneity in the feature space and open-set in the label space during training. Among the baselines, KPG is specifically designed for OSHeDA by using optimal transport. Then, SVM trained on transported source and labeled target data is used to make prediction. However, this method underperforms in our evaluation due to its difficulty in correctly transporting from source to target data. Moreover, this method is not applicable for complex data structures, such as those found in PTB-XL dataset. Other

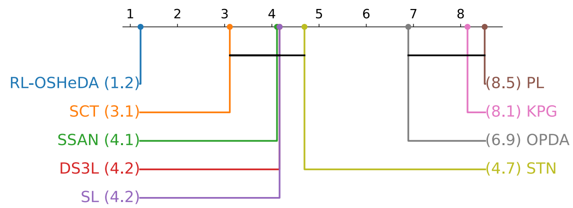


Figure 3: Critical Difference diagram for all methods calculated from 56 DA tasks. RL-OSHeDA is the highest ranked method on HOS metric, and its performance is significantly better than baselines (as indicated by the lack of connections between RL-OSHeDA and baselines in the diagram).

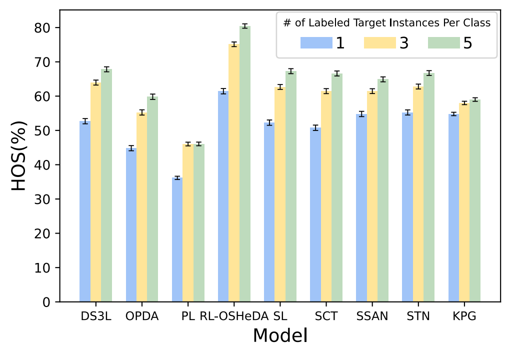


Figure 4: Performances w.r.t. different number of labeled target instances per class on CIFAR10 & ILSVRC2012 dataset.

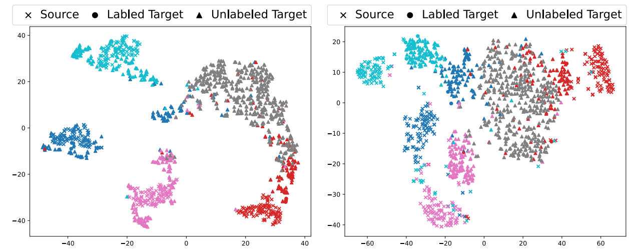
baselines achieve better prediction performances, but their HOS remains suboptimal due to their inability to handle novel classes or heterogeneous source data during training.

To further validate the superiority of RL-OSHeDA across all DA tasks, we conduct significance testing, including the Friedman test followed by the Nemenyi test (Demšar 2006). The results (see Figure 3) show that our method significantly outperforms the baselines, with a P-value smaller than 0.05. Among all the baselines, SCT, SSAN, STN, SL, and DS3L exhibit better prediction performances than KPG, OPDA, and PL. Note that all methods, except OPDA and KPG, utilize our approach to detect the unknown class based on logits of known classes. This result suggests that while this approach can partially address the open-set issue, it cannot fully resolve it. For OPDA, although it is designed to handle open-set issue, its inability to leverage heterogeneous source data limits its performance to adapting with only a small labeled target dataset, resulting in suboptimal performance.

Ablation study. We conduct an ablation study to better understand the contribution of each component in the objective function of our method. As shown in Table 2, removing any component deteriorates model performance. This finding highlights the importance of achieving a good pseudo-label model using 2-stage learning approach as well as aligning the data distribution of known classes between source and target domains while simultaneously detecting and segregating unknown class from known ones for OSHeDA.

Impact of labeled target data. We vary the number of instances per class in the labeled target data to investigate their impact on the DA process. Specifically, we conduct experiments on CIFAR10 & ILSVRC2012 dataset with 1, 3, and 5 instances per class in the labeled target data and visualize the result in Figure 4. Generally, we observe that increasing the number of labeled target instances facilitates better alignment and enhances the performance of all methods. This result demonstrates the importance of labeled target data for DA methods in OSHeDA.

Visualization of representation space. We perform a qualitative analysis to examine the learned representations of RL-OSHeDA and STN for NUSWIDE & ImageNet dataset. Specifically, we use t-SNE (Van der Maaten and Hinton 2008) to project these representations into a 2-dimensional space. As shown in Figure 5, our method effectively aligns representations of the known classes between source and target domains while simultaneously segregating the representations of the unknown class (grey color). This results in improved HOS scores compared to STN.



(a) RL-OSHeDA (b) STN

Figure 5: Visualization of representation spaces learned by RL-OSHeDA and STN for NUSWIDE & ImageNet dataset. Different colors represent different classes, with the unknown class denoted in grey.

Table 2: Ablation study for RL-OSHeDA on Multilingual Reuters Collection dataset. **Align** refers to using L_{inv}; **Segregate** refers to using L_{seg}; **OSD** refers to using L_{osd}; **2-stage** refers to using 2-stage learning approach.

Align	Segregate	OSD	2-stage	HOS	OS^*	UNK
✓	✓	✓	✓	**65.39**	**54.47**	**81.97**
✓	✓	✓	✗	59.40	49.47	74.42
✓	✗	✓	✓	61.92	52.63	75.37
✗	✓	✓	✓	58.23	51.13	68.01
✓	✓	✗	✓	59.96	53.10	68.97
✗	✗	✗	✗	58.33	51.86	66.68

7 Conclusion

This paper studied a novel domain adaptation scenario called open-set heterogeneous domain adaptation (OSHeDA). We first conducted a theoretical analysis to establish learning bounds in OSHeDA. Based on these theorems, we proposed a representation learning method that aligns the data distribution of known classes between source and target domains while simultaneously detecting and segregating unknown class from known ones. The resulting models trained with the proposed method generalize well to target domains. Experiments on real datasets across diverse domains, including healthcare, natural language processing, and computer vision, demonstrate the effectiveness of our proposed method.

Acknowledgements

This work was funded in part by the National Science Foundation under award number IIS-2145625 and by the National Institutes of Health under award number R01AI188576. We would like to express our sincere gratitude to Dr. Wei-Lun Chao from The Ohio State University for his valuable suggestions, which greatly improved this paper.

References

Albuquerque, I.; Monteiro, J.; Darvishi, M.; Falk, T. H.; and Mitliagkas, I. 2019. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*.

Amini, M. R.; Usunier, N.; and Goutte, C. 2009. Learning from multiple partially observed views-an application to multilingual text categorization. *Advances in neural information processing systems*, 22.

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

Biau, G.; Cadre, B.; Sangnier, M.; and Tanielian, U. 2020. Some theoretical properties of GANS. *The Annals of Statistics*, 48(3): 1539 – 1566.

Bucci, S.; Loghmani, M. R.; and Tommasi, T. 2020. On the effectiveness of image rotation for open set domain adaptation. In *European conference on computer vision*, 422–438. Springer.

Chen, Y.; Zhu, X.; Li, W.; and Gong, S. 2020. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3569–3576.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, 1–9.

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

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7: 1–30.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, 647–655. PMLR.

Du, P.; Zhao, S.; Sheng, Z.; Li, C.; and Chen, H. 2023. Semi-Supervised Learning via Weight-aware Distillation under Class Distribution Mismatch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16410–16420.

Fang, Z.; Lu, J.; Liu, F.; Xuan, J.; and Zhang, G. 2020. Open set domain adaptation: Theoretical bound and algorithm. *IEEE transactions on neural networks and learning systems*, 32(10): 4309–4322.

Fang, Z.; Lu, J.; Liu, F.; and Zhang, G. 2022. Semi-supervised heterogeneous domain adaptation: Theory and algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 1087–1105.

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

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grief, S. N.; Patel, J.; Kochendorfer, K. M.; Green, L. A.; Lussier, Y. A.; Li, J.; Burton, M.; and Boyd, A. D. 2016. Simulation of ICD-9 to ICD-10-CM transition for family medicine: simple or convoluted? *The Journal of the American Board of Family Medicine*, 29(1): 29–36.

Griffin, G.; Holub, A.; Perona, P.; et al. 2007. Caltech-256 object category dataset. Technical report, Technical Report 7694, California Institute of Technology Pasadena.

Gu, X.; Yang, Y.; Zeng, W.; Sun, J.; and Xu, Z. 2022. Keypoint-guided optimal transport with applications in heterogeneous domain adaptation. *Advances in Neural Information Processing Systems*, 35: 14972–14985.

Guo, L.-Z.; Zhang, Z.-Y.; Jiang, Y.; Li, Y.-F.; and Zhou, Z.-H. 2020. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International conference on machine learning*, 3897–3906. PMLR.

He, R.; Han, Z.; Lu, X.; and Yin, Y. 2022. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14585–14594.

Hoffman, J.; Rodner, E.; Donahue, J.; Darrell, T.; and Saenko, K. 2013. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*.

Hoffman, J.; Rodner, E.; Donahue, J.; Kulis, B.; and Saenko, K. 2014. Asymmetric and category invariant feature transformations for domain adaptation. *International journal of computer vision*, 109: 28–41.

Huang, J.; Fang, C.; Chen, W.; Chai, Z.; Wei, X.; Wei, P.; Lin, L.; and Li, G. 2021. Trash to treasure: Harvesting ood data with cross-modal matching for open-set semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8310–8319.

Huang, Z.; Yang, J.; and Gong, C. 2022. They are not completely useless: Towards recycling transferable unlabeled data for class-mismatched semi-supervised learning. *IEEE Transactions on Multimedia*, 25: 1844–1857.

Kiryo, R.; Niu, G.; Du Plessis, M. C.; and Sugiyama, M. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30.

Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; and Houlsby, N. 2019. Big transfer (BiT): General visual representation learning. arXiv 2020. *arXiv preprint arXiv:1912.11370*.

Koltchinskii, V.; and Panchenko, D. 2000. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, 443–457. Springer.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Tront*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

Li, G.; Kang, G.; Zhu, Y.; Wei, Y.; and Yang, Y. 2021. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9757–9766.

Li, J.; Lu, K.; Huang, Z.; Zhu, L.; and Shen, H. T. 2018. Heterogeneous domain adaptation through progressive alignment. *IEEE transactions on neural networks and learning systems*, 30(5): 1381–1391.

Li, S.; Xie, B.; Wu, J.; Zhao, Y.; Liu, C. H.; and Ding, Z. 2020. Simultaneous semantic alignment network for heterogeneous domain adaptation. In *Proceedings of the 28th ACM international conference on multimedia*, 3866–3874.

Li, W.; Duan, L.; Xu, D.; and Tsang, I. W. 2013. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern analysis and machine intelligence*, 36(6): 1134–1148.

Liang, P. 2016. CS229T/STAT231: Statistical learning theory (Winter 2016).

Liu, F.; Zhang, G.; and Lu, J. 2020. Heterogeneous domain adaptation: An unsupervised approach. *IEEE transactions on neural networks and learning systems*, 31(12): 5588–5602.

Liu, H.; Cao, Z.; Long, M.; Wang, J.; and Yang, Q. 2019. Separate to adapt: Open set domain adaptation via progressive separation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2927–2936.

Luo, Y.; Wang, Z.; Huang, Z.; and Baktashmotlagh, M. 2020. Progressive graph learning for open-set domain adaptation. In *International Conference on Machine Learning*, 6468–6478. PMLR.

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

Natarajan, B. K. 1989. On learning sets and functions. *Machine Learning*, 4: 67–97.

Nguyen, A. T.; Tran, T.; Gal, Y.; Torr, P.; and Baydin, A. G. 2021. KL Guided Domain Adaptation. In *International Conference on Learning Representations*.

Pham, T.-H.; Zhang, X.; and Zhang, P. 2023. Fairness and Accuracy under Domain Generalization. In *The Eleventh International Conference on Learning Representations*.

Pham, T.-H.; Zhang, X.; and Zhang, P. 2024. Non-stationary Domain Generalization: Theory and Algorithm. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Quiñonero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2022. *Dataset shift in machine learning*. Mit Press.

Rasiwasia, N.; Costa Pereira, J.; Coviello, E.; Doyle, G.; Lanckriet, G. R.; Levy, R.; and Vasconcelos, N. 2010. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, 251–260.

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

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, 213–226. Springer.

Saito, K.; Kim, D.; and Saenko, K. 2021. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. *Advances in Neural Information Processing Systems*, 34: 25956–25967.

Saito, K.; Yamamoto, S.; Ushiku, Y.; and Harada, T. 2018. Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision (ECCV)*, 153–168.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shen, C.; and Guo, Y. 2018. Unsupervised heterogeneous domain adaptation with sparse feature transformation. In *Asian conference on machine learning*, 375–390. PMLR.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wagner, P.; Strodthoff, N.; Bousseljot, R.-D.; Kreiseler, D.; Lunze, F. I.; Samek, W.; and Schaeffter, T. 2020. PTB-XL, a large publicly available electrocardiography dataset. *Scientific data*, 7(1): 1–15.

Wang, Y.; Qiao, P.; Liu, C.; Song, G.; Zheng, X.; and Chen, J. 2023. Out-of-distributed semantic pruning for robust semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23849–23858.

Xu, R.; Liu, P.; Zhang, Y.; Cai, F.; Wang, J.; Liang, S.; Ying, H.; and Yin, J. 2020. Joint Partial Optimal Transport for Open Set Domain Adaptation. In *IJCAI*, 2540–2546.

Yao, Y.; Zhang, Y.; Li, X.; and Ye, Y. 2019. Heterogeneous domain adaptation via soft transfer network. In *Proceedings of the 27th ACM MM*, 1578–1586.

Yao, Y.; Zhang, Y.; Li, X.; and Ye, Y. 2020. Discriminative distribution alignment: A unified framework for heterogeneous domain adaptation. *Pattern Recognition*, 101: 107165.

Yu, Q.; Ikami, D.; Irie, G.; and Aizawa, K. 2020. Multi-task curriculum framework for open-set semi-supervised learning. In *Computer Vision–ECCV 2020: 16th European Con-*

ference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16, 438–454. Springer.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33: 17283–17297.

Zeiberg, D.; Jain, S.; and Radivojac, P. 2020. Fast nonparametric estimation of class proportions in the positive-unlabeled classification setting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6729–6736.

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

Zhao, Y.; Li, S.; Zhang, R.; Liu, C. H.; Cao, W.; Wang, X.; and Tian, S. 2022. Semantic correlation transfer for heterogeneous domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*.

Zou, Y.; Yu, Z.; Kumar, B.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, 289–305.

A Proofs

A.1 Additional Lemmas

Lemma 1 *Given two domains s and t associated with two distributions $P_s(X,Y)$ and $P_t(X,Y)$, respectively, then for any classifier $h:\mathcal{X}\rightarrow\Delta(\mathcal{Y})$, the expected error of h in domain t can be upper bounded:*

$$\mathrm{E}\left(P_t,h\right)\leq\mathrm{E}\left(P_s,h\right)+\sqrt{2}C\times\mathcal{D}_{JS}\left(P_t(X,Y)\parallel P_s(X,Y)\right)^{1/2}$$

where $\mathcal{D}_{JS}\left(\cdot\parallel\cdot\right)$ is JS-divergence between two distributions.

Proof of Lemma 1 Let $\mathcal{D}_{KL}\left(\cdot\parallel\cdot\right)$ be KL-divergence, p_s, p_t are probability density functions associated with P_s, P_t, and $U=(X,Y)$ and $L(U)=L\left(h(X),Y\right)$. We first prove $\int_{\mathcal{E}}|p_t(u)-p_s(u)|\,du=\frac{1}{2}\int|p_t(u)-p_s(u)|\,du$ where $\mathcal{E}$ is the event that $p_t(u)\geq p_s(u)$ $(*)$ as follows:

$$\begin{aligned}\int_{\mathcal{E}}|p_t(u)-p_s(u)|\,du&=\int_{\mathcal{E}}\left(p_t(u)-p_s(u)\right)du\\&=\int_{\mathcal{E}\cup\overline{\mathcal{E}}}\left(p_t(u)-p_s(u)\right)du-\int_{\overline{\mathcal{E}}}\left(p_t(u)-p_s(u)\right)du\\&\overset{(1)}{=}\int_{\overline{\mathcal{E}}}\left(p_s(u)-p_t(u)\right)du\\&=\int_{\overline{\mathcal{E}}}|p_t(u)-p_s(u)|\,du\\&=\frac{1}{2}\int|p_t(u)-p_s(u)|\,du\end{aligned}$$

where $\overline{\mathcal{E}}$ is the complement of $\mathcal{E}$. We have $\overset{(1)}{=}$ because $\int_{\mathcal{E}\cup\overline{\mathcal{E}}}\left(p_t(u)-p_s(u)\right)du=\int_{\mathcal{U}}\left(p_t(u)-p_s(u)\right)du=0$. Then, we have:

$$\begin{aligned}\mathrm{E}\left(P_t,h\right)&=\mathbb{E}_{P_t}\left[L(U)\right]\\&=\int_{\mathcal{U}}L(u)p_t(u)du\\&=\int_{\mathcal{U}}L(u)p_s(u)du+\int_{\mathcal{U}}L(u)\left(p_t(u)-p_s(u)\right)du\\&=\mathbb{E}_{P_s}\left[L(U)\right]+\int_{\mathcal{U}}L(u)\left(p_t(u)-p_s(u)\right)du\\&=\mathrm{E}\left(P_s,h\right)+\int_{\mathcal{E}}L(u)\left(p_t(u)-p_s(u)\right)du+\int_{\overline{\mathcal{E}}}L(u)\left(p_t(u)-p_s(u)\right)du\\&\overset{(2)}{\leq}\mathrm{E}\left(P_s,h\right)+\int_{\mathcal{E}}L(u)\left(p_t(u)-p_s(u)\right)du\\&\overset{(3)}{\leq}\mathrm{E}\left(P_s,h\right)+C\int_{\mathcal{E}}\left(p_t(u)-p_s(u)\right)du\\&=\mathrm{E}\left(P_s,h\right)+C\int_{\mathcal{E}}|p_t(u)-p_s(u)|\,du\\&\overset{(4)}{=}\mathrm{E}\left(P_s,h\right)+\frac{C}{2}\int|p_t(u)-p_s(u)|\,du\\&\overset{(5)}{\leq}\mathrm{E}\left(P_s,h\right)+\frac{C}{2}\sqrt{2\min\left(\mathcal{D}_{KL}\left(P_s(U)\parallel P_t(U)\right),\mathcal{D}_{KL}\left(P_t(U)\parallel P_s(U)\right)\right)}\\&\leq\mathrm{E}\left(P_s,h\right)+\frac{C}{\sqrt{2}}\sqrt{\mathcal{D}_{KL}\left(P_t(U)\parallel P_s(U)\right)}\end{aligned}\tag{6}$$

We have $\overset{(2)}{\leq}$ because $\int_{\overline{\mathcal{E}}}L(u)\left(p_t(u)-p_s(u)\right)du\leq0$; $\overset{(3)}{\leq}$ because $L(u)$ is non-negative function and is bounded by C; $\overset{(4)}{=}$ by using $(*)$; $\overset{(5)}{\leq}$ by using Pinsker's inequality between total variation norm and KL-divergence.

Let $P_{s,t}(U) = \frac{1}{2}(P_t(U) + P_s(U))$. Apply Eq.(6) for two distributions P_t and $P_{s,t}$, we have:

$$\mathrm{E}(P_t, h) \le \mathrm{E}(P_{s,t}, h) + \frac{C}{\sqrt{2}}\sqrt{\mathcal{D}_{KL}(P_t(U) \parallel P_{s,t}(U))} \tag{7}$$

Apply Eq.(6) again for two distributions $P_{s,t}$ and P_s, we have:

$$\mathrm{E}(P_{s,t}, h) \le \mathrm{E}(P_s, h) + \frac{C}{\sqrt{2}}\sqrt{\mathcal{D}_{KL}(P_s(U) \parallel P_{s,t}(U))} \tag{8}$$

Adding Eq. (7) to Eq. (8) and subtracting $\mathrm{E}(P_{s,t}, h)$, we have:

$$\begin{aligned}
\mathrm{E}(P_t, h) &\le \mathrm{E}(P_s, h) + \frac{C}{\sqrt{2}}\left(\sqrt{\mathcal{D}_{KL}(P_t(U) \parallel P_{s,t}(U))} + \sqrt{\mathcal{D}_{KL}(P_s(U) \parallel P_{s,t}(U))}\right) \\
&\overset{(6)}{\le} \mathrm{E}(P_s, h) + \frac{C}{\sqrt{2}}\sqrt{2\left(\mathcal{D}_{KL}(P_t(U) \parallel P_{s,t}(U)) + \mathcal{D}_{KL}(P_s(U) \parallel P_{s,t}(U))\right)} \\
&= \mathrm{E}(P_s, h) + \frac{C}{\sqrt{2}}\sqrt{4\mathcal{D}_{JS}(P_s(U) \parallel P_t(U))} \\
&= \mathrm{E}(P_s, h) + \sqrt{2}C\sqrt{\mathcal{D}_{JS}(P_s(U) \parallel P_t(U))}
\end{aligned}$$

We have $\overset{(6)}{\le}$ by using Cauchy–Schwarz inequality.

Lemma 2 *Given two domains s and t associated with two distributions $P_s(X,Y)$ and $P_t(X,Y)$, respectively, then JS-divergence $\mathcal{D}_{JS}(P_s(X,Y) \parallel P_t(X,Y))$ can be decomposed as follows:*

$$\begin{aligned}
\mathcal{D}_{JS}(P_s(X,Y) \parallel P_t(X,Y)) \le \mathcal{D}_{JS}(P_s(Y) \parallel P_t(Y)) &+ \mathbb{E}_{P_s}\left[\mathcal{D}_{JS}(P_s(X|Y) \parallel P_t(X|Y))\right] \\
&+ \mathbb{E}_{P_t}\left[\mathcal{D}_{JS}(P_s(X|Y) \parallel P_t(X|Y))\right]
\end{aligned}$$

Proof of Lemma 2 First, we show the decomposition formulation for KL-divergence as follow.

$$\begin{aligned}
&\mathcal{D}_{KL}(P_s(X,Y) \parallel P_t(X,Y)) \\
&= \mathbb{E}_{P_s}\left[\log p_s(X,Y) - \log p_t(X,Y)\right] \\
&= \mathbb{E}_{P_s}\left[\log p_s(Y) + \log p_s(X|Y)\right] - \mathbb{E}_{p_s}\left[\log p_t(Y) + \log p_t(X|Y)\right] \\
&= \mathbb{E}_{P_s}\left[\log p_s(Y) - \log p_t(Y)\right] + \mathbb{E}_{P_s}\left[\log p_s(X|Y) - \log p_t(X|Y)\right] \\
&= \mathbb{E}_{P_s}\left[\log p_s(Y) - \log p_t(Y)\right] + \mathbb{E}_{y\sim P_s(Y)}\left[\mathbb{E}_{x\sim P_s(X|Y)}\left[\log p_s(X|Y) - \log p_t(X|Y)\right]\right] \\
&= \mathcal{D}_{KL}(P_s(Y) \parallel P_t(Y)) + \mathbb{E}_{P_s}\left[\mathcal{D}_{KL}(P_s(X|Y) \parallel P_t(X|Y))\right]
\end{aligned} \tag{9}$$

For JS-divergence, we have:

$$\begin{aligned}
&\mathcal{D}_{JS}(P_s(X,Y) \parallel P_t(X,Y)) \\
&= \frac{1}{2}\left(\mathcal{D}_{KL}(P_s(X,Y) \parallel P_{s,t}(X,Y))\right) + \frac{1}{2}\left(\mathcal{D}_{KL}(P_t(X,Y) \parallel P_{s,t}(X,Y))\right) \\
&\overset{(1)}{=} \frac{1}{2}\left(\mathcal{D}_{KL}(P_s(Y) \parallel P_{s,t}(Y))\right) + \frac{1}{2}\left(\mathbb{E}_{P_s}\left[\mathcal{D}_{KL}(P_s(X|Y) \parallel P_{s,t}(X|Y))\right]\right) \\
&+ \frac{1}{2}\left(\mathcal{D}_{KL}(P_t(Y) \parallel P_{s,t}(Y))\right) + \frac{1}{2}\left(\mathbb{E}_{P_t}\left[\mathcal{D}_{KL}(P_t(X|Y) \parallel P_{s,t}(X|Y))\right]\right) \\
&= \mathcal{D}_{JS}(P_s(Y) \parallel P_t(Y)) + \frac{1}{2}\left(\mathbb{E}_{P_s}\left[\mathcal{D}_{KL}(P_s(X|Y) \parallel P_{s,t}(X|Y))\right]\right) + \frac{1}{2}\left(\mathbb{E}_{P_t}\left[\mathcal{D}_{KL}(P_t(X|Y) \parallel P_{s,t}(X|Y))\right]\right) \\
&\le \mathcal{D}_{JS}(P_s(Y) \parallel P_t(Y)) + \frac{1}{2}\left(\mathbb{E}_{P_s}\left[\mathcal{D}_{KL}(P_s(X|Y) \parallel P_{s,t}(X|Y))\right]\right) + \frac{1}{2}\left(\mathbb{E}_{P_s}\left[\mathcal{D}_{KL}(P_t(X|Y) \parallel P_{s,t}(X|Y))\right]\right) \\
&+ \frac{1}{2}\left(\mathbb{E}_{P_t}\left[\mathcal{D}_{KL}(P_t(X|Y) \parallel P_{s,t}(X|Y))\right]\right) + \frac{1}{2}\left(\mathbb{E}_{P_t}\left[\mathcal{D}_{KL}(P_s(X|Y) \parallel P_{s,t}(X|Y))\right]\right) \\
&= \mathcal{D}_{JS}(P_s(Y) \parallel P_t(Y)) + \mathbb{E}_{P_s}\left[\mathcal{D}_{JS}(P_s(X|Y) \parallel P_t(X|Y))\right] + \mathbb{E}_{P_t}\left[\mathcal{D}_{JS}(P_s(X|Y) \parallel P_t(X|Y))\right]
\end{aligned}$$

We have $\overset{(1)}{=}$ by applying Eq. (9) for $\mathcal{D}_{KL}(P_s(X,Y) \parallel P_{s,t}(X,Y))$ and $\mathcal{D}_{KL}(P_t(X,Y) \parallel P_{s,t}(X,Y))$.

Lemma 3 *Given two domains s and t associated with two distributions $P_s(X,Y)$ and $P_t(X,Y)$, respectively, let $f : \mathcal{X} \rightarrow \mathcal{Z}$ be the representation mapping from input space $\mathcal{X}$ to representation space $\mathcal{Z}$. If the shift between domains s and t is covariate shift (i.e., $P_s(Y|X) = P_t(Y|X)$), and Assumption 2 holds for the representation Z, then the shift between these two domains in representation space is also covariate shift (i.e., $P_s(Y|Z) = P_t(Y|Z)$).*

Proof of Lemma 3 We have:

$$
\begin{aligned}
\log p_s(y|x) &= \log\left(\int p_s(y,z|x)dz\right) \\
&= \log\left(\int p_s(y|z)p(z|x)dz\right) \\
&= \log\left(\mathbb{E}_{P(Z|x)}\left[p_s(y|Z)\right]\right) \\
&\overset{(1)}{\geq} \mathbb{E}_{P(Z|x)}\left[\log p_s(y|Z)\right]
\end{aligned} \tag{10}
$$

We have $\overset{(1)}{\geq}$ by using Jensen's inequality. Taking expectation w.r.t. $P_t(X,Y)$ over both sides, we have:

$$
\begin{aligned}
&\mathbb{E}_{P_t(X,Y)}\left[\log p_s(Y|X) - \mathbb{E}_{P(Z|X)}\left[\log p_s(Y|Z)\right]\right] \\
&= \int\int \left(\log p_s(y|x) - \mathbb{E}_{P(Z|x)}\left[\log p_s(Y|Z)\right]\right) p_t(x,y)dxdy \\
&= \int\int \left(\log p_s(y|x) - \mathbb{E}_{P(Z|x)}\left[\log p_s(Y|Z)\right]\right) p_s(x,y)\frac{p_t(x,y)}{p_s(x,y)}dxdy \\
&= \mathbb{E}_{P_s(X,Y)}\left[\left(\log p_s(Y|X) - \mathbb{E}_{P(Z|X)}\left[\log p_s(Y|Z)\right]\right)\frac{p_t(X,Y)}{p_s(X,Y)}\right] \\
&\overset{(1)}{\leq} \left(\max_{x,y}\frac{p_t(x,y)}{p_s(x,y)}\right)\mathbb{E}_{P_s(X,Y)}\left[\log p_s(Y|X) - \mathbb{E}_{P(Z|X)}\left[\log p_s(Y|Z)\right]\right] \\
&= \left(\max_{x,y}\frac{p_t(x,y)}{p_s(x,y)}\right)\left(\mathbb{E}_{P_s(X,Y)}\left[\log p_s(Y|X)\right] - \mathbb{E}_{P_s(Z,Y)}\left[\log p_s(Y|Z)\right]\right) \\
&= \left(\max_{x,y}\frac{p_t(x,y)}{p_s(x,y)}\right)\left(H_s(Y,X) - H_s(Y,Z)\right) \\
&= \left(\max_{x,y}\frac{p_t(x,y)}{p_s(x,y)}\right)\left(\left(H_s(Y) - H_s(Y,Z)\right) - \left(H_s(Y) - H_s(Y,X)\right)\right) \\
&= \left(\max_{x,y}\frac{p_t(x,y)}{p_s(x,y)}\right)\left(I_s(Y,Z) - I_s(Y,X)\right) \\
&\overset{(2)}{=} 0
\end{aligned} \tag{11}
$$

We have $\overset{(1)}{\leq}$ because $\log p_s(y|x) - \mathbb{E}_{P(Z|x)}\left[\log p_s(y|z)\right] \geq 0$ according to Eq. (10); $\overset{(2)}{=}$ because $I_s(Y,Z) = I_s(Y,X)$ according to Assumption 2. Based on Eq. (11), we have:

$$
\begin{aligned}
\mathbb{E}_{P_t(X,Y)}\left[\log p_s(Y|X)\right] &= \mathbb{E}_{P_t(X,Y)}\left[\mathbb{E}_{P(Z|X)}\left[\log p_s(Y|Z)\right]\right] \\
&= \mathbb{E}_{P_t(Y,Z)}\left[\log p_s(Y|Z)\right]
\end{aligned} \tag{12}
$$

We also have:

$$
\begin{aligned}
\mathbb{E}_{P_t(Y,Z)}\left[\log P_t(Y|Z)\right] &= -H_t(Y|Z) \\
&= I_t(Y,Z) - H_t(Y) \\
&\overset{(1)}{\leq} I_t(Y,X) - H_t(Y) \\
&= -H_t(Y|X) \\
&= \mathbb{E}_{P_t(X,Y)}\left[\log P_t(Y|X)\right]
\end{aligned} \tag{13}
$$

We have $\overset{(1)}{\leq}$ by using data processing inequality. Finally, we have:

$$
\begin{aligned}
&\mathbb{E}_{P_t(Z)}\left[\mathcal{D}_{KL}\left(P_t(Y|Z) \parallel P_s(Y|Z)\right)\right] \\
&\overset{(1)}{=} \mathbb{E}_{P_t(Z)}\left[\mathcal{D}_{KL}\left(P_t(Y|Z) \parallel P_s(Y|Z)\right)\right] - \mathbb{E}_{P_t(X)}\left[\mathcal{D}_{KL}\left(P_t(Y|X) \parallel P_s(Y|X)\right)\right] \\
&= \mathbb{E}_{P_t(Y,Z)}\left[\log p_t(Y|Z) - \log p_s(Y|Z)\right] - \mathbb{E}_{P_t(X,Y)}\left[\log p_t(Y|X) - \log p_s(Y|X)\right] \\
&= \left(\mathbb{E}_{P_t(Y,Z)}\left[\log p_t(Y|Z)\right] - \mathbb{E}_{P_t(X,Y)}\left[\log p_t(Y|X)\right]\right) + \left(\mathbb{E}_{P_t(X,Y)}\left[\log p_s(Y|X)\right] - \mathbb{E}_{P_t(Y,Z)}\left[\log p_s(Y|Z)\right]\right) \\
&\overset{(2)}{=} 0
\end{aligned}
\tag{14}
$$

We have $\overset{(1)}{=}$ because the shift between two domains w.r.t. input space $\mathcal{X}$ is covariate shift; $\overset{(2)}{=}$ by using Eq. (12) and Eq. (13) and the fact that KL-divergence is non-negative. Note that Eq. (14) implies that the shift between these two domains w.r.t. representation space $\mathcal{Z}$ is also covariate shift (i.e., $P_s(Y|Z) = P_t(Y|Z)$).

Lemma 4 *Given domain d associated with a distribution $P_d(X,Y)$, then for any $\delta > 0$, with probability at least $1-\delta$ over sample S of size n drawn i.i.d from domain d, for all $h \in \mathcal{H} : \mathcal{X} \to \Delta(\mathcal{Y})$, the expected error of h in domain d can be upper bounded:*

$$
\mathrm{E}\left(P_d, h\right) \leq \widehat{\mathrm{E}}\left(P_d, h\right) + 2\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}\right) + 3C\sqrt{\frac{\log(2/\delta)}{2n}}
$$

where $\mathcal{L} \circ \mathcal{H} = \{(x,y) \to L\left(h\left(x\right), y\right) : h \in \mathcal{H}\}$ and $\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}\right)$ is an empirical Rademacher complexity of the function class $\mathcal{L} \circ \mathcal{H}$ computed from the sample S.

Proof of Lemma 4 We start from the Rademacher bound (Koltchinskii and Panchenko 2000) which is stated as follows.

Rademacher Bounds. Let $\mathcal{F}$ be a family of functions mapping from Z to $[0,1]$. Then, for any $0 < \delta < 1$, with probability at least $1-\delta$ over sample $S = \{z_1, \cdots, z_n\}$, the following holds for all $f \in \mathcal{F}$:

$$
\mathbb{E}\left[f^Z\right] \leq \frac{1}{n}\sum_{i=1}^{n} f(z_i) + 2\mathcal{R}_S(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}}
$$

where $\mathcal{R}_S\left(\mathcal{F}\right)$ is an empirical Rademacher complexity of function class $\mathcal{F}$ computed from the sample S.

We then apply this result to our setting with $Z = (X,Y)$, the loss function L bounded by C, and the function class $\mathcal{L} \circ \mathcal{H} = \{(x,y) \to L\left(h\left(x\right), y\right) : h \in \mathcal{H}\}$. In particular, we scale the loss function L to $[0,1]$ by dividing by C and denote the new class of scaled loss functions as $\mathcal{L} \circ \mathcal{H}/C$. Then, for any $\delta > 0$, with probability at least $1-\delta$, we have:

$$
\begin{aligned}
\frac{\mathrm{E}\left(P_d, h\right)}{C} &\leq \frac{\widehat{\mathrm{E}}\left(P_d, h\right)}{C} + 2\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}/C\right) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \\
&\overset{(1)}{=} \frac{\widehat{\mathrm{E}}\left(P_d, h\right)}{C} + \frac{2}{C}\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}\right) + 3\sqrt{\frac{\log(2/\delta)}{2n}}
\end{aligned}
\tag{15}
$$

We have $\overset{(1)}{=}$ by using the property of empirical Redamacher complexity that $\mathcal{R}_S(\alpha\mathcal{F}) = \alpha\mathcal{R}_S(\mathcal{F})$. We derive Lemma 4 by multiplying Eq. (15) by C.

Lemma 5 *Given a loss function L satisfied Assumption 1 and a sample $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ of size n, then an empirical Rademacher complexity $\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}\right)$ computed from the sample S is upper bounded as follows.*

$$
\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}\right) \leq C\sqrt{\frac{2d\log n + 4d\log|\mathcal{Y}|}{n}}
$$

where d is Natarajan dimension of hypothesis class $\mathcal{H}$.

Proof of Lemma 5 We have:

$$
\begin{aligned}
\mathcal{R}_S\left(\mathcal{L} \circ \mathcal{H}\right) &\overset{(1)}{\leq} C\sqrt{\frac{2\log|\mathcal{L} \circ \mathcal{H}|}{n}} \\
&\overset{(2)}{\leq} C\sqrt{\frac{2d\log n + 4d\log|\mathcal{Y}|}{n}}
\end{aligned}
$$

We have $\overset{(1)}{\leq}$ by applying Massart's finite lemma (Lemma 5 in Liang (2016)) for function class $\mathcal{L} \circ \mathcal{H}$ and note that $\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i)^2 \leq C^2$ because of Assumption 2; $\overset{(2)}{\leq}$ by using the fact that $|\mathcal{L} \circ \mathcal{H}| \leq |\mathcal{H}|$ and then applying Natarajan lemma (Lemma 29.4 in Shalev-Shwartz and Ben-David (2014)) for hypothesis class $\mathcal{H}$ with Natarajan dimension d.

A.2 Proof of main theorems

Proof of Theorem 1 We have:

$$\begin{aligned}
\mathrm{E}\left(P_t, h\circ f_t\right) &= \mathbb{E}_{P_t(Y,Z)}\left[L(h(Z),Y)\right] \\
&= \int\int L(h(z),y)p_t(y,z)dydz \\
&= \int\int L(h(z),y)\left(p_t(y,z|y\in\mathcal{Y}^s)p_t(y\in\mathcal{Y}^s) + p_t(y,z|y\notin\mathcal{Y}^s)p_t(y\notin\mathcal{Y}^s)\right)dydz \\
&= \lambda\mathbb{E}_{P_{t,k}(Y,Z)}\left[L(h(Z),Y)\right] + (1-\lambda)\mathbb{E}_{P_{t,u}(Y,Z)}\left[L(h(Z),Y)\right] \\
&= \lambda\,\mathrm{E}\left(P_{t,k}, h\circ f_t\right) + (1-\lambda)\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right)
\end{aligned} \tag{16}$$

Applying Lemma 1 for the two distributions $P_{t,k}(Z,Y)$ and $P_s(Z,Y)$, we have:

$$\mathrm{E}\left(P_{t,k}, h\circ f_t\right) \leq \mathrm{E}\left(P_s, h\circ f_s\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Y,Z), P_{t,k}(Y,Z)\right)\right)^{1/2} \tag{17}$$

Let $P_{t,k}^u$ is the distribution induced from $P_{t,k}$ by the mapping f_t^u. Then, we have:

$$\begin{aligned}
\mathrm{E}\left(P_t^u, h\circ f_t\right) &= \mathbb{E}_{P_t^u(Y,Z)}\left[L(h(Z),Y)\right] \\
&= \int\int L(h(z),y)p_t^u(y,z)dydz \\
&\overset{(1)}{=} \int\int L(h(z),y)\left(p_{t,k}^u(y,z)p_t(y\in\mathcal{Y}^s) + p_{t,u}(y,z)p_t(y\notin\mathcal{Y}^s)\right)dydz \\
&= \lambda\mathbb{E}_{P_{t,k}^u(Y,Z)}\left[L(h(Z),Y)\right] + (1-\lambda)\mathbb{E}_{P_{t,u}(Y,Z)}\left[L(h(Z),Y)\right] \\
&= \lambda\,\mathrm{E}\left(P_{t,k}^u, h\circ f_t\right) + (1-\lambda)\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right) \\
&\overset{(2)}{\geq} \lambda\left(\mathrm{E}\left(P_s^u, h\circ f_s\right) - \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_{t,k}^u(Y,Z)\parallel P_s^u(Y,Z)\right)\right)^{1/2}\right) + (1-\lambda)\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right) \\
&\overset{(3)}{\geq} \lambda\left(\mathrm{E}\left(P_s^u, h\circ f_s\right) - \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_{t,k}^u(Z)\parallel P_s^u(Z)\right) + \mathbb{E}_{P_{t,k}^u(Z)}\left[P_{t,k}^u(Y|Z)\parallel P_s^u(Y|Z)\right]\right.\right. \\
&\left.\left. +\ \mathbb{E}_{P_s^u(Z)}\left[P_{t,k}^u(Y|Z)\parallel P_s^u(Y|Z)\right]\right)^{1/2}\right) + (1-\lambda)\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right) \\
&\overset{(4)}{=} \lambda\,\mathrm{E}\left(P_s^u, h\circ f_s\right) - \sqrt{2}\lambda C\left(\mathcal{D}_{JS}\left(P_{t,k}^u(Z)\parallel P_s^u(Z)\right)\right)^{1/2} + (1-\lambda)\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right)
\end{aligned} \tag{18}$$

We have $\overset{(1)}{=}$ by using the fact that $P_t(Y,Z) = \lambda P_{t,k}(Y,Z) + (1-\lambda)P_{t,u}(Y,Z)$ and f_t^u is the mapping such that $f_t^u\left(X^t, Y\right) = (X^t, unk)$; $\overset{(2)}{\geq}$ by using Lemma 1; $\overset{(3)}{\geq}$ by using Lemma 2; $\overset{(4)}{=}$ because $P_{t,k}^u(Y|Z)\parallel P_s^u(Y|Z)$ (support of $Y=\{unk\}$). Finally, by combining Eq. (16), Eq. (17), and Eq. (18), we have:

$$\begin{aligned}
\mathrm{E}\left(P_t, h\circ f_t\right) \leq &\underbrace{\lambda\,\mathrm{E}\left(P_s, h\circ f_s\right)}_{\textbf{source error}} + \underbrace{\mathrm{E}\left(P_t^u, h\circ f_t\right) - \lambda\,\mathrm{E}\left(P_s^u, h\circ f_s\right)}_{\textbf{open-set difference}} \\
&+ \underbrace{\sqrt{2}\lambda C\left(\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}} + \left(\mathcal{D}_{JS}\left(P_s(Z,Y)\parallel P_{t,k}(Z,Y)\right)\right)^{\frac{1}{2}}\right)}_{\textbf{domain distance}}
\end{aligned}$$

Proof of Proposition 1 We have:

$$\begin{aligned}
\mathrm{E}\left(P_t, h\circ f_t\right) &= \mathbb{E}_{P_t(Y,Z)}\left[L(h(Z),Y)\right] \\
&= \int\int L(h(z),y)p_t(y,z)dydz \\
&= \int\int L(h(z),y)\left(p_t(y,z|y\in\mathcal{Y}^s)p_t(y\in\mathcal{Y}^s) + p_t(y,z|y\notin\mathcal{Y}^s)p_t(y\notin\mathcal{Y}^s)\right)dydz \\
&= \lambda\mathbb{E}_{P_{t,k}(Y,Z)}\left[L(h(Z),Y)\right] + (1-\lambda)\mathbb{E}_{P_{t,u}(Y,Z)}\left[L(h(Z),Y)\right] \\
&= \lambda\,\mathrm{E}\left(P_{t,k}, h\circ f_t\right) + (1-\lambda)\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right)
\end{aligned} \tag{19}$$

Next, applying Lemma 1 for the two distributions $P_{t,u}(Y,Z)$ and $P_s^u(Y,Z)$, we have:

$$
\begin{aligned}
\mathrm{E}\left(P_s^u, h\circ f_s\right) &\leq \mathrm{E}\left(P_{t,u}, h\circ f_t\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s^u(Y,Z) \parallel P_{t,u}(Y,Z)\right)\right)^{1/2} \\
&\overset{(1)}{\leq} \mathrm{E}\left(P_{t,u}, h\circ f_t\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s^u(Z) \parallel P_{t,u}(Z)\right)\right. \\
&\left. + \mathbb{E}_{P_s^u(Z)}\left[\mathcal{D}_{JS}\left(P_s^u(Y|Z) \parallel P_{t,u}(Y|Z)\right)\right] + \mathbb{E}_{P_{t,u}(Z)}\left[\mathcal{D}_{JS}\left(P_s^u(Y|Z) \parallel P_{t,u}(Y|Z)\right)\right]\right)^{1/2} \\
&\overset{(2)}{=} \mathrm{E}\left(P_{t,u}, h\circ f_t\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,u}(Z)\right)\right)^{1/2} \qquad (20)
\end{aligned}
$$

We have $\overset{(1)}{\leq}$ by applying Lemma 2; $\overset{(2)}{=}$ because $P_s^u(Y|Z) = P_{t,u}(Y|Z)$ (support of $Y = \{unk\}$) and $P_s^u(Z) = P_s(Z)$. Combining Eq. (19) and Eq. (20), we have:

$$
\mathrm{E}\left(P_t, h\circ f_t\right) \geq \lambda\,\mathrm{E}\left(P_{t,u}, h\circ f_t\right) + (1-\lambda)\,\mathrm{E}\left(P_s^u, h\circ f_s\right) - \sqrt{2}(1-\lambda)C\left(\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,u}(Z)\right)\right)^{1/2}
$$

Proof of Proposition 2 Before giving the proof, we first introduce following definition about adversarial learning for invariant representation.

Definition 1 *Given dataset $D_s = \{x_i^s\}_{i=1}^{n_s}$ and $D_t = \{x_i^t\}_{i=1}^{n_t}$ associated with the distributions $P_s(X_s)$ and $P_t(X_t)$, respectively, the goal of adversarial learning approach for invariant representation is to achieve $\widehat{L}_{adv} = \inf_{\alpha,\beta}\sup_\gamma\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\log\left(D_\gamma(F_\alpha(x_i^s))\right) + \frac{1}{n_t}\sum_{i=1}^{n_t}\log\left(1-D_\gamma(F_\beta(x_i^t))\right)\right)$ where F_α, F_β are the mappings from the feature spaces $\mathcal{X}^s, \mathcal{X}^t$ to the representation space $\mathcal{Z}$ parameterized by $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{B}$, and D_γ are the discriminator parameterized by $\gamma \in \Gamma$.*

Then Proposition 2 are formally state as follows.

Proposition 2 (Formal). Let $\alpha^*, \beta^*, \gamma^*$be the parameters learned by optimizing L_{adv} and $\widehat{\alpha}, \widehat{\beta}, \widehat{\gamma}$ be the parameters learned by optimizing $\widehat{L}_{adv}$. We have:

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{D}_{JS}\left(P_{\widehat{\alpha}}(Z) \parallel P_{\widehat{\beta}}(Z)\right)\right] \leq \mathcal{D}_{JS}\left(P_{\alpha^*}(Z) \parallel P_{\beta^*}(Z)\right) \\
&+ \mathcal{O}\left(\left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right) \times C(\mathcal{A},\mathcal{B},\Gamma)\right)
\end{aligned}
$$

where $C(\mathcal{A},\mathcal{B},\Gamma)$ is a constant specified by the parameter spaces $\mathcal{A},\mathcal{B},\Gamma$, and $L_{adv} = \inf_{\alpha,\beta}\sup_\gamma\int_{\mathcal{X}^s}\log\left(D_\gamma(F_\alpha(x^s))\right)p_s(x^s)dx^s + \int_{\mathcal{X}^t}\log\left(1-D_\gamma(F_\beta(x^t))\right)p_t(x^t)dx^t$ is the objective function of adversarial learning for invariant representation with infinite data.

The proof for Proposition 2 is based on the proof provided for GAN model by Biau et al. (2020). Let $L(\alpha,\beta,\gamma) =$

$\int_{\mathcal{Z}} \left(\log\left(D_\gamma(z)\right) p_\alpha(z) + \log\left(1 - D_\gamma(z)\right) p_\beta(z)\right) dz$, we have:

$$\begin{aligned}
2\mathcal{D}_{JS}\left(P_{\widehat{\alpha}}(Z) \parallel P_{\widehat{\beta}}(Z)\right) &= L(\widehat{\alpha}, \widehat{\beta}, \widehat{\gamma}) + \log(4) \\
&\le \sup_{\gamma} L(\widehat{\alpha}, \widehat{\beta}, \gamma) + \log(4) \\
&\le \sup_{\gamma} \left(\widehat{L}(\widehat{\alpha}, \widehat{\beta}, \gamma) + \left|\widehat{L}(\widehat{\alpha}, \widehat{\beta}, \gamma) - L(\widehat{\alpha}, \widehat{\beta}, \gamma)\right|\right) + \log(4) \\
&\le \sup_{\gamma} \widehat{L}(\widehat{\alpha}, \widehat{\beta}, \gamma) + \sup_{\gamma} \left|\widehat{L}(\widehat{\alpha}, \widehat{\beta}, \gamma) - L(\widehat{\alpha}, \widehat{\beta}, \gamma)\right| + \log(4) \\
&\le \inf_{\alpha,\beta} \sup_{\gamma} \widehat{L}(\alpha, \beta, \gamma) + \sup_{\alpha,\beta,\gamma} \left|\widehat{L}(\alpha, \beta, \gamma) - L(\alpha, \beta, \gamma)\right| + \log(4) \\
&\le \inf_{\alpha,\beta} \sup_{\gamma} L(\alpha, \beta, \gamma) + \left|\inf_{\alpha,\beta} \sup_{\gamma} \widehat{L}(\alpha, \beta, \gamma) - \inf_{\alpha,\beta} \sup_{\gamma} L(\alpha, \beta, \gamma)\right| \\
&+ \sup_{\alpha,\beta,\gamma} \left|\widehat{L}(\alpha, \beta, \gamma) - L(\alpha, \beta, \gamma)\right| + \log(4) \\
&\overset{(1)}{\le} \inf_{\alpha,\beta} \sup_{\gamma} L(\alpha, \beta, \gamma) + \sup_{\alpha,\beta} \left|\sup_{\gamma} \widehat{L}(\alpha, \beta, \gamma) - \sup_{\gamma} L(\alpha, \beta, \gamma)\right| \\
&+ \sup_{\alpha,\beta,\gamma} \left|\widehat{L}(\alpha, \beta, \gamma) - L(\alpha, \beta, \gamma)\right| + \log(4) \\
&\overset{(2)}{\le} \inf_{\alpha,\beta} \sup_{\gamma} L(\alpha, \beta, \gamma) + 2 \sup_{\alpha,\beta,\gamma} \left|\widehat{L}(\alpha, \beta, \gamma) - L(\alpha, \beta, \gamma)\right| + \log(4) \\
&= 2\mathcal{D}_{JS}\left(P_{\alpha^*}(Z) \parallel P_{\beta^*}(Z)\right) + 2 \sup_{\alpha,\beta,\gamma} \left|\widehat{L}(\alpha, \beta, \gamma) - L(\alpha, \beta, \gamma)\right|
\end{aligned}$$

We have $\overset{(1)}{\le}$ by using inequality $|\inf A - \inf B| \le \sup |A - B|$, $\overset{(2)}{\le}$ by using inequality $|\sup A - \sup B| \le \sup |A - B|$. Take the expectation and rearrange the both sides, we have:

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{D}_{JS}\left(P_{\widehat{\alpha}}(Z) \parallel P_{\widehat{\beta}}(Z)\right)\right] - \mathcal{D}_{JS}\left(P_{\alpha^*}(Z) \parallel P_{\beta^*}(Z)\right) \\
&\le \mathbb{E}\left[\sup_{\alpha,\beta,\gamma} \left|\widehat{L}(\alpha, \beta, \gamma) - L(\alpha, \beta, \gamma)\right|\right] \\
&= \mathbb{E}\left[\sup_{\alpha,\beta,\gamma} \left|\frac{1}{n_s} \sum_{i=1}^{n_s} \log\left(D_\gamma((z_i^s))\right) + \frac{1}{n_t} \sum_{i=1}^{n_t} \log\left(1 - D_\gamma(z_i^t)\right)\right.\right. \\
&\left.\left. - \int_{\mathcal{Z}} \left(\log\left(D_\gamma(z)\right) p_\alpha(z) + \log\left(1 - D_\gamma(z)\right) p_\beta(z)\right) dz\right|\right] \\
&\le \mathbb{E}\left[\sup_{\alpha,\beta,\gamma} \left|\underbrace{\frac{1}{n_s} \sum_{i=1}^{n_s} \log\left(D_\gamma((z_i^s))\right) - \int_{\mathcal{Z}} \left(\log\left(D_\gamma(z)\right) p_\alpha(z)\right) dz}_{A_s(\alpha,\beta,\gamma)}\right|\right] \\
&+ \mathbb{E}\left[\sup_{\alpha,\beta,\gamma} \left|\underbrace{\frac{1}{n_t} \sum_{i=1}^{n_t} \log\left(1 - D_\gamma(z_i^t)\right) - \int_{\mathcal{Z}} \left(\log\left(1 - D_\gamma(z)\right) p_\beta(z)\right) dz}_{A_t(\alpha,\beta,\gamma)}\right|\right]
\end{aligned}$$

Note that $\left(A_s(\alpha, \beta, \gamma)\right)_{\alpha \in \mathcal{A}, \beta \in \mathcal{B}, \gamma \in \Gamma}$ and $\left(A_t(\alpha, \beta, \gamma)\right)_{\alpha \in \mathcal{A}, \beta \in \mathcal{B}, \gamma \in \Gamma}$ are the subgaussian processes in the metric spaces $\left(\mathcal{A} \times \mathcal{B} \times \Gamma, C_1 \|\cdot\| / \sqrt{n_s}\right)$ and $\left(\mathcal{A} \times \mathcal{B} \times \Gamma, C_1 \|\cdot\| / \sqrt{n_t}\right)$ where C_1 is a constant and $\|\cdot\|$ is the Euclidean norm on $\mathcal{A} \times \mathcal{B} \times \Gamma$.

Then using Dudley's entropy integral, we have:

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{D}_{JS}\left(P_{\widehat{\alpha}}(Z) \parallel P_{\widehat{\beta}}(Z)\right)\right] - \mathcal{D}_{JS}\left(P_{\alpha^*}(Z) \parallel P_{\beta^*}(Z)\right) \\
&\leq \mathbb{E}\left[\sup_{\alpha,\beta,\gamma} A_s\left(\alpha,\beta,\gamma\right)||\right] + \mathbb{E}\left[\sup_{\alpha,\beta,\gamma} A_t\left(\alpha,\beta,\gamma\right)||\right] \\
&\leq 12\int_0^\infty \left(\sqrt{\log N(\mathcal{A}\times\mathcal{B}\times\Gamma, C\left\|\cdot\right\|/\sqrt{n_s},\epsilon)} + \sqrt{\log N(\mathcal{A}\times\mathcal{B}\times\Gamma, C\left\|\cdot\right\|/\sqrt{n_t},\epsilon)}\right)d\epsilon \\
&= 12C_1\left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right)\int_0^\infty \sqrt{\log N(\mathcal{A}\times\mathcal{B}\times\Gamma, \left\|\cdot\right\|,\epsilon)}d\epsilon \\
&\overset{(3)}{=} 12C_1\left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right)\int_0^{\mathrm{diam}(\mathcal{A}\times\mathcal{B}\times\Gamma)} \sqrt{\log N(\mathcal{A}\times\mathcal{B}\times\Gamma, \left\|\cdot\right\|,\epsilon)}d\epsilon \\
&\overset{(4)}{\leq} 12C_1\left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right)\int_0^{\mathrm{diam}(\mathcal{A}\times\mathcal{B}\times\Gamma)} \sqrt{\log\left(\left(\frac{2C_2\sqrt{\mathrm{dim}(\mathcal{A}\times\mathcal{B}\times\Gamma)}}{\epsilon}\right)^{\mathrm{dim}(\mathcal{A}\times\mathcal{B}\times\Gamma)}\right)}d\epsilon \\
&= \mathcal{O}\left(\left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right)\times C(\mathcal{A},\mathcal{B},\Gamma)\right)
\end{aligned}
$$

where $\mathrm{diam}(\cdot)$ and $\mathrm{dim}(\cdot)$ are the diameter and the dimension of the metric space, and $C(\mathcal{A},\mathcal{B},\Gamma)$ is the function of $\mathrm{diam}(\mathcal{A}\times\mathcal{B}\times\Gamma)$ and $\mathrm{dim}(\mathcal{A}\times\mathcal{B}\times\Gamma)$. We have $\overset{(3)}{=}$ because $N(\mathcal{A}\times\mathcal{B}\times\Gamma,\left\|\cdot\right\|,\epsilon)=1$ for $\epsilon > \mathrm{diam}(\mathcal{A}\times\mathcal{B}\times\Gamma)$, $\overset{(4)}{\leq}$ by using inequality $N(\mathcal{T},\|\cdot\|,\epsilon) \leq \left(\frac{2C_2\sqrt{d}}{\epsilon}\right)^d$ where $\mathcal{T}$ lied in Euclidean space $\mathbb{R}^d$ is the set of vectors whose length is at most C_2.

Proof of Theorem 2 We have:

$$
\begin{aligned}
\mathcal{D}_{JS}\left(P_s(Z,Y) \parallel P_{t,k}(Z,Y)\right) &\overset{(1)}{\leq} \mathcal{D}_{JS}\left(P_s(Z,Y) \parallel P_{t,k}(Z,g(Z))\right) + \mathcal{D}_{JS}\left(P_{t,k}(Z,g(Z)) \parallel P_{t,k}(Z,Y)\right) \\
&\overset{(2)}{\leq} \mathcal{D}_{JS}\left(P_s(Z,Y) \parallel P_{t,k}(Z,g(Z))\right) + \mathcal{D}_{JS}\left(P_{t,k}(Z) \parallel P_{t,k}(Z)\right) \\
&+ \mathbb{E}_{P_{t,k}(Z)}\left[\mathcal{D}_{JS}\left(P_{t,k}(g(Z)|Z) \parallel P_{t,k}(Y|Z)\right)\right] \\
&= \mathcal{D}_{JS}\left(P_s(Z,Y) \parallel P_{t,k}(Z,g(Z))\right) + \mathrm{N}\left(P_{t,k},g\right)
\end{aligned} \tag{21}
$$

We have $\overset{(1)}{\leq}$ by using triangle inequality for JS-divergence; $\overset{(2)}{\leq}$ by applying Lemma 2. According to Theorem 1, we have:

$$
\begin{aligned}
\mathrm{E}\left(P_t,h\circ f_t\right) &\leq \lambda\,\mathrm{E}\left(P_s,h\circ f_s\right) + \mathrm{E}\left(P_t^u,h\circ f_t\right) - \lambda\,\mathrm{E}\left(P_s^u,h\circ f_s\right) \\
&+ \sqrt{2}\lambda C\left(\left(\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}} + \left(\mathcal{D}_{JS}\left(P_s(Z,Y) \parallel P_{t,k}(Z,Y)\right)\right)^{\frac{1}{2}}\right) \\
&\overset{(1)}{\leq} \lambda\,\mathrm{E}\left(P_s,h\circ f_s\right) + \mathrm{E}\left(P_t^u,h\circ f_t\right) - \lambda\,\mathrm{E}\left(P_s^u,h\circ f_s\right) \\
&+ \sqrt{2}\lambda C\left(\left(\mathcal{D}_{JS}\left(P_s(Z) \parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}} + \left(\mathcal{D}_{JS}\left(P_s(Z,Y) \parallel P_{t,k}(Z,g(Z))\right)\right)^{\frac{1}{2}} + \left(\mathrm{N}\left(P_{t,k},g\right)\right)^{\frac{1}{2}}\right)
\end{aligned} \tag{22}
$$

We have $\overset{(1)}{\leq}$ by applying Eq. (21) and using inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. Finally, by applying Lemma 4 for $\mathrm{E}\left(P_s,h\circ f_s\right)$, $\mathrm{E}\left(P_t^u,h\circ f_t\right)$, and $\mathrm{E}\left(P_s^u,h\circ f_s\right)$ in Eq. (22), then with probability at least $1-\delta$ over the choice of source and target datasets D_s and D_t, we have:

$$
\begin{aligned}
\mathrm{E}\left(P_t, h\circ f_t\right) &\leq \lambda\widehat{\mathrm{E}}\left(P_s, h\circ f_s\right) + \widehat{\mathrm{E}}\left(P_t^u, h\circ f_t\right) - \lambda\widehat{\mathrm{E}}\left(P_s^u, h\circ f_s\right)\\
&+ \sqrt{2}\lambda C\left(\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}} + \left(\mathcal{D}_{JS}\left(P_s(Z,Y)\parallel P_{t,k}(Z,g(Z))\right)\right)^{\frac{1}{2}} + \left(\mathrm{N}\left(P_{t,k}, g\right)\right)^{\frac{1}{2}}\right)\\
&+ 2\lambda\mathcal{R}_{D_s}\left(\mathcal{L}\circ\mathcal{H}\circ\mathcal{F}_s\right) + 2\mathcal{R}_{D_t^u}\left(\mathcal{L}\circ\mathcal{H}\circ\mathcal{F}_t\right) + 2\lambda\mathcal{R}_{D_s^u}\left(\mathcal{L}\circ\mathcal{H}\circ\mathcal{F}_s\right) + \mathcal{O}\left(C\sqrt{\log(1/\delta)}\left(\frac{\lambda}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right)\right)\\
&\overset{(1)}{\leq} \lambda\widehat{\mathrm{E}}\left(P_s, h\circ f_s\right) + \widehat{\mathrm{E}}\left(P_t^u, h\circ f_t\right) - \lambda\widehat{\mathrm{E}}\left(P_s^u, h\circ f_s\right)\\
&+ \sqrt{2}\lambda C\left(\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_{t,k}(Z)\right)\right)^{\frac{1}{2}} + \left(\mathcal{D}_{JS}\left(P_s(Z,Y)\parallel P_{t,k}(Z,g(Z))\right)\right)^{\frac{1}{2}} + \left(\mathrm{N}(P_{t,k}, g)\right)^{\frac{1}{2}}\right)\\
&+ \mathcal{O}\left(\lambda C\sqrt{\frac{d_s\log n_s + d_s\log|\mathcal{Y}^t| + \log\frac{1}{\delta}}{n_s}} + C\sqrt{\frac{d_t\log n_t + d_t\log|\mathcal{Y}^t| + \log\frac{1}{\delta}}{n_t}}\right)
\end{aligned}
\tag{23}
$$

where D_s^u and D_t^u are datasets induced from D_s and D_t using mappings f_s^u and f_t^u, respectively. We have $\overset{(1)}{\leq}$ by applying Lemma 5 for $\mathcal{R}_{D_s}\left(\mathcal{L}\circ\mathcal{H}\circ\mathcal{F}_s\right)$, $\mathcal{R}_{D_t^u}\left(\mathcal{L}\circ\mathcal{H}\circ\mathcal{F}_t\right)$, and $\mathcal{R}_{D_s^u}\left(\mathcal{L}\circ\mathcal{H}\circ\mathcal{F}_s\right)$.

Proof of Proposition 3 We have:

$$
\begin{aligned}
\mathrm{E}\left(P_t, h\circ f\right) &\overset{(1)}{\leq} \mathrm{E}\left(P_s, h\circ f\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Y,Z)\parallel P_t(Y,Z)\right)\right)^{1/2}\\
&\overset{(2)}{\leq} \mathrm{E}\left(P_s, h\circ f\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_t(Z)\right) + 2\mathbb{E}_{P_{s,t}}\left[\mathcal{D}_{JS}\left(P_s(Y|Z)\parallel P_t(Y|Z)\right)\right]\right)^{1/2}\\
&\overset{(3)}{=} \mathrm{E}\left(P_s, h\circ f\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_t(Z)\right) + 2\mathbb{E}_{P_{s,t}}\left[\mathcal{D}_{JS}\left(P_s(Y|Z)\parallel P_s(Y|Z)\right)\right]\right)^{1/2}\\
&= \mathrm{E}\left(P_s, h\circ f\right) + \sqrt{2}C\left(\mathcal{D}_{JS}\left(P_s(Z)\parallel P_t(Z)\right)\right)^{1/2}
\end{aligned}
$$

We have $\overset{(1)}{\leq}$ by applying Lemma 1 for two distributions $P_s(Z,Y)$ and $P_t(Z,Y)$, and note that $\mathrm{E}\left(P_d, h\circ f\right) = \mathbb{E}_{P_d(X,Y)}\left[L(h(f(X)),Y)\right] = \mathbb{E}_{P_d(Z,Y)}\left[L(h(Z),Y)\right] = \mathrm{E}\left(P_d, h\right)$ for $d\in\{s,t\}$; $\overset{(2)}{\leq}$ by applying Lemma 2 for $\mathcal{D}_{JS}\left(P_s(Y,Z)\parallel P_t(Y,Z)\right)$; $\overset{(3)}{=}$ by applying Lemma 3.

B Model Details

In this section, we provide a comprehensive overview of the architectures utilized in our experiment, along with the pseudocode for training our proposed method.

B.1 Backbone Architecture Details

To ensure fair comparisons between methods, we use the same backbone architectures across all approaches. Most methods involve representation mapping and classifier networks, except for SSAN and SCT, which include an additional discriminator network, and KPG, which employs optimal transport to transform data within the input space. For representation mapping, we implement multi-layer feed-forward neural networks on the CIFAR10 & ILSVRC2012, Wikipedia, Multilingual Reuters Collection, NUSWIDE & ImageNet, Office & Caltech256, and ImageCLEF-DA datasets, while using ResNet/WideResNet to encode paper and digital ECG data in the PTB-XL dataset. A linear classifier is applied across all datasets. The specifics of these networks are detailed in Tables 3-6 below.

Table 3: Details of backbone networks used in CIFAR10 & ILSVRC2012, Wikipedia, Multilingual Reuters Collection, NUSWIDE & ImageNet, Office & Caltech256, and ImageCLEF-DA datasets. **d_source**, **d_target**, and **n_output** represent the dimensions of the source feature space, target feature space, and output space, respectively. For RL-OSHeDA and OPDA, **n_output** is set to $|\mathcal{Y}^t|$ while for other methods, **n_output** is set to $|\mathcal{Y}^s|$.

Networks	Layers
Representation Mapping f_s	Linear(input dim=**d_source**, output dim=(**d_source** + 256)/2) LeakyReLU(negative_slope=0.2) Linear(input dim=(**d_source** + 256)/2, output dim=256) LeakyReLU(negative_slope=0.2) Normalize(p=2)
Representation Mapping f_t	Linear(input dim=**d_target**, output dim=(**d_target** + 256)/2) LeakyReLU(negative_slope=0.2) Linear(input dim=(**d_target** + 256)/2, output dim=256) LeakyReLU(negative_slope=0.2) Normalize(p=2)
Classifier h	Linear(input dim=256, output dim=**n_output**) LeakyReLU(negative_slope=0.2)

Table 4: Details of backbone networks used in PTB-XL dataset. The representation mapping networks are constructed from multiple ResNetBlock1d (for digital ECG) and ResNetBlock2d (for paper ECG) sub-modules. **n_output** represent the dimensions of output space. For RL-OSHeDA and OPDA, **n_output** is set to $|\mathcal{Y}^t|$ while for other methods, **n_output** is set to $|\mathcal{Y}^s|$.

Networks	Layers
ResNetBlock1d(i_ch, o_ch)	Conv1d(input channel=**i_ch**, output channel=**o_ch**, kernel=3, stride=**o_ch / i_ch**, padding=1) BatchNorm1d ReLU Conv1d(input channel=**o_ch**, output channel=**o_ch**, kernel=3, padding=1) BatchNorm1d ReLU
ResNetBlock2d(i_ch, o_ch)	Conv2d(input channel=**i_ch**, output channel=**o_ch**, kernel=3, stride=**o_ch / i_ch**, padding=1) BatchNorm2d ReLU Conv2d(input channel=**o_ch**, output channel=**o_ch**, kernel=3, padding=1) BatchNorm2d ReLU
ResNet1d	Conv1d(input channel=1, output channel=64, kernel=7, stride=2, padding=3) BatchNorm1d ReLU MaxPool1d ResNetBlock1d(i_ch=64, o_ch=64) $\times$ 2 ResNetBlock1d(i_ch=64, o_ch=128) ResNetBlock1d(i_ch=128, o_ch=128) ResNetBlock1d(i_ch=128, o_ch=256) ResNetBlock1d(i_ch=256, o_ch=256) ResNetBlock1d(i_ch=256, o_ch=512) ResNetBlock1d(i_ch=512, o_ch=512) BatchNorm1d ReLU AdaptiveAvgPool1d(output size=7) Linear(input dim=3584, output dim=256)
Representation Mapping f_s	Concatenation of 12 ResNet1d modules Linear(input dim=256 $\times$ 12, output dim=256)
Representation Mapping f_t	Conv2d(input channel=3, output channel=64, kernel=7, stride=2, padding=3) BatchNorm2d ReLU MaxPool2d ResNetBlock2d(i_ch=64, o_ch=64) $\times$ 2 ResNetBlock2d(i_ch=64, o_ch=128) ResNetBlock2d(i_ch=128, o_ch=128) ResNetBlock2d(i_ch=128, o_ch=256) ResNetBlock2d(i_ch=256, o_ch=256) ResNetBlock2d(i_ch=256, o_ch=512) ResNetBlock2d(i_ch=512, o_ch=512) AdaptiveAvgPool2d(output size=1) Linear(input dim=512, output dim=256)
Classifier h	Linear(input dim=256, output dim=**n_output**) LeakyReLU(negative_slope=0.2)

Table 5: Details of discriminator network used in SSAN and SCT methods.

Networks	Layers
Discriminator D	Linear(input dim=256, output dim=1) Sigmoid

Table 6: Details of the WideResNet backbone. This network is employed in DS3L for the PTB-XL dataset, in place of the standard ResNet backbone.

Networks	Layers
ResNetUnit(i_ch, o_ch)	BatchNorm2d
	LeakyReLU(negative_slope=0.1)
	Conv2d(input channel=**i_ch**, output channel=**o_ch**, kernel=3, stride=**o_ch / i_ch**, padding=1)
	BatchNorm2d
	LeakyReLU(negative_slope=0.1)
	Conv2d(input channel=**o_ch**, output channel=**o_ch**, kernel=3, stride=1, padding=1)
ResNetBlock(i_ch, o_ch)	ResNetUnit(i_ch=**i_ch**, o_ch=**o_ch**)
	ResNetUnit(i_ch=**o_ch**, o_ch=**o_ch**) $\times$ 3
Representation Mapping f_t	Conv2d(input channel=3, output channel=16, kernel=3, padding=1)
	ResNetBlock(i_ch=16, o_ch=32)
	ResNetBlock(i_ch=32, o_ch=64)
	ResNetBlock(i_ch=64, o_ch=128)
	BatchNorm2d
	LeakyReLU(negative_slope=0.1)
	AveragePool2d(output size=1)
	Linear(input dim=128, output dim=256)

B.2 Training Details

We utilize a two-stage learning approach for our proposed method, RL-OSHeDA. Specifically, in stage 1, we update the model parameters by optimizing L_{cls} as defined in Eq. (2). In stage 2, we update the model parameters by optimizing L as specified in Eq. (1), with the assistance of pseudo-labels. The details of this training process are outlined in Algorithm 1 below.

Algorithm 1: Two-stage learning process for RL-OSHeDA

Inputs: Source datasets $D_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$, labeled target datasets $D_{t_l} = \{x_i^t, y_i^t\}_{i=1}^{n_{t_l}}$, unlabeled target datasets $D_{t_u} = \{x_i^t\}_{i=1}^{n_{t_u}}$, known class prior λ, threshold T, number of epochs T_{max}
Outputs: Trained network parameters $\theta_{f_s}, \theta_{f_t}, \theta_h$
Initialize network parameters $\theta_{f_s}, \theta_{f_t}, \theta_h$
for $epoch = 1$ **to** T_{max} **do**
  Sample source minibatch B_s
  Sample labeled target minibatch B_{t_l}
  Sample unlabeled target minibatch B_{t_u}
  if $epoch < T$ **then**
    /* Stage 1 */
    Calculate L_{cls} in Eq. (2) using B_s, B_{t_l}, B_{t_u}, λ
    Update parameters:
    $\theta_{f_s} = \theta_{f_s} - \gamma \nabla_{\theta_{f_s}} L_{cls}$
    $\theta_{f_t} = \theta_{f_t} - \gamma \nabla_{\theta_{f_t}} L_{cls}$
    $\theta_h = \theta_h - \gamma \nabla_{\theta_h} L_{cls}$
  else
    /* Stage 2 */
    Generate pseudo-labels for B_{t_u}
    Calculate L_{cls} in Eq. (2) using B_s, B_{t_l}, B_{t_u}, λ
    Calculate L_{inv} in Eq. (3) using B_s, B_{t_l}, B_{t_u}, pseudo-labels
    Calculate L_{seg} in Eq. (4) using B_s, B_{t_l}, B_{t_u}, pseudo-labels
    Calculate L_{osd} in Eq. (5) using B_s, B_{t_l}, B_{t_u}, λ
    Update parameters:
    $\theta_{f_s} = \theta_{f_s} - \gamma \nabla_{\theta_{f_s}} (L_{cls} + L_{inv} - L_{seg} + L_{osd})$
    $\theta_{f_t} = \theta_{f_t} - \gamma \nabla_{\theta_{f_t}} (L_{cls} + L_{inv} - L_{seg} + L_{osd})$
    $\theta_h = \theta_h - \gamma \nabla_{\theta_h} (L_{cls} + L_{inv} - L_{seg} + L_{osd})$
  end if
end for
return Trained parameters $\theta_{f_s}, \theta_{f_t}, \theta_h$

C Experimental Details

C.1 Datasets

We conduct experiments using seven datasets covering the clinical, computer vision, and natural language processing domains. Detailed descriptions of these datasets are provided below, with corresponding data statistics presented in Tables 7-13.

CIFAR-10 (Krizhevsky 2009) & ILSVRC2012 (Russakovsky et al. 2015). These two datasets are used for the image-to-image adaptation task. Big Transfer-M with ResNet-50 and ResNet-101 (Kolesnikov et al. 2019) are utilized to extract features from the images. In the target domain (ILSVRC2012), 4 out of 8 shared classes are designated as unknown classes. For the source (CIFAR-10) and unlabeled target data, 50 instances are randomly selected for each class, and we randomly choose 1, 3, or 5 instances per class as labeled target data. This process results in 6 DA tasks.

Wikipedia (Rasiwasia et al. 2010). This dataset, consisting of text-image pairs, is used for image-to-text and text-to-image adaptation tasks. Big Transfer-M with ResNet-101 and Big Bird (Zaheer et al. 2020) are used to extract features for image and text, respectively. 5 out of 10 classes are designated as unknown classes. All data in the source domain are used as labeled source data. For the target domain, we randomly select 5 instances per class as labeled target data and randomly select 50 instances per class from the remaining data as unlabeled target data. This process results in 2 DA tasks.

Multilingual Reuters Collection (Amini, Usunier, and Goutte 2009). This dataset, consisting of articles in 5 languages, is used for text-to-text adaptation tasks. Bag-of-Words with TF-IDF, followed by Principal Component Analysis, is used to generate features for each article. English, French, Italian, and German are used as the source domains, while Spanish is used as the target domain. 3 out of 6 classes are designated as unknown classes. For the source and unlabeled target datasets, 100 and 500 instances are randomly selected for each class, respectively, and we randomly choose 20 instances per class as labeled target data. This process results in 4 DA tasks.

NUS-WIDE (Chua et al. 2009) and ImageNet (Deng et al. 2009). These datasets are used for the text-to-image adaptation task. We utilize the tag information from NUS-WIDE as the source domain (text) and the image data from ImageNet as the target domain (image). 4 out of 8 shared classes between the two datasets are designated as unknown classes. Features for NUS-WIDE tags are extracted using a pre-trained 5-layer neural network, while DeCAF6 (Donahue et al. 2014) features are used for images in ImageNet. For NUS-WIDE, 100 instances per class are selected, whereas for ImageNet, 3 instances per class are sampled as labeled target data, with all remaining images used as unlabeled target data.

Office (Saenko et al. 2010) and Caltech-256 (Griffin et al. 2007). These datasets, which include 4 domains Amazon, Webcam, DSLR from Office, and Caltech from Caltech-256, are used for the image-to-image adaptation task. Amazon, Webcam, and Caltech are used as source domains while Amazon, Webcam, DSLR, and Caltech are used as target domains. SURF (Saenko et al. 2010) and DeCAF6 are utilized as 2 different feature sets for images in these datasets. 5 out of 10 classes are designated as unknown classes, and 3 instances per class are sampled as labeled target data. This process results in 18 DA tasks.

ImageCLEF-DA (Griffin et al. 2007). This data, which include 4 domains Caltech, ImageNet, Bing, and PascalVOC, are used for the image-to-image adaptation task. ResNet50 and VGG-19 (Simonyan and Zisserman 2014) are utilized as 2 different feature sets for images in this dataset. 6 out of 12 classes are designated as unknown classes, and 3 instances per class are sampled as labeled target data. This process results in 24 DA tasks.

PTB-XL (Wagner et al. 2020). This dataset is used for the digital-to-paper electrocardiogram (ECG) adaptation task. Frequent classes including NORM, Old MI, STTC, and CD are designated as known classes, while the remaining classes are considered unknown. For each known class, we sample 2,000 instances per class to construct source (1,000 instances) and target (1,000 instances) datasets. All instances from the unknown class are used for the target dataset. This process results in 1 DA task.

Table 7: Data statistics for each domain adaptation task in CIFAR10 & ILSVRC2012 dataset.

CIFAR10 & ILSVRC2012									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (1)	4	2000	2048	4	4	2048	5	4000	2048
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (3)	4	2000	2048	4	12	2048	5	4000	2048
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (5)	4	2000	2048	4	20	2048	5	4000	2048
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (1)	4	2000	2048	4	4	2048	5	4000	2048
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (3)	4	2000	2048	4	12	2048	5	4000	2048
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (5)	4	2000	2048	4	20	2048	5	4000	2048

Table 8: Data statistics for each domain adaptation task in Wikipedia dataset.

Wikipedia									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
Image ⇒ Text	5	1472	768	5	25	2048	6	500	2048
Text ⇒ Image	5	1472	2048	5	25	768	6	500	768

Table 9: Data statistics for each domain adaptation task in Multilingual Reuters Collection dataset.

Multilingual Reuters Collection									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
English ⇒ Spanish	3	300	1131	3	60	807	4	2910	807
French ⇒ Spanish	3	300	1230	3	60	807	4	2910	807
German ⇒ Spanish	3	300	1417	3	60	807	4	2910	807
Italian ⇒ Spanish	3	300	1041	3	60	807	4	2910	807

Table 10: Data statistics for each domain adaptation task in NUSWIDE & ImageNet dataset.

NUSWIDE & ImageNet									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
Text ⇒ Image	4	400	64	4	12	4096	5	800	4096

Table 11: Data statistics for each domain adaptation task in Office & Caltech256 dataset.

Office & Caltech256									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
Amazon (DeCAF6) ⇒ Caltech (SURF)	5	100	4096	5	15	800	6	1093	800
Amazon (DeCAF6) ⇒ DSLR (SURF)	5	100	4096	5	15	800	6	127	800
Amazon (DeCAF6) ⇒ Webcam (SURF)	5	100	4096	5	15	800	6	265	800
Amazon (SURF) ⇒ Caltech (DeCAF6)	5	100	800	5	15	4096	6	1093	4096
Amazon (SURF) ⇒ DSLR (DeCAF6)	5	100	800	5	15	4096	6	127	4096
Amazon (SURF) ⇒ Webcam (DeCAF6)	5	100	800	5	15	4096	6	265	4096
Caltech (DeCAF6) ⇒ Amazon (SURF)	5	100	4096	5	15	800	6	928	800
Caltech (DeCAF6) ⇒ DSLR (SURF)	5	100	4096	5	15	800	6	127	800
Caltech (DeCAF6) ⇒ Webcam (SURF)	5	100	4096	5	15	800	6	265	800
Caltech (SURF) ⇒ Amazon (DeCAF6)	5	100	800	5	15	4096	6	928	4096
Caltech (SURF) ⇒ DSLR (DeCAF6)	5	100	800	5	15	4096	6	127	4096
Caltech (SURF) ⇒ Webcam (DeCAF6)	5	100	800	5	15	4096	6	265	4096
Webcam (DeCAF6) ⇒ Amazon (SURF)	5	100	4096	5	15	800	6	928	800
Webcam (DeCAF6) ⇒ Caltech (SURF)	5	100	4096	5	15	800	6	1093	800
Webcam (DeCAF6) ⇒ DSLR (SURF)	5	100	4096	5	15	800	6	127	800
Webcam (SURF) ⇒ Amazon (DeCAF6)	5	100	800	5	15	4096	6	928	4096
Webcam (SURF) ⇒ Caltech (DeCAF6)	5	100	800	5	15	4096	6	1093	4096
Webcam (SURF) ⇒ DSLR (DeCAF6)	5	100	800	5	15	4096	6	127	4096

Table 12: Data statistics for each domain adaptation task in ImageCLEF-DA dataset.

ImageCLEF-DA									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
Bing (Reset-50) ⇒ Caltech (VGG-19)	6	120	2048	6	18	4096	7	564	4096
Bing (Reset-50) ⇒ ImageNet (VGG-19)	6	120	2048	6	18	4096	7	564	4096
Bing (Reset-50) ⇒ PascalVOC (VGG-19)	6	120	2048	6	18	4096	7	564	4096
Bing (VGG-19) ⇒ Caltech (Reset-50)	6	120	4096	6	18	2048	7	564	2048
Bing (VGG-19) ⇒ ImageNet (Reset-50)	6	120	4096	6	18	2048	7	564	2048
Bing (VGG-19) ⇒ PascalVOC (Reset-50)	6	120	4096	6	18	2048	7	564	2048
Caltech (Reset-50) ⇒ Bing (VGG-19)	6	120	2048	6	18	4096	7	564	4096
Caltech (Reset-50) ⇒ ImageNet (VGG-19)	6	120	2048	6	18	4096	7	564	4096
Caltech (Reset-50) ⇒ PascalVOC (VGG-19)	6	120	2048	6	18	4096	7	564	4096
Caltech (VGG-19) ⇒ Bing (Reset-50)	6	120	4096	6	18	2048	7	564	2048
Caltech (VGG-19) ⇒ ImageNet (Reset-50)	6	120	4096	6	18	2048	7	564	2048
Caltech (VGG-19) ⇒ PascalVOC (Reset-50)	6	120	4096	6	18	2048	7	564	2048
ImageNet (Reset-50) ⇒ Bing (VGG-19)	6	120	2048	6	18	4096	7	564	4096
ImageNet (Reset-50) ⇒ Caltech (VGG-19)	6	120	2048	6	18	4096	7	564	4096
ImageNet (Reset-50) ⇒ PascalVOC (VGG-19)	6	120	2048	6	18	4096	7	564	4096
ImageNet (VGG-19) ⇒ Bing (Reset-50)	6	120	4096	6	18	2048	7	564	2048
ImageNet (VGG-19) ⇒ Caltech (Reset-50)	6	120	4096	6	18	2048	7	564	2048
ImageNet (VGG-19) ⇒ PascalVOC (Reset-50)	6	120	4096	6	18	2048	7	564	2048
PascalVOC (Reset-50) ⇒ Bing (VGG-19)	6	120	2048	6	18	4096	7	564	4096
PascalVOC (Reset-50) ⇒ Caltech (VGG-19)	6	120	2048	6	18	4096	7	564	4096
PascalVOC (Reset-50) ⇒ ImageNet (VGG-19)	6	120	2048	6	18	4096	7	564	4096
PascalVOC (VGG-19) ⇒ Bing (Reset-50)	6	120	4096	6	18	2048	7	564	2048
PascalVOC (VGG-19) ⇒ Caltech (Reset-50)	6	120	4096	6	18	2048	7	564	2048
PascalVOC (VGG-19) ⇒ ImageNet (Reset-50)	6	120	4096	6	18	2048	7	564	2048

Table 13: Data statistics for each domain adaptation task in PTB-XL dataset.

PTB-XL									
	Source Dataset			Labeled Target Dataset			Unlabeled Target Dataset		
	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim	# of classes	# of instances	feature dim
Digital ECG ⇒ Paper ECG	4	4000	12×5000	4	80	$3\times224\times224$	5	4602	$3\times224\times224$

C.2 Baselines

We experiment with diverse baselines from heterogeneous domain adaptation, open-set domain adaptation, open-set semi-supervised learning, supervised learning, and semi-supervised learning. The details of these baselines are as follows.

Heterogeneous Domain Adaptation. Heterogeneous domain adaptation methods are trained on both source and target data. During inference, these methods classify instances as unknown using the same method as our pseudo-label model g (see Section 5.2).

- SSAN (Li et al. 2020): This method maps heterogeneous source features into a shared representation space and then makes predictions from this space. To adapt from the source to the target domains, it aligns the marginal and label-conditional representation distributions between the two domains using Maximum Mean Discrepancy (MMD). Additionally, it employs pseudo-labels generated from geometric similarity and hard predictions made by the classifier.
- STN (Yao et al. 2019): This method is similar to SSAN, but it calculates the MMD distance using soft labels (i.e., softmax probabilities) rather than hard labels.
- SCT (Zhao et al. 2022): This method is similar to SSAN but differs in that it aligns marginal and label-conditional representation distributions between source and target domains using cosine similarity. Additionally, it generates pseudo-labels based on geometric distances from the source data.
- KPG (Gu et al. 2022): This method utilizes partial optimal transport and the Gromov-Wasserstein distance to map features from the source domain to the target domain. An SVM, trained on the transported source data and labeled target data, is then used for making predictions.

Open-Set Domain Adaptation.

- OPDA (Saito et al. 2018): This method is trained exclusively on target data. It employs adversarial training to train a classifier that distinguishes between labeled and unlabeled target samples, while a generator is trained to push the unlabeled samples away from the decision boundary. This setup provides the generator with two options: aligning unlabeled samples with labeled ones or classifying them as unknown. Consequently, this approach enables the extraction of features that effectively differentiate between known and unknown target samples. Unlike other baselines, OPDA can directly classify instances as unknown during inference based on the classifier's output.

Open-Set Semi-Supervised Learning.

- DS3L (Guo et al. 2020): This method is trained exclusively on target data and selectively uses unlabeled data while monitoring its impact to mitigate performance risks. Specifically, DS3L weakens the influence of unlabeled data with unknown classes to enhance distribution matching and maintain strong generalization. Simultaneously, it reinforces the use of labeled data to prevent performance degradation. These considerations are integrated into a unified bi-level optimization framework. During inference, DS3L classifies instances as unknown using the same method as our pseudo-label model g.

Supervised Learning.

- Supervised Learning (SL): We train the model directly on the labeled target dataset by minimizing the classification loss. During inference, SL classifies instances as unknown using the same method as our pseudo-label model g.

Semi-Supervised Learning.

- Pseudo Labeling (PL): Similar to supervised learning, but we use the model to generate pseudo-labels for the unlabeled target data and then incorporate these pseudo-labeled examples into the training process. During inference, PL classifies instances as unknown using the same method as our pseudo-label model g.

C.3 Implementation Details

Data, model implementation, and training script are included in the code & data supplementary material. We train each model on each domain adaptation task with 10 different random seeds and report the average prediction performances. All experiments are conducted on a machine with 24-Core CPU, 4 RTX A4000 GPUs, and 128G RAM.

C.4 Additional Results

In this section, we provide a comprehensive overview of the results for domain adaptation tasks across seven datasets. Detailed results are presented in Tables 14-20. Additionally, Table 21 includes the results of statistical tests used to assess the significance of our method in comparison to the baseline approaches across all domain adaptation tasks.

Table 14: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in CIFAR10 & ILSVRC2012 dataset.

CIFAR10 & ILSVRC2012									
	DS3L			KPG			OPDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (1)	52.69±0.74	48.53±0.93	57.77±1.10	56.92±0.48	53.95±0.00	60.33±1.11	45.42±0.75	38.96±0.90	54.83±1.07
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (3)	65.87±0.72	64.34±0.99	67.49±1.03	57.25±0.48	54.98±0.00	59.77±1.12	57.34±0.81	53.38±1.06	62.25±1.10
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (5)	66.98±0.73	66.02±1.01	67.97±1.08	57.32±0.52	55.73±0.00	59.02±1.13	58.35±0.79	54.74±1.10	62.65±1.13
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (1)	52.70±0.78	47.72±1.11	59.02±1.07	52.68±0.51	47.70±0.00	59.53±1.14	44.22±0.76	36.41±0.94	56.61±1.15
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (3)	62.04±0.73	59.69±0.98	64.74±1.06	58.77±0.51	56.53±0.00	61.26±1.09	53.16±0.73	47.59±0.94	60.50±1.08
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (5)	68.67±0.72	67.95±0.99	69.42±1.01	60.70±0.53	59.52±0.00	61.92±1.10	61.28±0.77	58.23±1.08	64.73±1.07
	PL			SCT			SSAN		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (1)	36.84±0.61	29.92±0.67	49.36±1.12	51.80±0.78	47.85±1.03	56.66±1.16	55.23±0.76	53.41±0.97	57.31±1.15
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (3)	47.34±0.57	42.81±0.64	53.82±1.10	62.68±0.75	61.22±0.97	64.28±1.11	62.34±0.71	61.34±0.98	63.39±1.05
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (5)	43.77±0.55	38.15±0.49	53.93±1.10	65.04±0.74	64.19±1.00	65.92±1.07	63.22±0.74	62.71±1.04	63.74±1.09
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (1)	35.49±0.34	27.96±0.27	51.08±1.12	49.74±0.78	45.12±1.07	55.76±1.07	54.36±0.78	51.13±1.11	58.58±1.11
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (3)	44.72±0.51	39.19±0.52	52.61±1.05	60.24±0.71	58.16±0.94	62.55±1.08	60.57±0.70	59.28±0.97	62.06±1.03
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (5)	48.35±0.51	44.71±0.38	53.07±1.10	68.19±0.72	67.61±0.96	68.79±1.02	66.58±0.70	66.19±0.95	66.97±1.03
	STN			SL			OSHeDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (1)	55.22±0.75	51.92±1.01	59.13±1.08	52.90±0.74	48.24±0.96	58.70±1.14	**60.45±0.79**	**56.70±1.04**	**65.18±1.16**
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (3)	63.51±0.69	60.68±0.90	66.81±1.10	63.97±0.69	62.39±0.89	65.65±1.07	**75.41±0.70**	**71.26±0.94**	**80.43±0.95**
ImageNet (ResNet-101) ⇒ CIFAR (ResNet-50) (5)	65.62±0.75	63.47±1.00	67.97±1.09	66.18±0.72	65.18±1.02	67.21±1.07	**78.24±0.68**	**73.49±1.00**	**83.70±0.86**
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (1)	55.27±0.74	51.39±1.03	60.18±1.09	51.63±0.81	46.92±1.14	57.71±1.09	**62.43±0.77**	**55.94±1.10**	**71.40±1.08**
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (3)	62.09±0.72	59.28±1.03	65.30±0.99	61.38±0.74	59.29±1.01	63.69±1.09	**74.82±0.66**	**70.87±0.91**	**79.50±0.96**
ImageNet (ResNet-50) ⇒ CIFAR (ResNet-101) (5)	67.84±0.64	66.06±0.89	69.81±0.92	68.38±0.74	67.72±1.01	69.06±1.00	**82.65±0.60**	**79.06±0.89**	**86.67±0.79**

Table 15: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in Multilingual Reuters Collection dataset.

Multilingual Reuters Collection									
	DS3L			KPG			OPDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
English ⇒ Spanish	59.35±0.96	52.92±1.31	67.57±1.20	11.48±0.08	8.80±0.00	17.12±0.95	56.42±1.02	49.09±1.32	66.41±1.23
French ⇒ Spanish	59.35±0.95	52.92±1.25	67.57±1.26	11.32±0.08	8.65±0.00	17.02±0.96	56.37±0.94	49.14±1.24	66.14±1.22
German ⇒ Spanish	59.35±0.92	52.92±1.24	67.57±1.27	11.11±0.07	8.43±0.00	17.00±0.97	55.32±0.85	48.16±1.05	65.07±1.21
Italian ⇒ Spanish	59.35±0.94	52.92±1.30	67.57±1.22	11.17±0.07	8.48±0.00	17.02±0.97	55.27±1.02	47.51±1.30	66.15±1.24
	PL			SCT			SSAN		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
English ⇒ Spanish	42.85±0.81	34.56±1.02	57.86±1.29	60.99±0.92	54.70±1.25	68.96±1.24	58.28±0.94	51.99±1.21	66.36±1.24
French ⇒ Spanish	42.85±0.81	34.56±1.01	57.86±1.29	61.03±0.94	**54.94±1.32**	68.67±1.20	57.59±0.91	51.04±1.24	66.07±1.21
German ⇒ Spanish	42.85±0.78	34.56±0.98	57.86±1.28	61.58±0.96	**55.49±1.33**	69.21±1.22	58.65±0.97	52.77±1.29	66.04±1.29
Italian ⇒ Spanish	42.85±0.82	34.56±0.98	57.86±1.30	61.07±0.95	54.69±1.31	69.17±1.18	58.47±0.92	52.15±1.25	66.58±1.20
	STN			SL			OSHeDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
English ⇒ Spanish	59.56±0.97	53.26±1.33	67.59±1.20	58.53±0.96	52.14±1.32	66.74±1.19	**65.85±0.90**	**55.09±1.25**	**81.97±1.07**
French ⇒ Spanish	59.19±0.98	52.95±1.29	67.13±1.27	58.53±0.95	52.14±1.28	66.74±1.22	**65.34±0.97**	54.52±1.21	**81.64±0.85**
German ⇒ Spanish	59.12±0.96	52.70±1.28	67.34±1.28	58.53±0.98	52.14±1.38	66.74±1.19	**65.54±0.88**	54.63±1.17	**82.22±0.94**
Italian ⇒ Spanish	58.95±0.95	52.72±1.32	66.89±1.19	58.53±0.96	52.14±1.31	66.74±1.25	**64.81±0.91**	**53.63±1.19**	**82.07±0.98**

Table 16: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in NUSWIDE & ImageNet dataset.

NUSWIDE & ImageNet									
	DS3L			KPG			OPDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Text ⇒ Image	67.61±1.65	66.17±2.22	69.20±2.30	55.18±1.17	52.60±0.00	58.10±2.45	71.06±1.44	66.60±1.98	76.38±2.09
	PL			SCT			SSAN		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Text ⇒ Image	42.43±0.26	34.05±0.00	61.15±2.10	70.42±1.49	68.00±2.20	73.10±1.99	67.98±1.49	66.25±2.04	69.85±2.21
	STN			SL			OSHeDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Text ⇒ Image	67.75±1.23	64.80±1.42	71.08±2.16	69.41±1.64	66.62±2.26	72.57±2.23	**80.01±1.30**	**74.65±2.01**	**86.35±0.81**

Table 17: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in Wikipedia dataset.

Wikipedia									
	DS3L			KPG			OPDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Image ⇒ Text	35.90±2.42	26.96±2.70	54.68±3.09	25.37±0.40	16.80±0.00	51.92±3.19	39.60±2.46	29.48±2.60	**61.16±3.26**
Text ⇒ Image	76.10±1.61	74.48±2.02	77.80±2.51	24.27±0.44	16.00±0.00	53.12±3.16	65.72±1.38	62.40±1.03	69.68±2.98
	PL			SCT			SSAN		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Image ⇒ Text	27.11±2.26	18.28±2.10	54.00±3.18	38.51±2.45	29.28±2.77	56.56±3.00	39.52±2.03	30.16±2.18	57.52±3.10
Text ⇒ Image	56.64±1.04	52.00±0.85	62.80±3.02	78.31±1.58	76.44±2.00	80.28±2.47	77.23±1.48	75.36±1.73	79.20±2.50
	STN			SL			OSHeDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Image ⇒ Text	39.96±2.11	30.24±2.21	59.20±3.16	37.80±2.48	28.68±2.70	55.68±3.07	**40.27±2.40**	**31.60±2.79**	56.68±3.22
Text ⇒ Image	75.54±1.70	72.56±2.05	78.80±2.79	76.40±1.47	74.52±1.68	78.40±2.46	**85.93±1.39**	**82.92±2.11**	**89.40±1.51**

Table 18: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in PTB-XL dataset.

PTB-XL									
	DS3L			KPG			OPDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Digital ECG ⇒ Paper ECG	30.30±1.19	34.95±0.58	26.74±1.82	N/A	N/A	N/A	31.47±1.22	36.35±0.63	27.74±1.86
	PL			SCT			SSAN		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Digital ECG ⇒ Paper ECG	26.18±1.37	36.43±0.55	20.43±1.66	26.23±1.65	**46.48±0.71**	18.27±1.60	25.16±1.47	40.40±0.65	18.27±1.54
	STN			SL			OSHeDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Digital ECG ⇒ Paper ECG	20.64±0.96	22.23±0.26	19.27±1.65	25.74±1.55	44.50±0.76	18.11±1.52	**47.48±1.25**	44.30±1.39	**51.16±1.86**

Table 19: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in ImageCLEF-DA dataset.

ImageCLEF-DA	DS3L			KPG			OPDA		
	HOS	OS^{*}	UNK	HOS	OS^{*}	UNK	HOS	OS^{*}	UNK
Bing (Reset-50) ⇒ Caltech (VGG-19)	70.23±1.83	66.23±2.70	74.77±2.45	30.67±0.63	25.73±0.00	38.26±2.74	64.88±1.58	58.83±1.88	72.58±2.60
Bing (Reset-50) ⇒ ImageNet (VGG-19)	62.00±2.01	56.90±2.66	68.19±2.92	35.41±0.91	28.57±0.00	47.26±2.80	57.22±1.72	49.68±2.15	67.64±2.76
Bing (Reset-50) ⇒ PascalVOC (VGG-19)	57.21±1.86	52.42±2.34	63.01±2.84	31.10±0.49	24.74±0.00	42.55±2.93	54.47±1.75	48.13±2.15	62.84±2.92
Bing (VGG-19) ⇒ Caltech (Reset-50)	72.67±1.71	68.03±2.55	78.12±2.23	25.93±0.95	22.29±0.00	31.16±2.59	63.25±1.89	56.22±2.62	72.45±2.76
Bing (VGG-19) ⇒ ImageNet (Reset-50)	64.53±1.90	59.87±2.68	70.00±2.55	36.97±0.85	30.33±0.00	47.41±2.90	55.67±1.76	47.57±2.14	67.28±2.68
Bing (VGG-19) ⇒ PascalVOC (Reset-50)	57.80±2.06	52.40±2.82	64.77±2.95	33.20±1.11	27.76±0.00	41.95±3.01	53.48±1.86	45.24±2.37	66.06±2.75
Caltech (Reset-50) ⇒ Bing (VGG-19)	42.91±2.29	34.04±2.82	59.09±2.89	36.15±0.75	27.81±0.00	52.35±3.10	38.29±2.23	28.41±2.35	59.58±2.90
Caltech (Reset-50) ⇒ ImageNet (VGG-19)	63.21±1.89	58.17±2.57	69.24±2.72	60.98±1.14	55.06±0.00	68.37±2.81	58.15±1.59	49.52±1.98	70.66±2.60
Caltech (Reset-50) ⇒ PascalVOC (VGG-19)	57.99±1.99	53.49±2.76	63.37±2.90	50.82±1.21	45.66±0.00	57.34±2.92	54.29±1.75	47.28±2.15	63.90±2.85
Caltech (VGG-19) ⇒ Bing (Reset-50)	43.83±2.38	34.97±2.77	59.00±2.88	39.75±0.76	31.25±0.00	54.81±2.96	36.94±2.16	27.29±1.98	57.89±2.96
Caltech (VGG-19) ⇒ ImageNet (Reset-50)	61.00±2.16	55.16±2.86	68.44±2.78	60.80±1.15	54.94±0.00	68.13±2.79	52.45±2.00	43.71±2.08	66.12±2.67
Caltech (VGG-19) ⇒ PascalVOC (Reset-50)	56.73±2.06	51.97±2.83	62.67±2.85	54.46±1.19	49.96±0.00	59.96±2.79	52.02±1.79	45.04±2.17	61.81±3.05
ImageNet (Reset-50) ⇒ Bing (VGG-19)	42.94±2.23	34.30±2.64	57.93±2.97	31.59±0.57	23.22±0.00	49.72±2.94	41.76±2.28	32.22±2.47	59.61±2.80
ImageNet (Reset-50) ⇒ Caltech (VGG-19)	66.64±2.00	62.64±2.87	71.25±2.71	40.26±1.13	34.71±0.00	48.99±2.94	65.01±1.76	58.64±2.36	73.17±2.56
ImageNet (Reset-50) ⇒ PascalVOC (VGG-19)	56.71±1.98	51.21±2.77	63.79±2.82	42.87±1.15	37.32±0.00	50.57±3.04	53.19±1.85	46.02±2.40	63.62±2.76
ImageNet (VGG-19) ⇒ Bing (Reset-50)	45.41±2.26	36.84±2.47	59.76±3.05	30.35±0.74	22.36±0.00	48.82±2.88	38.51±2.25	28.99±2.18	58.17±2.88
ImageNet (VGG-19) ⇒ Caltech (Reset-50)	73.97±1.96	69.74±2.79	78.77±2.50	42.12±1.31	38.33±0.00	46.79±3.18	65.33±1.62	58.91±1.87	73.65±2.75
ImageNet (VGG-19) ⇒ PascalVOC (Reset-50)	59.28±1.96	54.79±2.66	64.66±2.90	45.17±1.15	40.38±0.00	51.41±2.95	52.73±1.58	45.01±1.70	63.90±2.88
PascalVOC (Reset-50) ⇒ Bing (VGG-19)	43.02±2.16	34.05±2.63	59.16±2.89	30.05±0.53	22.05±0.00	47.96±2.88	41.07±2.01	31.28±2.09	60.25±2.89
PascalVOC (Reset-50) ⇒ Caltech (VGG-19)	70.64±1.94	66.84±2.71	74.95±2.64	41.55±1.17	36.89±0.00	47.67±2.91	67.29±1.86	61.26±2.45	74.95±2.50
PascalVOC (Reset-50) ⇒ ImageNet (VGG-19)	58.36±1.98	53.49±2.62	64.31±2.88	49.54±1.06	42.80±0.00	58.96±2.73	57.47±1.73	48.60±2.10	70.76±2.68
PascalVOC (VGG-19) ⇒ Bing (Reset-50)	43.67±2.23	34.93±2.71	58.58±2.94	29.42±0.51	21.29±0.00	47.92±2.97	36.78±1.80	27.52±1.80	55.61±2.88
PascalVOC (VGG-19) ⇒ Caltech (Reset-50)	72.40±1.92	68.02±2.62	77.47±2.50	45.18±1.09	41.64±0.00	49.46±3.01	61.53±1.45	54.89±1.78	70.22±2.49
PascalVOC (VGG-19) ⇒ ImageNet (Reset-50)	63.79±2.09	58.40±2.96	70.41±2.70	51.87±0.93	45.20±0.00	61.12±2.97	54.17±1.60	46.41±1.97	65.65±2.62

	PL			SCT			SSAN		
	HOS	OS^{*}	UNK	HOS	OS^{*}	UNK	HOS	OS^{*}	UNK
Bing (Reset-50) ⇒ Caltech (VGG-19)	48.18±1.62	42.69±1.76	57.98±2.90	71.56±1.80	67.63±2.62	75.99±2.41	73.40±1.82	69.14±2.65	78.26±2.39
Bing (Reset-50) ⇒ ImageNet (VGG-19)	38.03±1.19	30.41±1.11	52.64±2.90	62.26±1.84	57.55±2.47	67.85±2.78	62.68±1.90	57.53±2.57	68.92±2.65
Bing (Reset-50) ⇒ PascalVOC (VGG-19)	38.83±1.77	31.11±1.82	52.27±2.92	57.59±1.88	52.47±2.47	63.90±2.84	59.88±1.93	**55.69±2.59**	64.79±2.94
Bing (VGG-19) ⇒ Caltech (Reset-50)	48.61±1.56	42.54±1.77	60.43±2.89	72.41±1.77	68.19±2.58	77.26±2.29	70.23±1.66	66.07±2.48	75.02±2.21
Bing (VGG-19) ⇒ ImageNet (Reset-50)	39.29±1.51	32.53±1.53	52.11±2.99	64.56±1.95	60.21±2.79	69.59±2.46	63.48±1.88	58.73±2.73	69.12±2.40
Bing (VGG-19) ⇒ PascalVOC (Reset-50)	36.41±1.64	28.98±1.81	52.56±3.04	55.77±2.13	50.69±2.96	62.56±2.86	57.16±2.08	52.65±2.96	62.89±2.80
Caltech (Reset-50) ⇒ Bing (VGG-19)	28.85±1.71	20.19±1.71	52.39±2.84	43.52±2.25	35.02±2.71	58.60±2.94	43.57±2.15	35.25±2.64	58.14±2.92
Caltech (Reset-50) ⇒ ImageNet (VGG-19)	35.80±1.36	28.53±1.38	50.07±2.92	63.68±2.09	58.62±2.87	69.72±2.80	63.73±2.03	58.75±2.94	69.65±2.62
Caltech (Reset-50) ⇒ PascalVOC (VGG-19)	41.41±1.50	34.49±1.35	53.76±2.99	58.39±2.00	53.63±2.83	64.11±2.78	59.14±2.08	55.00±2.86	64.01±2.94
Caltech (VGG-19) ⇒ Bing (Reset-50)	28.18±2.28	19.46±2.09	53.70±2.83	43.51±2.26	34.82±2.45	58.24±2.86	43.93±2.34	35.49±2.66	57.79±2.81
Caltech (VGG-19) ⇒ ImageNet (Reset-50)	41.85±1.57	34.43±1.72	56.84±2.91	60.68±2.00	55.16±2.54	67.62±2.78	59.69±1.99	54.46±2.87	66.26±2.58
Caltech (VGG-19) ⇒ PascalVOC (Reset-50)	43.02±1.54	36.20±1.69	54.04±2.98	56.00±1.99	50.98±2.61	62.20±2.93	56.24±2.13	52.38±2.90	61.05±3.01
ImageNet (Reset-50) ⇒ Bing (VGG-19)	34.91±2.10	25.45±2.08	56.04±2.97	43.40±2.34	34.67±2.74	58.39±2.90	42.34±2.31	34.10±2.65	56.04±2.88
ImageNet (Reset-50) ⇒ Caltech (VGG-19)	50.76±1.70	45.62±1.88	57.39±3.04	68.57±1.90	64.30±2.62	73.48±2.76	69.17±1.99	65.20±2.86	73.76±2.58
ImageNet (Reset-50) ⇒ PascalVOC (VGG-19)	41.64±1.75	34.00±1.92	54.29±2.94	55.81±2.11	50.62±2.95	62.38±2.74	57.32±2.01	52.58±2.78	63.09±2.85
ImageNet (VGG-19) ⇒ Bing (Reset-50)	29.13±1.21	20.48±0.96	52.32±2.92	45.51±2.30	36.83±2.58	60.14±2.88	45.26±2.29	36.81±2.62	59.38±2.99
ImageNet (VGG-19) ⇒ Caltech (Reset-50)	48.44±1.50	43.63±1.56	54.91±2.98	74.62±1.94	70.45±2.72	79.31±2.53	71.50±1.89	67.69±2.76	75.78±2.61
ImageNet (VGG-19) ⇒ PascalVOC (Reset-50)	33.69±1.66	26.41±1.80	49.60±2.91	58.37±2.11	53.99±2.79	63.57±2.97	58.01±2.00	54.49±2.78	62.06±2.94
PascalVOC (Reset-50) ⇒ Bing (VGG-19)	30.27±1.53	21.65±1.20	51.93±2.93	43.19±2.26	34.11±2.68	59.47±2.77	42.71±2.32	33.96±2.65	57.93±2.83
PascalVOC (Reset-50) ⇒ Caltech (VGG-19)	51.12±1.57	46.25±1.77	57.39±2.89	71.63±2.04	68.24±2.86	75.37±2.81	71.85±1.99	67.97±2.78	76.34±2.57
PascalVOC (Reset-50) ⇒ ImageNet (VGG-19)	40.68±1.62	32.62±1.68	56.28±2.71	58.99±1.91	53.96±2.67	65.17±2.89	58.94±1.92	53.51±2.50	65.90±2.95
PascalVOC (VGG-19) ⇒ Bing (Reset-50)	27.81±2.03	19.12±1.82	53.01±3.02	43.25±2.35	34.68±2.84	57.92±2.95	43.49±2.26	35.16±2.83	57.44±2.81
PascalVOC (VGG-19) ⇒ Caltech (Reset-50)	47.14±1.45	40.53±1.53	58.59±2.73	72.75±1.84	68.53±2.62	77.58±2.53	70.04±2.00	65.85±2.82	74.84±2.69
PascalVOC (VGG-19) ⇒ ImageNet (Reset-50)	36.78±1.60	29.01±1.79	53.67±2.81	64.23±2.17	58.91±2.96	70.68±2.80	62.91±2.11	57.97±2.87	68.88±2.82

	STN			SL			OSHeDA		
	HOS	OS^{*}	UNK	HOS	OS^{*}	UNK	HOS	OS^{*}	UNK
Bing (Reset-50) ⇒ Caltech (VGG-19)	71.09±1.87	67.14±2.64	75.54±2.63	71.47±1.86	67.43±2.69	76.06±2.48	**76.55±1.58**	**72.18±2.28**	**81.67±2.08**
Bing (Reset-50) ⇒ ImageNet (VGG-19)	60.25±1.94	53.43±2.63	69.27±2.61	61.65±1.96	56.74±2.62	67.60±2.83	**70.18±1.96**	**62.60±2.80**	**80.07±2.37**
Bing (Reset-50) ⇒ PascalVOC (VGG-19)	57.23±2.14	51.56±2.91	64.82±2.84	57.79±1.92	53.10±2.49	63.44±2.87	**60.84±2.00**	53.07±2.56	**71.42±2.90**
Bing (VGG-19) ⇒ Caltech (Reset-50)	66.76±1.93	61.38±2.77	73.43±2.44	71.99±1.77	67.20±2.59	77.65±2.22	**76.36±1.71**	**69.83±2.59**	**84.40±1.70**
Bing (VGG-19) ⇒ ImageNet (Reset-50)	62.89±2.03	57.80±2.88	68.98±2.81	64.71±1.95	60.10±2.79	70.10±2.52	**70.84±1.89**	**65.64±2.74**	**77.18±2.19**
Bing (VGG-19) ⇒ PascalVOC (Reset-50)	55.04±2.00	49.74±2.81	61.91±2.66	56.71±2.03	51.47±2.86	63.57±2.87	**60.95±2.01**	**52.66±2.79**	**73.43±2.38**
Caltech (Reset-50) ⇒ Bing (VGG-19)	40.32±2.39	31.43±2.64	56.98±2.95	43.12±2.23	34.25±2.76	59.23±2.91	**45.55±2.46**	**35.61±2.66**	**63.93±2.80**
Caltech (Reset-50) ⇒ ImageNet (VGG-19)	61.17±1.98	54.71±2.82	69.48±2.69	63.61±2.00	58.64±2.72	69.51±2.68	**71.75±2.03**	**64.46±2.89**	**81.18±2.14**
Caltech (Reset-50) ⇒ PascalVOC (VGG-19)	55.64±2.11	49.19±2.91	64.36±2.86	57.44±1.95	52.85±2.60	62.94±2.84	**62.36±1.97**	**55.16±2.77**	**71.84±2.65**
Caltech (VGG-19) ⇒ Bing (Reset-50)	41.09±2.49	31.89±2.86	57.85±2.79	43.90±2.42	34.85±2.76	59.41±2.98	**48.94±2.38**	**39.74±2.85**	**63.94±2.89**
Caltech (VGG-19) ⇒ ImageNet (Reset-50)	58.46±1.92	52.73±2.56	65.71±2.75	61.59±2.01	56.07±2.73	68.40±2.72	**68.65±2.00**	**62.78±2.84**	**76.67±2.07**
Caltech (VGG-19) ⇒ PascalVOC (Reset-50)	52.98±1.99	47.28±2.61	60.58±3.07	56.82±1.98	51.98±2.67	62.89±2.93	**61.71±2.06**	**54.25±2.83**	**72.27±2.95**
ImageNet (Reset-50) ⇒ Bing (VGG-19)	41.13±2.08	32.05±2.22	57.86±2.89	43.07±2.20	34.35±2.56	58.14±3.12	**46.31±2.22**	**36.58±2.81**	**64.39±2.99**
ImageNet (Reset-50) ⇒ Caltech (VGG-19)	66.59±2.01	61.96±2.91	72.09±2.73	67.09±2.10	63.08±2.96	71.71±2.79	**71.80±2.00**	**68.39±2.83**	**76.03±2.75**
ImageNet (Reset-50) ⇒ PascalVOC (VGG-19)	56.95±2.06	50.42±2.86	65.85±2.77	56.56±2.00	51.33±2.78	63.26±2.82	**60.55±2.00**	**53.17±2.80**	**70.78±2.53**
ImageNet (VGG-19) ⇒ Bing (Reset-50)	40.85±2.43	31.87±2.70	57.16±2.88	45.53±2.30	36.84±2.65	60.14±2.88	**50.23±2.22**	**39.96±2.53**	**68.03±2.92**
ImageNet (VGG-19) ⇒ Caltech (Reset-50)	69.29±1.74	64.26±2.35	75.31±2.52	74.77±1.89	70.60±2.71	79.49±2.42	**79.93±1.86**	**73.53±2.69**	**87.62±1.85**
ImageNet (VGG-19) ⇒ PascalVOC (Reset-50)	52.75±1.95	47.10±2.41	60.11±2.92	58.68±2.06	53.82±2.77	64.62±2.79	**65.44±2.02**	**57.92±2.65**	**75.52±2.71**
PascalVOC (Reset-50) ⇒ Bing (VGG-19)	40.19±2.20	30.51±2.46	59.58±2.90	43.03±2.29	33.91±2.63	59.68±2.86	**46.50±2.38**	**36.76±2.76**	**63.61±2.92**
PascalVOC (Reset-50) ⇒ Caltech (VGG-19)	72.63±1.88	68.33±2.77	77.56±2.37	69.90±1.99	66.21±2.78	74.04±2.64	**78.18±1.85**	**72.63±2.70**	**85.02±2.39**
PascalVOC (Reset-50) ⇒ ImageNet (VGG-19)	56.07±1.98	49.15±2.77	65.63±2.77	57.55±1.92	52.57±2.53	63.68±2.80	**64.81±2.06**	**56.21±2.65**	**77.33±2.22**
PascalVOC (VGG-19) ⇒ Bing (Reset-50)	41.60±2.33	32.53±2.70	58.44±2.84	44.07±2.34	35.52±2.78	58.30±3.01	**48.24±2.37**	**37.92±2.71**	**66.71±2.73**
PascalVOC (VGG-19) ⇒ Caltech (Reset-50)	66.98±2.02	62.46±2.67	72.35±2.83	71.69±1.89	67.32±2.70	76.75±2.51	**77.00±1.86**	**72.12±2.66**	**82.78±2.42**
PascalVOC (VGG-19) ⇒ ImageNet (Reset-50)	62.10±2.02	56.35±2.63	69.22±2.78	63.46±2.05	57.82±2.76	70.41±2.66	**71.86±2.02**	**65.90±2.92**	**79.49±2.63**

Table 20: Prediction performances of RL-OSHeDA as well as baselines for all domain adaptation tasks in Office & Caltech256 dataset.

Office & Caltech256	DS3L			KPG			OPDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Amazon (DeCAF6) ⇒ Caltech (SURF)	46.08±1.41	37.41±1.60	60.15±2.05	36.51±0.69	30.20±0.00	46.22±2.22	43.26±1.58	33.61±1.87	60.84±2.01
Amazon (DeCAF6) ⇒ DSLR (SURF)	70.73±4.37	61.96±6.58	82.57±4.50	55.39±1.70	46.05±0.00	69.59±5.04	61.74±4.41	50.73±4.98	79.32±4.97
Amazon (DeCAF6) ⇒ Webcam (SURF)	71.64±2.73	67.43±4.05	76.62±3.64	61.25±1.62	56.91±0.00	66.34±3.81	61.79±2.88	53.28±3.61	74.00±4.06
Amazon (SURF) ⇒ Caltech (DeCAF6)	68.29±1.31	65.51±1.83	71.37±1.98	36.27±0.91	34.10±0.00	38.80±2.06	58.80±1.40	52.44±2.00	67.48±1.85
Amazon (SURF) ⇒ DSLR (DeCAF6)	86.97±3.62	83.57±5.92	90.68±3.41	24.18±0.94	16.86±0.00	44.32±5.69	81.76±3.69	75.07±5.53	90.00±3.28
Amazon (SURF) ⇒ Webcam (DeCAF6)	83.31±1.92	81.56±2.76	85.17±2.58	29.59±1.34	24.79±0.00	36.97±3.90	76.18±2.55	70.34±3.91	83.59±2.91
Caltech (DeCAF6) ⇒ Amazon (SURF)	59.62±1.55	54.12±2.14	66.47±2.16	42.35±0.87	36.87±0.00	49.89±2.27	53.67±1.48	46.16±1.95	64.31±2.24
Caltech (DeCAF6) ⇒ DSLR (SURF)	70.73±4.53	61.96±6.67	82.57±4.46	37.67±1.36	28.80±0.00	54.86±5.40	61.74±4.48	50.73±5.07	79.32±5.22
Caltech (DeCAF6) ⇒ Webcam (SURF)	71.64±2.92	67.43±4.31	76.62±3.72	44.51±1.41	39.21±0.00	51.52±4.14	61.79±2.88	53.28±3.61	74.00±4.20
Caltech (SURF) ⇒ Amazon (DeCAF6)	83.06±1.13	82.42±1.58	83.72±1.59	11.18±0.34	8.48±0.00	16.55±1.69	78.11±1.16	74.73±1.72	81.85±1.58
Caltech (SURF) ⇒ DSLR (DeCAF6)	86.97±3.55	83.57±5.96	90.68±3.48	3.93±0.02	2.20±0.00	30.95±5.32	81.76±3.67	75.07±5.49	90.00±3.16
Caltech (SURF) ⇒ Webcam (DeCAF6)	83.31±1.96	81.56±2.87	85.17±2.68	8.19±0.50	5.21±0.00	22.34±3.45	76.18±2.50	70.34±3.83	83.59±2.93
Webcam (DeCAF6) ⇒ Amazon (SURF)	59.62±1.54	54.12±2.11	66.47±2.23	55.55±0.93	50.11±0.00	62.37±2.24	53.67±1.54	46.16±2.03	64.31±2.23
Webcam (DeCAF6) ⇒ Caltech (SURF)	46.08±1.44	37.41±1.61	60.15±2.09	37.74±0.67	31.18±0.00	47.86±2.25	43.26±1.53	33.61±1.80	60.84±2.14
Webcam (DeCAF6) ⇒ DSLR (SURF)	70.73±4.54	61.96±6.79	82.57±4.39	57.27±1.65	47.80±0.00	71.49±4.83	61.74±4.44	50.73±5.08	79.32±5.20
Webcam (SURF) ⇒ Amazon (DeCAF6)	83.06±1.13	82.42±1.60	83.72±1.64	24.29±0.95	21.34±0.00	28.28±2.22	78.11±1.17	74.73±1.76	81.85±1.57
Webcam (SURF) ⇒ Caltech (DeCAF6)	68.29±1.37	65.51±1.86	71.37±2.04	33.67±0.94	31.11±0.00	36.74±2.18	58.80±1.47	52.44±2.08	67.48±1.87
Webcam (SURF) ⇒ DSLR (DeCAF6)	86.97±3.64	83.57±5.95	90.68±3.32	20.74±0.94	14.02±0.00	40.95±5.73	81.76±3.57	75.07±5.45	90.00±3.16

	PL			SCT			SSAN		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Amazon (DeCAF6) ⇒ Caltech (SURF)	28.23±1.18	19.50±1.18	53.47±2.21	**47.81±1.35**	38.66±1.55	**62.75±2.11**	43.85±1.73	35.11±1.91	58.89±2.13
Amazon (DeCAF6) ⇒ DSLR (SURF)	35.22±0.51	23.69±0.16	72.84±5.04	**72.68±4.36**	**64.10±6.65**	84.05±4.40	66.44±4.69	57.19±6.61	79.59±4.94
Amazon (DeCAF6) ⇒ Webcam (SURF)	32.23±2.14	22.87±2.21	58.62±4.18	75.46±2.54	71.00±3.65	80.55±3.28	70.98±2.78	65.72±4.19	77.31±3.47
Amazon (SURF) ⇒ Caltech (DeCAF6)	54.21±1.05	46.73±1.21	66.66±1.79	70.29±1.39	67.05±1.94	73.89±1.93	71.94±1.33	68.67±1.78	75.59±2.03
Amazon (SURF) ⇒ DSLR (DeCAF6)	67.80±1.78	58.16±0.00	82.70±4.70	93.35±2.26	91.40±3.62	**95.41±2.34**	90.18±2.45	87.74±3.70	92.84±2.94
Amazon (SURF) ⇒ Webcam (DeCAF6)	66.08±2.23	60.34±2.59	73.38±3.77	88.13±1.60	85.38±2.29	91.17±2.19	86.68±2.14	83.47±3.14	90.28±2.88
Caltech (DeCAF6) ⇒ Amazon (SURF)	36.43±1.27	27.67±1.35	55.15±2.45	63.17±1.51	57.01±2.17	70.90±2.14	59.90±1.49	55.04±2.02	65.82±2.17
Caltech (DeCAF6) ⇒ DSLR (SURF)	35.22±0.47	23.69±0.00	72.84±4.99	**72.83±4.44**	64.86±6.74	**83.11±4.35**	65.39±4.32	56.48±6.14	77.84±5.10
Caltech (DeCAF6) ⇒ Webcam (SURF)	32.23±2.06	22.87±2.15	58.62±4.00	75.66±2.51	71.82±3.76	80.00±3.60	70.76±2.56	65.35±3.72	77.31±3.34
Caltech (SURF) ⇒ Amazon (DeCAF6)	68.53±0.95	63.58±1.11	75.11±1.61	88.53±1.00	87.22±1.45	89.87±1.34	89.04±1.00	87.75±1.51	90.38±1.26
Caltech (SURF) ⇒ DSLR (DeCAF6)	67.80±1.81	58.16±0.32	82.70±4.74	93.50±2.70	91.93±4.64	**95.14±2.32**	91.36±2.59	88.96±4.08	93.92±2.95
Caltech (SURF) ⇒ Webcam (DeCAF6)	66.08±2.21	60.34±2.51	73.38±3.85	87.79±1.67	84.82±2.51	91.03±2.04	87.13±2.22	83.43±3.32	91.31±2.65
Webcam (DeCAF6) ⇒ Amazon (SURF)	36.43±1.24	27.67±1.32	55.15±2.21	62.57±1.50	56.43±2.06	70.27±2.06	60.23±1.45	55.25±2.03	66.28±2.20
Webcam (DeCAF6) ⇒ Caltech (SURF)	28.23±1.11	19.50±1.08	53.47±2.13	**47.38±1.45**	**38.50±1.70**	61.81±2.15	43.79±1.68	34.81±1.89	59.85±2.12
Webcam (DeCAF6) ⇒ DSLR (SURF)	35.22±0.55	23.69±0.22	72.84±5.26	71.98±3.87	63.60±5.95	83.11±3.91	65.60±4.92	56.35±6.71	78.65±5.29
Webcam (SURF) ⇒ Amazon (DeCAF6)	68.53±0.92	63.58±1.04	75.11±1.60	87.85±1.01	86.45±1.50	89.31±1.37	87.85±1.04	86.85±1.53	88.89±1.39
Webcam (SURF) ⇒ Caltech (DeCAF6)	54.21±1.05	46.73±1.23	66.66±1.82	70.55±1.30	66.91±1.90	74.68±1.85	72.99±1.33	69.26±1.82	77.21±1.87
Webcam (SURF) ⇒ DSLR (DeCAF6)	67.80±1.87	58.16±0.57	82.70±4.81	93.41±2.11	91.78±3.48	95.14±2.31	89.06±2.81	86.35±4.49	92.03±3.17

	STN			SL			OSHeDA		
	HOS	OS^*	UNK	HOS	OS^*	UNK	HOS	OS^*	UNK
Amazon (DeCAF6) ⇒ Caltech (SURF)	47.74±1.61	**39.29±1.97**	60.99±2.23	45.87±1.48	37.12±1.82	60.25±2.22	46.69±1.39	37.78±1.65	61.41±2.32
Amazon (DeCAF6) ⇒ DSLR (SURF)	72.52±4.17	63.01±6.13	**85.54±3.83**	70.36±4.71	62.20±6.92	81.08±4.81	71.88±4.38	63.98±6.39	82.30±4.44
Amazon (DeCAF6) ⇒ Webcam (SURF)	**77.24±2.61**	**72.38±3.91**	82.90±3.20	71.76±2.81	67.75±4.18	76.34±3.68	75.72±2.68	68.42±3.90	**84.97±2.98**
Amazon (SURF) ⇒ Caltech (DeCAF6)	69.07±1.33	65.04±1.92	73.68±1.82	68.20±1.37	66.15±1.93	70.40±1.96	**79.28±1.13**	**72.99±1.74**	**86.85±0.99**
Amazon (SURF) ⇒ DSLR (DeCAF6)	80.02±2.28	74.04±3.82	87.43±2.26	86.88±3.67	84.13±6.05	89.86±3.43	**94.59±2.13**	**95.35±2.31**	94.05±3.31
Amazon (SURF) ⇒ Webcam (DeCAF6)	81.48±2.18	77.27±2.87	86.28±3.32	83.33±2.01	80.97±2.76	85.93±2.74	**92.45±1.30**	**90.28±1.94**	**94.97±1.72**
Caltech (DeCAF6) ⇒ Amazon (SURF)	61.69±1.44	55.02±1.93	70.27±2.10	60.31±1.53	54.92±2.12	66.97±2.15	**66.11±1.53**	**57.86±2.16**	**77.33±1.80**
Caltech (DeCAF6) ⇒ DSLR (SURF)	69.10±4.13	59.61±6.27	82.57±4.10	70.36±4.51	62.20±6.67	81.08±4.82	72.05±4.18	**65.33±5.65**	81.35±5.62
Caltech (DeCAF6) ⇒ Webcam (SURF)	**77.24±2.58**	**72.52±3.98**	82.76±3.14	71.76±2.95	67.75±4.23	76.34±3.78	74.86±2.76	67.65±3.92	**84.14±3.39**
Caltech (SURF) ⇒ Amazon (DeCAF6)	85.54±1.12	83.07±1.61	88.26±1.46	83.90±1.09	83.09±1.57	84.73±1.51	**94.36±0.71**	**91.43±1.13**	**97.50±0.83**
Caltech (SURF) ⇒ DSLR (DeCAF6)	80.56±3.42	75.17±5.36	86.89±3.77	86.88±3.51	84.13±5.87	89.86±3.36	**94.32±2.02**	**96.30±2.63**	92.57±2.91
Caltech (SURF) ⇒ Webcam (DeCAF6)	78.34±2.00	74.47±2.27	82.76±3.37	83.33±1.99	80.97±2.79	85.93±2.74	**92.00±1.24**	**88.87±1.77**	**95.79±1.71**
Webcam (DeCAF6) ⇒ Amazon (SURF)	61.26±1.48	54.27±1.91	70.40±2.17	60.31±1.53	54.92±2.18	66.97±2.16	**66.00±1.51**	**57.20±2.12**	**78.11±1.84**
Webcam (DeCAF6) ⇒ Caltech (SURF)	46.97±1.59	38.19±1.94	61.11±2.17	45.87±1.53	37.12±1.87	60.25±2.24	46.72±1.41	37.53±1.73	**62.21±2.15**
Webcam (DeCAF6) ⇒ DSLR (SURF)	72.18±4.16	63.57±6.06	**83.65±4.24**	70.36±4.88	62.20±7.04	81.08±4.82	**71.91±4.48**	**63.68±6.80**	82.70±4.63
Webcam (SURF) ⇒ Amazon (DeCAF6)	86.87±1.07	84.97±1.55	88.87±1.43	83.90±1.11	83.09±1.58	84.73±1.50	**94.44±0.75**	**91.42±1.27**	**97.69±0.65**
Webcam (SURF) ⇒ Caltech (DeCAF6)	69.43±1.34	65.14±1.84	74.41±1.99	68.20±1.38	66.15±1.87	70.40±1.95	**79.49±1.18**	**73.01±1.81**	**87.37±1.01**
Webcam (SURF) ⇒ DSLR (DeCAF6)	83.46±2.50	79.20±3.97	88.38±2.92	86.88±3.65	84.13±5.95	89.86±3.49	**94.36±2.21**	**95.68±3.54**	**93.24±2.70**

Table 21: Pairwise p-values from the Nemenyi test conducted on 56 domain adaptation tasks. P-values less than 0.05 indicate a statistically significant difference in prediction performance between the two corresponding methods.

	DS3L	KPG	OPDA	PL	SCT	SSAN	STN	SL	RL-OSHeDA
DS3L	1.000	0.001	0.001	0.001	0.525	0.900	0.900	0.900	**0.001**
KPG	0.001	1.000	0.283	0.900	0.001	0.001	0.001	0.001	**0.001**
OPDA	0.001	0.283	1.000	0.050	0.001	0.001	0.001	0.001	**0.001**
PL	0.001	0.900	0.050	1.000	0.001	0.001	0.001	0.001	**0.001**
SCT	0.525	0.001	0.001	0.001	1.000	0.589	0.062	0.525	**0.009**
SSAN	0.900	0.001	0.001	0.001	0.589	1.000	0.900	0.900	**0.001**
STN	0.900	0.001	0.001	0.001	0.062	0.900	1.000	0.900	**0.001**
SL	0.900	0.001	0.001	0.001	0.525	0.900	0.900	1.000	**0.001**
RL-OSHeDA	**0.001**	**0.001**	**0.001**	**0.001**	**0.009**	**0.001**	**0.001**	**0.001**	**1.000**