On the Value of Label and Semantic Information in Domain Generalization⋆,⋆⋆

Fan Zhou[a], Shichun Yang[b], Boyu Wang[c,d] and Brahim Chaib-draa[a]

[a]Department of Computer Science, Laval University, 2325 rue de l'universite, Quebec, G1V 0A6, Canada
[b]Department of Automotive Engineering, Beihang University, No.37 Xueyuan Road, Haidian District, 100191, Beijing, China
[c]Department of Computer Science, Western University, 1151 Richmond St, London, N6A 3K7, Ontario, Canada
[d]Vector Institute, Toronto, 661 University Ave Suite 710, M5G 1M1, Ontario, Canada

ARTICLE INFO

Keywords:
Domain Generalization
Conditional Matching
Label and Semantic Information

ABSTRACT

In this work, we tackle the domain generalization (DG) problems aiming to learn a universal predictor on several source domains and deploy it on an unseen target domain. Many existing DG approaches were mainly motivated by the domain adaptation techniques, which only aligned the marginal feature distribution but ignored conditional relations and label information in the source domains. Although some recent advances started to take advantage of conditional semantic distributions, theoretical justifications were still missing. To this end, we investigate the theoretical guarantee for a successful generalization process by focusing on how to control the target domain error. Our results reveal that to control the target risk, one should jointly control the source errors that are weighted according to label information and align the semantic conditional distributions between different source domains. The theoretical analysis then leads to an efficient algorithm to control the label distributions as well as match the semantic conditional distributions. To verify the effectiveness of our method, we evaluate it against recent baseline algorithms on several benchmarks. Empirical results show that the proposed method outperforms most of the baseline methods and shows state-of-the-art performances.

1. Introduction

Recent machine learning and deep learning progresses usually depend on a large amount of labelled data, which is expensive to annotate. To alleviate this issue, many transfer learning (Maurer et al., 2013) related approaches, *e.g.*, multi-task learning (MTL) (Long et al., 2015; Zhou et al., 2021c,a), domain adaptation (DA) (Ben-David et al., 2010; Wen et al., 2019; Guan et al., 2021; Achituve et al., 2021; Fang et al., 2020) and domain generalization (DG) (Dou et al., 2019; Matsuura and Harada, 2020; Zhou et al., 2021e; Zhao et al., 2020; Zhou et al., 2021d), have been proposed to take advantage of shared knowledge from different but related data sources. The key idea behind these transfer learning-related methods is to discover transferable feature representations that generalize well to new domains.

Most existing DA and MTL approaches have been devoted to adopt discrepancy metric minimization (Li et al., 2017), statistic distance minimization (Long et al., 2015, 2017) or adversarial training (Ganin et al., 2016; Shen et al., 2018a) methods to learn the transferable features (Li et al., 2020; Shui et al., 2019; Mao et al., 2020), which only control the marginal feature distributions. In addition, in the context of DA, the target data are usually partially available during training. However, we cannot always expect such a setting holds in practice. For example, considering an autonomous driving system, the training and deploying environments could differ from each other, and the model would not be able to expect to access the deploying (target) data during training. To this end, we tackle the DG problem, which aims to extract the knowledge from source domains that generalizes well to an unseen target (test) domain. We illustrate a general workflow of DG in Fig. 1.

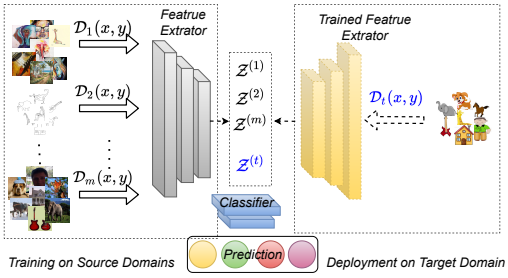


Figure 1: General workflow domain generalization: The model is trained on several source domains $(\mathcal{D}_1, \mathcal{D}_2, \dots \mathcal{D}_m)$ while deployed on an unseen target domain. During the training phase, the source data are mixed as input and feed into the model, during which both the source feature $\mathbb{P}(\mathbf{x})$ and label $\mathbb{P}(y)$ are available to the learner. During the deployment phase, the model is frozen and tested on the target domain $\mathcal{D}_t$, which is inaccessible to the model during the training phase.

Due to the similar problem settings with DA, many DA methodologies, especially the adversarial training (Ganin et al., 2016) based approaches (Li et al., 2017; Dou et al., 2019; Zhou et al., 2021b), were borrowed for DG. However, these approaches only align the feature distribution $\mathbb{P}(\mathbf{x})$ and rely on the theoretical results under the assumption that the combined error between the source and target domain is small (Ben-David et al., 2010), which could not hold in practice. Zhao et al. (2019) showed that conditional

⋆This document is the results of the research project funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and China Scholarship Council.
ORCID(s): 0000-0003-1736-2641 (F. Zhou)

shift problems can degrade the prediction performance. Besides, if we only align the feature distribution $\mathbb{P}(\mathbf{x})$ while ignoring the conditional semantic $\mathbb{P}(\mathbf{x}|y)$ and labelling $\mathbb{P}(y)$ distribution, the class information for each category among different domains can be lost, which leads to indiscriminative features, *a.k.a.* semantic misalignment problem (Dou et al., 2019; Zhou et al., 2021b). As a consequence, the model may suffer from ambiguous classification boundaries (Zhou et al., 2021b), which hinders the generalization performance.

To address this issue, some recent studies (e.g. Dou et al., 2019; Zhou et al., 2021b) have leveraged the label information to explore the semantic relation for the DG. However, the theoretical justifications for the benefits of semantic alignment remain elusive. Existing theoretical results (e.g. Zhao et al., 2020; Li et al., 2018c) only focused on minimizing the conditional distribution divergences from an optimization perspective, while the analysis for the generalization properties are still missing.

In this work, we aim to develop theoretical insights into how to ensure a successful generalization process by investigating the test error on the target domain. Our results reveal the necessity of controlling the semantic conditional distributions as well as the label distribution divergence across all the source domains. The contributions of our work are three-fold:

1. We build a theoretical analysis framework to understand the domain generalization process upon bounding the test error on the target domain with total variation distance, which provides a deeper understanding of the role of semantic alignment for general DG problems.

2. Our analysis also reveals the importance of controlling the label distribution divergence for each domain to minimize the generalization error.

3. On the algorithmic side, our theoretical results inspire a novel DG algorithm that jointly minimizes the source errors as well as semantic distribution matching for all the source domains.

Specifically, we propose to simultaneously match the semantic distributions via minimizing the centroid statistics across distributions and controlling the label distribution losses. We conduct extensive experiments and the results show that the proposed algorithm outperforms various strong baselines, especially when label shift occurs.

2. Related Works

2.1. Domain Adaptation

Domain Adaptation (DA) has been an active research area in recent years. We refer the reader to some existing literature surveys (Wang and Deng, 2018; Redko et al., 2020) to have a comprehensive summary of the recent progress. Specifically, the label shift problem tackled in this work has some connections with the *heterogeneous* domain adaptation (Liu et al., 2020a) and *open set* domain adaptation (Fang et al., 2020) problems. In this context, Liu et al. (2020a) theoretically analyzed the guarantees of the correctness of transferring knowledge together with an angle-based metric to measure the distance between the source and target domains under a heterogeneous DA setting. Latterly Liu et al. (2020b) investigated the multi-source heterogeneous DA problems with a shared-fuzzy-equivalence-relation neural network model. In the case of the open set learning scenario, it will be more challenging to match the features. In this context, some practical approaches have been devoted to the open set learning problems (Liu et al., 2019; Shu et al., 2021), and theoretical justifications (Fang et al., 2021a,b) for the open set domain adaptation problems.

2.2. Domain Generalization

Similar to DA problems, the underlying assumption of DG is that there exists an invariant feature distribution across all the domains, which consequently generalizes well to an unseen domain. From a methodological perspective, existing DG approaches can be categorized into three groups: 1) Distribution matching-based approaches, 2) Episodic training-based approaches, and 3) Data augmentation-based approaches.

The distribution matching methods were mainly motivated by the theoretical results in the DA literature Ben-David et al. (2010); Redko et al. (2017), where the domains were aligned via some distribution matching, distribution distance minimization or adversarial training methods to discover the shared knowledge. For example, maximum mean discrepancy (MMD) was adopted in Li et al. (2018b) as a distribution regularizer together with the adversarial autoencoder (AAE) to learn the invariant features. Muandet et al. (2013) proposed the kernel-based *Domain Invariant Component Analysis* (DICA) algorithm, where a kernel-based optimization algorithm was adopted to learn a domain-invariant transformation by minimizing the dissimilarities between domains. Ghifary et al. (2015) proposed to use adversarial training techniques to extract domain-invariant features under a multi-task learning style setting. Li et al. (2018c) proposed a DG approach by leveraging deep neural networks for domain-invariant representation learning.

Recently, some approaches addressed the DG problem in a meta-learning manner via the episodic training paradigm. The notions of *meta-train* and *meta-test* are used to simulate the distribution shift during each training iteration on the source domain datasets. Specifically, MetaReg Balaji et al. (2018) explored the regularization functions for DG within a learning-to-learn framework. Meta Agnostic Meta-Learning (MAML) Finn et al. (2017) was adopted by Li et al. (2018a) to back-propagate the gradient of the losses of the meta-test tasks Dou et al. (2019) for DG. Du et al. (2020) proposed to model the shared classifier model parameters as a probabilistic meta-learning model. Sharifi-Noghabi et al. (2020) also adopted meta-learning to simulate the domain shift and adopted

an entropy-based loss to give pseudo-labels together with class-level centroids to ensure semantic properties. Gong et al. (2021) introduced an interesting setting where the target domain was assumed as a compound of several unknown domains that were treated as sub-target domains. Then, a meta-learning algorithm was adopted to fuse the sub-target domains together with the MAML algorithm for handling the generalization process.

We also notice that some recent works Zhou et al. (2020b); Mancini (2020); Xu et al. (2021) started to implement some data augmentation methods to generate new instances for training. This kind of work typically relies more on the new data rather than transferring the knowledge, which is somehow out of the problem scope of our work.

In terms of theoretical analysis, Blanchard et al. (2011) firstly proposed the notion of *average risk* for a binary classification problem. Albuquerque et al. (2019) derived the target domain bound using $\mathcal{H}$ divergence by assuming the target domain is within the convex hull of the source domains. However, they both only focused on aligning the feature marginal distributions, ignoring the semantic information in the source domains. More recently, Zhao et al. (2020) proposed the conditional matching algorithm by minimizing the prediction entropy $\mathcal{H}\big(\mathbb{P}(y|x)\big)$ across all the source domains, but the theoretical analysis therein was developed from an optimization perspective, without examining the generalization performance in the target domain. In contrast, our work provides the generalization bounds to understand the DG process, which also motivates an efficient algorithm to control the semantic conditional distributions, which then enables us to design a novel semantic matching algorithm.

2.3. Conditional Matching for DA and DG

Learning and leveraging the semantic conditional distribution $\mathbb{P}(\mathbf{x}|y)$ is an important aspect of machine learning, which has been prevalent in different learning paradigms such as few-shot learning (Motiian et al., 2017; Luo et al., 2017), transfer learning (Long et al., 2014), etc. In the context of DA, Xie et al. (2018) theoretically analyzed the semantic transfer method with pseudo labels using $\mathcal{H}$-divergence Ben-David et al. (2010). Zhang et al. (2019) explored the class-specific prototype semantic feature learning using a symmetric network. In the context of DG, semantic misalignment problems could hinder the generalization performance. Aiming to solve this issue, Dou et al. (2019) adopted the triplet loss as an auxiliary learning objective on top of the meta-learning based DG approach (Li et al., 2018a). Matsuura and Harada (2020) proposed to adopt an unsupervised learning objective to explore the class-level similarities to enhance the semantic separation. Zhou et al. (2021b) adopted the Wasserstein adversarial training (Shen et al., 2018a) to achieve the domain level alignment while exploring the class-level similarities to force the instances from the same class to be close to each other and push the instances from different classes away from each other, *i.e.*, achieving the semantic separation with a metric learning objective (Wang et al., 2019). Even though these works have shown the benefits of considering the semantic conditional distributions, however, the theoretical justifications are still missing. In this work, we provide the first theoretical analysis on the benefits of controlling the semantic conditional distributions and provide a concrete algorithm to jointly minimize the label and semantic distribution divergence.

3. Notations and Preliminaries

We start by introducing some preliminaries with notations and definitions. Then we analyze the importance of leveraging the label and semantic distribution. After that, we show the harm of label and semantic distribution shifts in domain generalization.

3.1. Notations and Definitions

Let $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ be a training example drawn from some unknown distribution $\mathcal{D}$, where $\mathbf{x}$ is the data point, and y is its label. A hypothesis is a function $h \in \mathcal{H}$ that maps $\mathcal{X}$ to the set $\mathcal{Y}'$ sometimes different from $\mathcal{Y}$, where $\mathcal{H}$ is a hypothesis class. For a non-negative loss function $\ell : \mathcal{Y}' \times \mathcal{Y} \mapsto \mathbb{R}_+$, we denote by $\ell(h(\mathbf{x}), y)$ the loss of hypothesis h at $(\mathbf{x}, y)$. Let $S = \{(\mathbf{x}_j, y_j)\}_{j=1}^N$ be a set of N training examples drawn independently from $\mathcal{D}$. The empirical loss of h on S and its generalization loss over $\mathcal{D}$ are defined, respectively, by $\hat{R}(h) = \frac{1}{N}\sum_{j=1}^N \ell(h(\mathbf{x}_j), y_j)$, and $R(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\ell(h(\mathbf{x}), y)$.

In the context of DG, we are given m source tasks $\{S_i\}_{i=1}^m$, where $S_i = \{(\mathbf{x}_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$ is drawn from a distribution $\mathcal{D}_i$. The objective of a DG algorithm is to learn a feature representation that extracts the knowledge that can be shared across all the known source domains so that it can also generalize well to an unseen target domain distribution $\mathcal{D}_t$.

3.2. Distribution Distance Measure

To measure the marginal and conditional distributions, we need a tool to measure the distribution distances, which is crucial in recent domain adaptation or generalization methodologies. In this paper, we adopt the Jensen-Shannon divergence in our analysis, which has been extensively studied in recent literature in transfer learning (Dou et al., 2019; Matsuura and Harada, 2020; Zhao et al., 2019).

Definition 1 (Jensen Shannon (J-S) Divergence Lin (1991))**.** *Let $\mathcal{D}_i(\mathbf{x}, y)$ and $\mathcal{D}_j(\mathbf{x}, y)$ be two distribution over $\mathcal{X}\times\mathcal{Y}$, and let $\mathcal{M} = \frac{1}{2}(\mathcal{D}_i + \mathcal{D}_j)$, then the J-S divergence between $\mathcal{D}_i$ and $\mathcal{D}_j$ is defined as*

$$D_{JS}(\mathcal{D}_i\|\mathcal{D}_j) = \frac{1}{2}[D_{KL}(\mathcal{D}_i\|\mathcal{M}) + D_{KL}(\mathcal{D}_j\|\mathcal{M})] \quad (1)$$

where $D_{KL}(\mathcal{D}_i\|\mathcal{D}_j)$ is the Kullback–Leibler divergence. The square root of Jensen-Shannon Divergence, i.e.*, $\sqrt{D_{JS}}$ is also know as* Jensen-Shannon Distance *Fuglede and Topsoe (2004).*

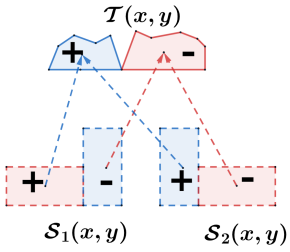


Figure 2: A example in semantic shift. The dashed lines indicate the matching process. Let the feature marginal distribution as the color of the instance while the label distribution as positive (+) or negative (-), *i.e.*, $X = \{\text{red}, \text{blue}\}$, $Y = \{+, -\}$. For the source distribution $\mathcal{S}_1$, $\mathbb{P}_{\mathcal{S}_1}(X = \text{red}|Y = +) = 1$ while for the source distribution $\mathcal{S}_2$, $\mathbb{P}_{\mathcal{S}_2}(X = \text{red}|Y = +) = 0$.

In our analysis, we also consider the *Total Variation Distance*, which is an upper bound of D_{JS}, we provide the definition below.

Definition 2 (Total Variation Distance Lin (1991))**.** *Let $\mathcal{D}_i(\mathbf{x}, y)$ and $\mathcal{D}_j(\mathbf{x}, y)$ be two distribution over $\mathcal{X} \times \mathcal{Y}$, then the total variation distance could be measured by*

$$d_{TV}(\mathcal{D}_i, \mathcal{D}_j) = \frac{1}{2}|\mathcal{D}_i - \mathcal{D}_j| \tag{2}$$

The J-S divergence, J-S distance and TV distance are usually studied together when bounding the distances between different data distributions (Zhou et al., 2021a) since they enjoy the bounding properties (Shui et al., 2022; Polyanskiy and Wu, 2019) that can provide us with good theoretical analysis tools.

3.3. The Value of Label and Semantic Information

In the context of DG, a learner can only access the data from the source domains (seen), while no target data is available during the training phase (unseen). As aforementioned, many DA techniques have been introduced to DG problems due to the similar setting. Early approaches (e.g. Li et al., 2018a,b; Carlucci et al., 2019) usually only focused on aligning the feature distribution $\mathbb{P}(\mathbf{x})$ while ignoring the labeling $\mathbb{P}(y)$ and semantic $\mathbb{P}(\mathbf{x}|y)$ distributions. Some previous work (e.g. Dou et al., 2019; Zhou et al., 2021b) pointed out that only aligning the feature distribution via distribution matching or adversarial training can lead to the semantic misalignment problems (Zhou et al., 2021b,a). Though some recent methods (e.g. Dou et al., 2019; Matsuura and Harada, 2020; Zhou et al., 2021b) start to consider the semantic distribution matching, their theoretical justifications remain elusive. Our work provides a complete framework to understand DG's generalization properties, enabling us to design an efficient semantic conditional matching algorithm.

On the other hand, many of the current DG approaches assumed that the label distribution across all the domains is the same. However, this assumption is not necessarily held in practice. A long-neglected issue is the *label shift* problem, which has been explored in the literature of multi-task learning and domain adaptation Panareda Busto and Gall (2017); Geng et al. (2020); Azizzadenesheli et al. (2019) but missing in domain generalization. More formally, the label shift between two domains $\mathcal{D}_i$ and $\mathcal{D}_j$ indicates $D_{\text{JS}}(\mathcal{D}_i(y), \mathcal{D}_j(y)) \neq 0$ Zhou et al. (2021a).

We present an example to show the necessity of controlling semantic divergence in Fig. 2. Suppose we have two source distributions $\mathcal{S}_1(\mathbf{x}, y)$ and $\mathcal{S}_2(\mathbf{x}, y)$, and hope to match to the target distribution $\mathcal{T}(\mathbf{x}, y)$. The feature marginal distribution is represented by the color of the region while the label distribution is indicated by positive (+) or negative (-), *i.e.*, $X = \{\text{red}, \text{blue}\}$, $Y = \{+, -\}$. For the source distribution $\mathcal{S}_1$, $\mathbb{P}_{\mathcal{S}_1}(X = \text{red}|Y = +) = 1$ while for the source distribution $\mathcal{S}_2$, $\mathbb{P}_{\mathcal{S}_2}(X = \text{red}|Y = +) = 0$. In this case, if we only use the general adversarial training or MMD based approaches to align the marginal distribution, it will be difficult to fix the semantic shift problem. We should also consider to match the semantic distributions for each domain. Another practical example can be the multi-source generalization problems on the digits problems. Let MNIST, which is a grey-scaled digits dataset, be $\mathcal{D}_i$, and let SVHN dataset, which consists of colorful images of street numbers, be $\mathcal{D}_j$. If we consider a specific class $Y = y_k$, we can easily see that $\mathcal{D}_i(\mathbf{x}|y) \neq \mathcal{D}_j(\mathbf{x}|y)$ since the color and digits styles are obviously different from each other.

On the other hand, *label shift* problem may also hurt the generalization performance. For example, for a health diagnostic learning task using DG Liu et al. (2021), when collecting the data from different hospitals, the labels may vary from each other across different datasets. The ultimate goal of DG is to align $\mathbb{P}(\mathbf{x}, y) = \mathbb{P}(\mathbf{x}|y)\mathbb{P}(y)$ between domains, if $\mathbb{P}(y)$ changes, even we can match $\mathbb{P}(\mathbf{x}|y)$ properly for all the domains, the prediction of the classifier can still diverge since the label distribution is not necessarily aligned during either the supervised classification process or the semantic matching process.

All these examples indicate that we need to consider both the label and semantic distribution alignments when designing DG algorithms. In the next section, we develop the theoretical justifications for controlling the conditional semantic and label distributions. Moreover, our results also lead to an efficient algorithm for DG problems.

4. Theoretical Analysis and Methodology

4.1. Theoretical Analysis

One fundamental assumption of DG is that all the domains are not far from each other in terms of distribution distances (Zhou et al., 2021b). More formally, among all the source domains, let $\mathcal{D}^\star$ be the nearest one to the target domain, *i.e.*, $\epsilon^\star \triangleq d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \leq d_{TV}(\mathcal{D}_i, \mathcal{D}_t), \forall i$. Then, it's reasonable to assume that $\epsilon^\star$ is small for DG problems since if the distance between the source and target is arbitrarily large, the learner will fail to generalize to the target domain. Note that the nearest domain $\mathcal{D}^\star$ is not

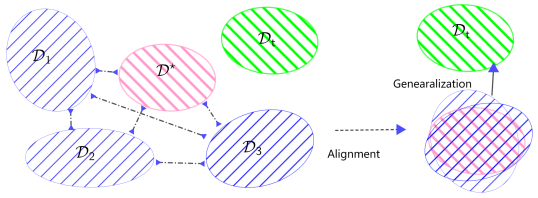


Figure 3: The domain generalization process where there are several source domains and a target domain. In case we have limited number of source domains, there exists a source domain $\mathcal{D}^\star$ that is the nearest to the target domain. At the training phase, we implement the alignment process for all the source domains to learn the transferable features that could be generalized to the target domain.

known yet always exists when we have a finite number of source domains. Later we show that the empirical algorithm can be designed without relying on the nearest domain $\mathcal{D}^\star$. The assumption $\epsilon^\star \triangleq d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \leq d_{TV}(\mathcal{D}_i, \mathcal{D}_t), \forall i$ shares the similar assumption of the common assumption of DG that source domains and target domain come from the same meta-distribution. Then, we can also assume $\mathcal{D}^\star$ and $\mathcal{D}_t$ satisfy a semantic conditional distance, *i.e.*, $d_{TV}\left(\mathcal{D}^\star(\mathbf{x}|y), \mathcal{D}_t(\mathbf{x}|y)\right) \leq \kappa^\star$ where $\kappa^\star$ is a constant that is not arbitrarily large.

We show a generalization process in Fig. 3 where we have several source domains, and the target domain is unseen but assumed not to be far away from the source domains. Then, we can bound the learning risk on the target domain $R_{\mathcal{D}_t}(h)$ as shown in Theorem 1.

Theorem 1. *Suppose we have m source domains $\mathcal{D}_1, \dots \mathcal{D}_m$, and $\mathcal{D}^\star$ is the nearest source domain to the target $\mathcal{D}_t$, and $\epsilon^\star \triangleq d_{TV}(\mathcal{D}^\star, \mathcal{D}_t)$. Then the target domain risk is bounded by,*

$$R_{\mathcal{D}_t}(h) \leq \frac{1}{m}\sum_{i=1}^{m} R_{\mathcal{D}_i}(h) + \epsilon^\star + \frac{1}{m}\sum_i d_{TV}(\mathcal{D}^\star(\mathbf{x}, y), \mathcal{D}_i(\mathbf{x}, y)) \tag{3}$$

Remark: The first term in Eq. 3 is the averaged source error which can be approximated by the empirical risk minimization. The second term is unobservable but is assumed to be small and can be ignored. The third term can also not be estimated directly since we don't know which source domain is the nearest one to the target. However, it can be minimized by pair-wised distribution matching between all source domains.

Theorem 1 bounds the target generalization error in terms of the joint distributions between source domains. To motivate a more concrete DG algorithm that leverages the label ($\mathcal{D}(y)$) and semantic conditional ($\mathcal{D}(\mathbf{x}|y)$) information, we have the following Corollary.

Corollary 1. *Following the assumptions of Theorem 1, then the target domain risk could be bounded by,*

$$\begin{aligned} R_{\mathcal{D}_t}(h) \leq \frac{1}{m}\sum_{i=1}^{m} R_{\mathcal{D}_i}(h) + \epsilon^\star \\ + \frac{1}{m}\sum_i \Big[\underbrace{\sqrt{D_{JS}(\mathcal{D}^\star(y)||\mathcal{D}_i(y))}}_{\mathbf{I}} \\ + \underbrace{\sqrt{\mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{JS}(\mathcal{D}^\star(\mathbf{x}|y)||\mathcal{D}_i(\mathbf{x}|y))}}_{\mathbf{II}} \\ + \underbrace{\sqrt{\mathbb{E}_{y\sim\mathcal{D}_i(y)} D_{JS}(\mathcal{D}^\star(\mathbf{x}|y)||\mathcal{D}_i(\mathbf{x}|y))}}_{\mathbf{III}} \Big] \end{aligned} \tag{4}$$

In order to minimize Eq. 4, except for minimizing the source domain risks $\frac{1}{m}\sum_{i=1}^{m} R_{\mathcal{D}_i}(h)$ we need to consider the last three terms **I**: J-S distance between the label distribution $\mathcal{D}^\star(y)$ and $\mathcal{D}_i(y)$, as well as **II** and **III**, which are the J-S distance between the semantic distributions.

For **I** in Eq. 4, we could adopt a reweighted loss $\hat{\mathcal{L}}^{\boldsymbol{\alpha}}_{\mathcal{D}_i}$ (will be introduced in Eq. 8) to balance the label distribution for each pair of source domains so that $D_{JS}(\mathcal{D}_i(y)||\mathcal{D}_j(y)) = 0$ for all the domain pairs i, j. In this case, term **II** and **III** will be identical to each other and we can bound the generalization risk on the target domain as follows,

Corollary 2. *Following the assumptions of Theorem 1 and assume the semantic distribution between the nearest source domain to the target domain is a constant,* i.e., $d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y = k), \mathcal{D}_t(\mathbf{x})|Y = k) \leq \kappa^\star$. *Let $\hat{\mathcal{L}}^{\boldsymbol{\alpha}}_{\mathcal{D}_i}(h)$ be the reweighted loss and the prediction loss function is bounded by* $[0, 1]$, *then the target domain risk could be bounded by,*

$$\begin{aligned} R_{\mathcal{D}_t}(h) \leq \underbrace{\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{x\sim\mathcal{D}_i} \hat{\mathcal{L}}^{\boldsymbol{\alpha}}_{\mathcal{D}_i}(h)}_{\textit{Re-weighted source risks}} + \underbrace{\kappa^\star}_{\textit{Constant}} \\ + \frac{1}{K}\sum_{k=1}^{K} \underbrace{\Big[\frac{1}{m}\sum_{i=1}^{m} d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y = k), \mathcal{D}_i(\mathbf{x}|Y = k)) \Big]}_{\textit{Achieved by pair-wise semantic matching}} \end{aligned} \tag{5}$$

Remark: The first term is the balanced source errors that can help to handle the label distributions shift. The second term is a small constant. The third term could be minimized by a pair-wised semantic matching scheme, and we will elaborate this point in the next section.

Now, we show that by aligning the semantic conditional distributions $\mathcal{D}(\mathbf{x}|y)$, we could also align the marginal distributions $\mathcal{D}(\mathbf{x})$. We notice that, for a pair of source domain

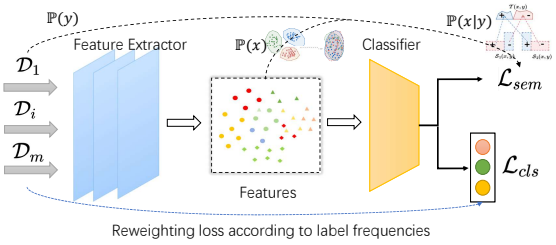


Figure 4: The overall model architecture. The feature extractor is trained to find out the shared features, and the extracted features are also used for computing the centroids of the semantic distributions to compute the semantic objective. The classifier is trained using the source domain data and is committed to performing well on the unseen target domain. For all the domains, the model parameters are shared with each other and trainable on all the source domains.

distributions $\mathcal{D}_i$ and $\mathcal{D}_j$,

$$\begin{aligned}
&\mathbb{E}_x|\mathcal{D}_i(\mathbf{x}) - \mathcal{D}_j(\mathbf{x})| = \mathbb{E}_x \sum_y |\mathcal{D}_i(y)\mathcal{D}_i(\mathbf{x}|y) - \mathcal{D}_j(y)\mathcal{D}_j(\mathbf{x}|y)| \\
&= \mathbb{E}_x \sum_{k=1}^{K} |\mathcal{D}_i(Y=k)\mathcal{D}_i(\mathbf{x}|Y=k) - \mathcal{D}_j(Y=k)\mathcal{D}_j(\mathbf{x}|Y=k)| \\
&= \frac{1}{K}\mathbb{E}_x| \sum_y (\mathcal{D}_i(\mathbf{x}|y) - \mathcal{D}_j(\mathbf{x}|y))| \\
&\leq \frac{1}{K} \sum_y \mathbb{E}_x|\mathcal{D}_i(\mathbf{x}|y) - \mathcal{D}_j(\mathbf{x}|y)| \\
&= \frac{2}{K} \sum_y d_{\mathrm{TV}}(\mathcal{D}_i(\mathbf{x}|y), \mathcal{D}_j(\mathbf{x}|y))
\end{aligned} \tag{6}$$

Eq. 6 shows that by minimizing the total variation distance between the two semantic conditional distributions $\mathcal{D}_i(\mathbf{x}|y)$ and $\mathcal{D}_j(\mathbf{x}|y)$, we could also take care of the marginal distribution of these two domains $\mathcal{D}_i(\mathbf{x})$ and $\mathcal{D}_j(\mathbf{x})$. That is, *when matching the semantic conditional distributions, we could also align the marginal features simultaneously.*

Now, based on the analysis above, we could summarize that to minimize the target risk, we need to follow the two principles:

- minimizing the weighted source risks (will be introduced in Eq. 8).
- matching the semantic divergences between each source domains (will be introduced in Eq. 14).

With these two principles, we could introduce our methodology in the next section.

4.2. Methodology

4.2.1. The overview of our model

The model architecture is presented in Fig. 4. It consists of two parts: feature extractor and classifier. The feature extractor, parameterized by $\boldsymbol{\theta}^f$, is trained to extract both feature and semantic information that is shared across the sources domains. Once the domains are aligned properly, the classifier, parameterized by $\boldsymbol{\theta}^c$, is trained to make universal predictions for all the domains. For classification, we adopt the cross-entropy loss.

$$\ell = -\sum_{i=1}^{m}\sum_{j=1}^{N_i} y_j^{(i)} \log(\mathbb{P}(\boldsymbol{\theta}^c(\boldsymbol{\theta}^f(\mathbf{x}_j^{(i)})))) \tag{7}$$

As analyzed before, to minimize the risk of the prediction on the target domain, we should both control the semantic conditional distance and the label distribution divergence. In case of the label distributions differ from each other, some minor classes may be regarded as noise, and the minor classes will be neglected Zhou et al. (2021a). In order to alleviate the impacts of the source domains' label space shifts, we could re-weigh the importance of each class to correct the loss based the total instance number in that category Lipton et al. (2018),

$$\hat{\mathcal{L}}^{\boldsymbol{\alpha}}_{\mathcal{D}_i}(h) = \sum_{(\mathbf{x}_i, y_i)\in\hat{D}_i} \alpha(y_i)\ell(h(\mathbf{x}_i), y_i)) \tag{8}$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots \alpha_k, \dots, \alpha_K]^T$ is the weighting vector for all K classes in each domain. For a certain class k, suppose we have m_k instances in that category, we could compute the weight by,

$$\alpha_k = \sum \frac{|\mathbb{1}[y = y_k]|}{m_k} \tag{9}$$

Through Eq. 9, the cross-entropy loss could be reweighted via the frequency of the number of instances from a specific class, which could ensure the data from different classes among all the domains could have the same probability to be sampled during training. By this process, the learner will be guided to pay attention to the classes with few instances, which could help to handle the label distribution drift. Then, the classification objective could be computed as,

$$\mathcal{L}^{\boldsymbol{\alpha}}_C = \sum_{i=1}^{m} \hat{\mathcal{L}}^{\boldsymbol{\alpha}}_{\mathcal{D}_i} = \sum_{i=1}^{m} \sum_{(\mathbf{x}_i, y_i)\in\hat{D}_i} \alpha(y_i)\ell(h(\mathbf{x}_i), y_i)) \tag{10}$$

Except for the reweighted loss, we also need to guide the learner to leverage the semantic distributions $\mathcal{D}(\mathbf{x}|y)$ across the domains. To this end, we adopt the extracted features z_i from domain i, to condition the semantic distributions $\mathbb{P}(z|y)$.

To align the semantic distributions, *i.e.*, minimizing $D_{JS}\big(\mathcal{D}_i(\mathbf{x}|y)\|\mathcal{D}_j(\mathbf{x}|y)\big)$ for all domains pairs i, j, one could have several solutions (*e.g.* conditional GAN training, moment matching, etc.). We adopted an alternative yet popular approach: *class-level feature mean matching* method that is prevalent in the general machine learning literature (e.g. Dou et al., 2019; Chopra et al., 2005; Xie et al., 2018; Zhou et al., 2021a).

The semantic minimization objective is computed across all the source domains. We can take out the extracted

features and compute the corresponding semantic centroids. For instances from source domains $S_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^{N_i}$ from all the categories $k \in \{1 \dots K\}$, similarly with Dou et al. (2019), we condition the extracted features on each class k to measure the semantic conditional distributions. Then, the empirical semantic centroid is estimated by,

$$\begin{aligned}\hat{z}_{c_i}^k &= \frac{1}{|D_i^k|}\sum_{x_i \in D_i^k} z_i^k = \frac{1}{|D_i^k|}\sum_{x_i \in D_i^k} \theta^f(\mathbf{x}_j) \\ &\approx \mathbb{E}_{D_i}[\theta^f(\mathbf{x}_i)|Y = k]\end{aligned} \tag{11}$$

Through this process, we compute the feature centroids. We then follow the strategy of (Xie et al., 2018; Zhou et al., 2021a) to maintain a global matrix $\mathcal{Z}_{D_i}$ for each source domain to maintain the semantic centroids,

$$\mathcal{Z}_{D_i}^k \leftarrow \gamma \hat{z}_{c_i}^k + (1-\gamma)\hat{z}_{c_i}^k \tag{12}$$

Eq.12 defines a moving averaging method for the batch training of $\mathcal{Z}$, where λ is a coefficient to control the moving average temperature. Then, we could maintain a matrix $\mathcal{Z}_i = [\mathcal{Z}_{D_i}^1, \dots, \mathcal{Z}_{D_i}^K]^T$ to trace the semantic relations between domains, through which we could match the semantic distributions via minimize the Euclidean distance $\Phi(\mathcal{Z}_{D_i}^k, \mathcal{Z}_{D_j}^k)$ between two centroids in the embedding space, which is computed as,

$$\Phi(\mathcal{Z}_{D_i}^k, \mathcal{Z}_{D_j}^k) = \|\mathcal{Z}_{D_i}^k - \mathcal{Z}_{D_j}^k\|^2 \tag{13}$$

Here the function $\Phi(\mathcal{Z}_{D_i}^k, \mathcal{Z}_{D_j}^k)$ is the approximation of the total variation $d_{TV}(\mathcal{Z}_{D_i}^k, \mathcal{Z}_{D_j}^k)$, which is the upper bound of $D_{JS}(\mathcal{Z}_{D_i}^k \| \mathcal{Z}_{D_j}^k)$. Then for each training epoch, the semantic loss $\mathcal{L}_S$ is updated by,

$$\mathcal{L}_{Sem} \leftarrow \mathcal{L}_{Sem} + \Phi(\mathcal{Z}_{D_i}, \mathcal{Z}_{D_j}) \tag{14}$$

By minimizing the semantic objectives of all the domains, we could achieve semantic invariant features.

Now, with the components described above, we could summarize the learning objective of our method as,

$$\mathcal{L} = \mathcal{L}_C^{\boldsymbol{\alpha}} + \lambda_s \mathcal{L}_{Sem} \tag{15}$$

where $\mathcal{L}_C^{\boldsymbol{\alpha}}$ is the modified classification objective defined in Eq. 10, $\mathcal{L}_{Sem}$ is the semantic learning objective defined in Eq. 14 and λ_s is a coefficient to regularize the semantic learning objective.

Remark: The learning objective $\mathcal{L}_{Sem}$ can be viewed as an extra regularization term on top of the classification objective, which can ensure semantic invariance, leading to better generalization performances.

We show the whole learning process in Algorithm 1 and the model architecture in Fig. 4. The algorithm mainly consists of several parts: first, to measure the label distributions and compute the reweighted classification objective to enforce the class-level alignment, and second to enforce the domain-level semantic alignment for all the domains. We then evaluate the effectiveness of our method in the next part.

Algorithm 1 The proposed SMDG algorithm

Require: Samples from different source domains $\{D_i\}_{i=1}^m$

Ensure: Neural network parameters θ^f, θ^c

1: **for** mini-batch of samples $\{(\mathbf{x}_s^{(i)}, y_s^{(i)})\}$ from source domains **do**

2: Compute the classification loss $\mathcal{L}_C^{\boldsymbol{\alpha}}$ over all the domains according to Eq. 10

3: Mix the instances and compute the semantic matching objective $\mathcal{L}_{Sem}$ via Eq. 14

4: Update θ^f, θ^c by solving Eq. 15 with learning rate η:

$$\begin{aligned}\boldsymbol{\theta}_f &\leftarrow \boldsymbol{\theta}_f - \eta\frac{\partial(\mathcal{L}_C^{\boldsymbol{\alpha}} + \lambda_s\mathcal{L}_{Sem})}{\partial\boldsymbol{\theta}^f}, \\ \boldsymbol{\theta}_c &\leftarrow \boldsymbol{\theta}_c - \eta\frac{\partial(\mathcal{L}_C^{\boldsymbol{\alpha}} + \lambda_s\mathcal{L}_{Sem})}{\partial\boldsymbol{\theta}^c}\end{aligned}$$

5: **end for**

6: Return the optimal parameters $\boldsymbol{\theta}^{f\star}$ and $\boldsymbol{\theta}^{c\star}$

Table 1
Empirical Results (accuracy %) on each target domain on PACS dataset. (Some results of the proposed method in this table are under double check)

Method	Art	Cartoon	Sketch	Photo	Avg.
Deep All	63.30	63.13	54.07	87.70	67.05
CDANN	62.70	69.73	64.45	78.65	68.88
MLDG	66.23	66.88	58.96	88.00	70.01
D-SAM	63.87	70.70	64.66	85.55	71.20
JiGen	67.63	71.71	65.18	89.00	73.38
MMLD	**69.27**	**72.83**	66.44	88.98	74.38
VREx	67.04	67.97	89.74	59.81	71.14
Ours	67.87	72.14	**70.16**	90.45	**75.16**

5. Experiments and Results

We verify the effectiveness of the proposed approach on several common-used benchmarks, including the PACS, VLCS and Office-home dataset, comparing with several baselines using common evaluation protocols. Furthermore, apart from these aforementioned benchmarks, we also evaluate the algorithm on the recent *DomainBed* framework. We first evaluate the results compared with baselines showing the state-of-the-art performance on benchmarks. To further understand the method, we then do the ablation studies, evaluations under label distributions shift as well as time efficiency evaluations to confirm the effectiveness of our method.

Table 2
Empirical Results (accuracy %) on VLCS dataset with pre-trained AlexNet as Feature Extractor.

Method	Caltech	LabelMe	Pascal	Sun	Avg.
Deep All	92.86	63.10	68.67	64.11	72.19
D-MATE	89.05	60.13	63.90	61.33	68.60
CDANN	88.83	63.06	64.38	62.10	69.59
TF	93.63	**63.49**	69.99	61.32	72.11
MMD-AAE	94.40	62.60	67.70	64.40	72.28
D-SAM	91.75	56.95	58.95	60.84	67.03
MLDG	94.4	61.3	67.7	65.9	73.30
JiGen	96.93	60.90	70.62	64.30	73.19
MMLD	96.66	58.77	**71.96**	**68.13**	73.88
S-MLDG	96.40	64.80	64.00	68.70	73.50
VREx	96.72	60.40	63.68	70.49	73.30
Ours	**97.54**	63.41	69.36	65.63	**73.98**

Table 3
Empirical Results on Office-home dataset with pre-trained ResNet-18 as feature extractor

	Art	Clipart	Product	Real-World	Avg.
Deep All	52.15	45.86	70.86	73.15	60.51
D-SAM	58.03	44.37	69.22	71.45	60.77
JiGen	53.04	**47.51**	71.47	72.79	61.20
JAN-COMBO	48.09	45.20	66.52	68.35	57.04
SagNets	60.20	45.38	70.42	73.38	62.34
WADG	55.34	44.82	72.03	73.55	61.44
Ours	**58.76**	45.49	**72.46**	**75.21**	**62.98**

Table 4
Empirical Results (accuracy %) on PACS dataset with pre-trained ResNet-18 as feature extractor.

Method	Art	Cartoon	Sketch	Photo	Avg.
Deep All	77.87	75.89	69.27	95.19	79.55
D-SAM	77.33	72.43	77.83	95.30	80.72
JiGen	79.42	75.25	71.35	96.03	80.51
MASF	80.29	77.17	71.69	94.99	81.04
MMLD	81.28	77.16	72.29	96.09	81.83
S-MLDG	80.50	77.80	72.80	94.80	81.50
DGER	80.70	76.40	71.77	**96.65**	81.38
DDAIG	84.20	78.10	74.70	95.30	83.10
SagNets	83.58	77.66	76.30	95.47	83.25
WADG	**81.56**	78.02	78.42	95.82	83.45
Ours	81.10	**79.66**	**78.92**	95.87	**83.89**

5.1. Baselines and Implementation Details

We test our algorithm on the benchmark datasets with the following principled domain generalization approaches. Specifically, we consider several principled approaches: 1) matching-based approaches, 2) meta-learning-based approaches and 3) conditional alignment approaches. Specially, we compared the following baselines on the benchmarks: ***Deep All***: Train the model on source domains only. We implement the pre-trained AlexNet or ResNet-18 as the feature extractor and aggregate the classification loss of all source domains as the learning objective; ***CDANN*** (Li et al., 2018c): We adopt the conditional alignment method by (Li et al., 2018c), which targets to extract the conditional-invariant feature via varying the class prior so that the conditional distributions among domains could be matched; ***MLDG*** (Li et al., 2018a): MLDG is a meta-learning based domain generalization method. It stimulates the domain shift by splitting the source data into *meta-train*, and *meta-test* sets to learn the invariant features for generalization; ***D-SAM*** (D'Innocente and Caputo, 2018): It is a method that aggregates several domain-specific modules, which allows the model to merge general and specific information from all the domains to generalize to a new domain; ***MMD-AAE*** (Li et al., 2018b): is a Mean-Max Discrepancy (MMD) based approach to map the latent features to kernel space for the MMD minimization. The model is combined with the Adversarial AutoEncoder (AAE) model with shallow layers, and later in this work, we adopt their MMD mappings with a deep model while relaxing the reconstruction objective. ***MixUp*** (Yan et al., 2020): It proposes to leverage the feature level consistency to facilitate the inter-domain regularization. ***JiGen*** (Carlucci et al., 2019): It leverages the Jigsaw puzzle under an unsupervised task to achieve domain invariant features for generalization. ***MASF*** (Dou et al., 2019): MASF is also a meta-learning-based approach that combines the MLDG with the Constrictive Loss and Triplet Loss to encourage class-level alignment. ***MMLD*** (Matsuura and Harada, 2020): MMLD is an approach that mixes all the source features together with an unsupervised objective to extract domain-independent feature space. ***DGER*** (Zhao et al., 2020): DGER is an approach that focuses on minimizing the prediction entropy. ***DDAIG*** (Zhou et al., 2020a): DDAIG is a generation-based method that consists of a domain transformation module to the unseen domain. **DomainBed** (Gulrajani and Lopez-Paz, 2021): DomainBed is a unified framework that compromises several recent baselines with standard evaluation benchmarks; We adopt the baselines provided therein. ***WADG*** (Zhou et al., 2021b): is a method that combines the Wasserstein adversarial training with a metric similarity learning objective to achieve both the domain-level and class-level alignment.

We first adopt the pre-trained AlexNet model as the feature extractor to evaluate the algorithms on the PACS and VLCS datasets. For the PACS and VLCS datasets on AlexNet, we train the model with mini-batch size 64 and test batch-size 16. The model is trained with Adam optimizer with a learning rate of 2×10^{-4} for a total of 180 epochs. For the AlexNet backbone, we extract the intermediate layer feature with size 256 to match the semantic features.

The results on PACS and VLCS benchmarks with AlexNet are represented in Table. 1 and Table. 2, respectively. We refer to the results of the baseline using the original value reported in their manuscripts. From the results, we could see that our method could outperform the baselines on these two benchmarks by achieving state-of-the-art performance. We then follow the evaluation protocols of (Zhou et al., 2021b; Dou et al., 2019; Matsuura and Harada, 2020) to implement the experiments on the

Table 5
Empirical Results on Office-Home Dataset with pre-trained ResNet 50 as the feature extractor.

Method	Art	Clipart	Product	Real-World	Avg.
ERM	61.3	52.4	75.8	76.6	66.5
IRM	58.9	52.2	72.1	74.0	64.3
GroupDRO	60.4	52.7	75.0	76.0	66.0
MMD	60.4	53.3	74.3	77.4	66.3
DANN	59.9	53.0	73.6	76.9	65.9
CDANN	61.5	50.4	74.4	76.6	65.8
MTL	61.5	52.4	74.9	76.8	66.4
ARM	58.9	51.0	74.1	75.2	64.8
VREx	60.7	53.0	75.3	76.6	66.4
RSC	60.7	51.4	74.8	75.1	65.5
Ours	58.9	55.1	75.3	77.1	66.6

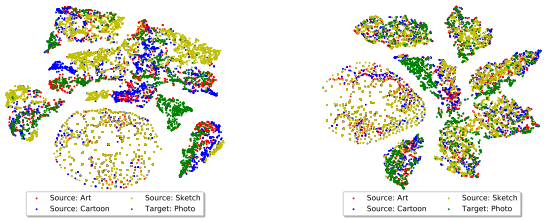
PACS dataset and Office-home with deeper backbones as the feature extractor to show the benefits of our method. We adopted the pre-trained ResNet-18 model as the feature extractor and trained the model with mini-batch size 64 and test batch size 16. The model is optimized with Adam optimizer with a learning rate of 2×10^{-4} to 5×10^{-5} on PACS and VLCS datasets while 3×10^{-3} on the Office-home dataset. For the ResNet backbone, we extract the intermediate layer feature with size 256 for computing the semantic matching objective. The test results on PACS and Office-Home benchmarks with ResNet-18 feature extractor are reported in table 4 and Table 3, respectively. For the experimental results on all the datasets, we empirically set $\lambda = 0.1$ and $\gamma = 0.3$.

We then evaluate our algorithm on a more recent challenging framework, namely *DomainBed* Gulrajani and Lopez-Paz (2021), to verify the empirical performances. Following the setting of DomainBed, we opt for the pre-trained ResNet-50 model as the feature extractor and conduct the experiments on OfficeHome. The results are displayed in Table 5. We set the learning rate as 5×10^{-5} using the Adam optimizer. More details for the experiments are delegated to the Appendix files.

From the test result, we could observe an improvement on the benchmarks performances achieving the state-of-the-art performances. Furthermore, here we would like to note that, compared with the methods based on the metric learning objectives (*e.g.*, Dou et al. (2019); Zhou et al. (2021b)), our method doesn't require a large batch size for the triplet property to achieve better performances. For example, on the Office-Home benchmark, to ensure the triplet property, one needs a batch size of at least 195. When we adopt some deeper backbones (*e.g.*, ResNet-50) as feature extractors, the computational cost will be prohibitive. This also confirms the effectiveness of our method.

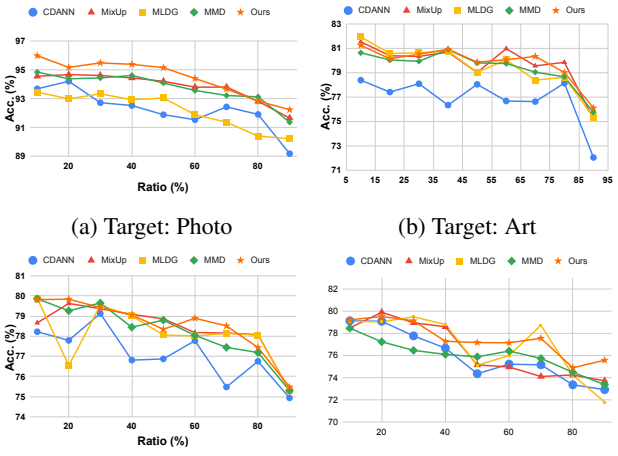
5.2. Further Analysis

Except for the standard benchmark evaluations, we then further investigate our method in several aspects, including the t-SNE visualizations, ablation studies, performance under label shift and time efficiencies.



(a) Source only (b) Full method

Figure 5: t-SNE visualizations of our method on PACS dataset



(a) Target: Photo (b) Target: Art

(c) Target: Cartoon (d) Target: Sketch

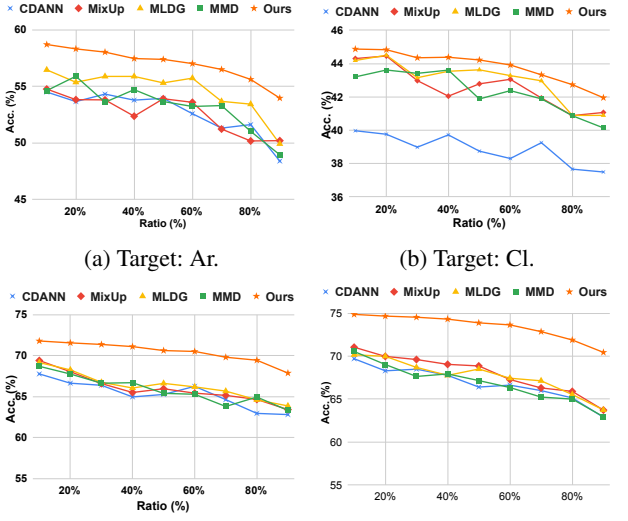
Figure 6: Performance comparison under label shift situation on PACS dataset with respect to the four target domains.

t-SNE Visualization We first show the t-SNE visualization of our method to show the alignment performance on the PACS dataset, comparing the source-only training only and the full method. The results on PACS is illustrated on Fig. 5. The results show that our method could well align the features, which confirms the effectiveness of our method on category alignment.

Ablation studies To confirm the effectiveness of each component of our proposed method, we did the ablation studies on each part of our proposed work. We implement the following ablations: 1) *Cls. only*: only train the model on the source domains using the classification objectives without the re-weighting technique; 2) *No Sem.* We omit the semantic alignment objective while keeping the classification objective with the re-weighting technique; 3) *No Re-weighting*: We omit the re-weighting technique in the classification objective while keeping the semantic matching and original cross-entropy classification objective. To better evaluate the effectiveness of our method with depth understanding, we implement the ablations on PACS dataset with AlexNet and ResNet-18 model as feature extractor, as well as the ablation studies on Office-Home dataset with ResNet-18 as the feature extractor. The results of ablation

Table 6
The ablation studies on PACS and Office-Home datasets.

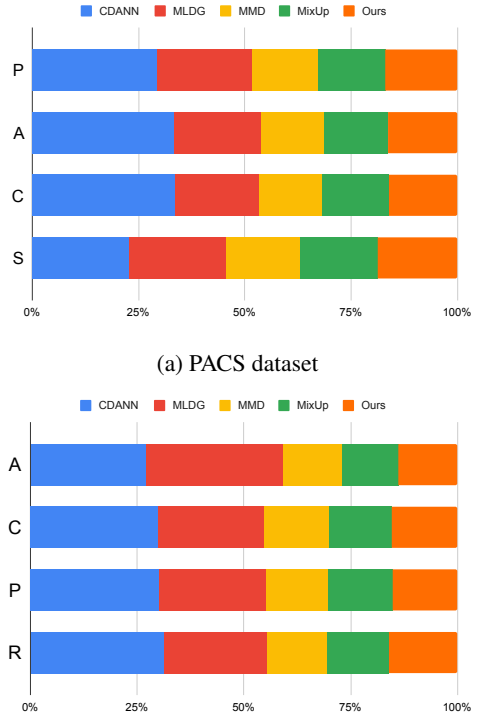
Benchmark	PACS-AlexNet					PACS-ResNet18					Office-Home				
Ablation	A	P	C	S	Avg.	A	P	C	S	Avg.	Ar	Cl	Pr	Rw	Avg.
Deep-All	63.30	87.7	63.13	54.07	67.05	77.87	95.19	75.89	69.27	79.55	52.15	45.86	70.86	73.15	60.51
No-sem.	64.40	87.37	67.55	65.36	71.71	80.08	94.68	79.26	76.75	82.69	57.37	43.37	71.51	73.93	61.54
No re-weight	64.55	86.55	68.33	68.70	72.03	79.17	94.91	78.85	76.71	82.41	58.35	45.06	72.21	75.05	62.67
Full	67.87	90.45	72.14	70.16	75.16	81.10	79.66	78.92	95.87	83.89	72.46	58.76	45.49	75.21	62.98



(a) Target: Ar. (b) Target: Cl.

(c) Target: Pr. (d) Target: Rw.

Figure 7: Performance comparison under label shift situation on Office-Home dataset with respect to the four target domains for each generalization task.



(a) PACS dataset

(b) Office-Home dataset

Figure 8: Relative time comparison on PACS and Office-Home dataset.

studies are presented in Table 6. As we could observe from the ablation results, semantic domain alignment is crucial to our method. If we omit the semantic alignment objective, there could be a rapid drop-off in the performance. Besides, the label correction objective could also help to improve the performance compared with the original cross-entropy learning objective.

Performance under label distribution shift As our theoretical analysis (section 4.1) demonstrates the necessity of controlling the label shift and our algorithm is committed to handling label distribution shift. To confirm the effectiveness of overcoming label shift problems, we conduct the experiments to check the DG algorithms' performance under label shift scenarios where the label distributions from all the domains drift from each other, *i.e.*, we randomly remove a certain percentage of instances from each domain. We implement the label drift process on PACS and Office-Home datasets. We compared our method with the following four principled methods: 1) The conditional alignment method, namely the CDANN method Li et al. (2018c), 2) The meta learning-based method, namely the MLDG method Li et al. (2018a), 3) The Mean-Max Discrepancy (MMD) minimization based method Li et al. (2017) and 4) The MixUp method Yan et al. (2020). For the PACS dataset, for each source domain, we remove a certain ratio (10% $\sim$ 90%) of instances from 2 classes. For the Office-Home dataset, for each source domain, we remove a certain ratio (10% $\sim$ 90%) of instances randomly from 15 categories. The compared results curves on PACS and Office-Home datasets with different target domains are illustrated in Fig. 6 and Fig. 7, respectively.

From the results, we could observe that our method could outperform the baselines under all the drift ratios. Specifically, on the PACS dataset, we could observe that the MLDG method and MixUp method could have a similar performance comparing with ours under certain shift ratios when choosing *Art* and *Sketch* as the target domain. However, on the Office-Home dataset, our method could have obvious improvements compared with all the baselines, which confirmed the effectiveness of our method. Since

the number of classes of the PACS dataset (7) is obviously smaller than the number of classes of Office-Home dataset (65), the simulated label shift does not have obvious changes to the data distribution, which may lead to similar performances on the shift on PACS dataset. Furthermore, the number of instances in each domain of PACS dataset is relatively more than the number of instances in each domain of the Office-Home dataset. Thus, the baseline methods are more sensitive to label shift on the Office-Home benchmark than PACS benchmark. This also confirms the effectiveness of our method when handling a minor number of instances when the label shift problem occurs.

Time efficiency We then evaluate the time efficiency of our method, comparing it with the four principled baselines on both the PACS and Office-Home benchmark to demonstrate the effectiveness of our method. We demonstrate the time efficiency by comparing the relative average time, setting our time as the unit time for one training round. The results are presented as a relative percentage bar chart by setting the time costs of our method as a unit in Fig. 8. From the results, we could observe that our method has similar time efficiency with MMD and MixUp methods while has better time efficiency than CDANN and MLDG.

6. Conclusion

In this work, we considered the generalization property in DG problems via exploring the value of the label and semantic information across domains, which were mostly neglected by the previous work. We investigated the theoretical guarantee for a successful generalization process by focusing on how to control the target domain error. Our results revealed that to control the target risk, we should jointly control the source errors that are weighted according to label information and align the semantic conditional distributions between different source domains. The theoretical analysis then inspired an efficient algorithm to control the label distributions and match the semantic conditional distributions The empirical results showed that our method outperformed most of the baselines, achieving state-of-the-art performances on the benchmarks. Furthermore, the time efficiency of the method showed that our method could achieve better benchmark performances with better time efficiencies. Besides, our method also showed better performances under the label shift situations, which could not perfectly be handled by the baselines.

Acknowledgement

This work has been partially supported by the Natural Sciences and Engineering Research Council Discovery Grant, and partially supported by the China Scholarship Council.

Supplementary Materials

A. Proof to the theoretical results

In this section we provide the proof to Theorem 1, Corollary 1 and Corollary 2.

A.1. Proof to Theorem 1

Proof. Let's first consider the risk on the target domain *w.r.t* to the nearest source domain $\mathcal{D}^\star$,

$$\begin{aligned} R_{\mathcal{D}_t}(h) &\leq R_{\mathcal{D}^\star}(h) + d_{TV}(\mathcal{D}_t, \mathcal{D}^\star) \\ &= \min_{\mathcal{D}_1,\dots,\mathcal{D}_m} R_{\mathcal{D}_i} + d_{TV}(\mathcal{D}_t, \mathcal{D}_i) \end{aligned} \tag{16}$$

In the context of DG, the learner has no access to the target domain, so we have no idea about which source domain is the nearest one to the target. In this case, we try to find out the minimization over all the source domains,

$$\begin{aligned} R_{\mathcal{D}_t} \leq R_{\mathcal{D}_1}(h) + d_{TV}(\mathcal{D}_t, \mathcal{D}_1) \leq R_{\mathcal{D}_1} + d_{TV}(\mathcal{D}^\star, \mathcal{D}_1) \\ + d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \\ \dots \\ R_{\mathcal{D}_t} \leq R_{\mathcal{D}_i}(h) + d_{TV}(\mathcal{D}_t, \mathcal{D}_i) \leq R_{\mathcal{D}_i} + d_{TV}(\mathcal{D}^\star, \mathcal{D}_i) \\ + d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \\ \dots \\ R_{\mathcal{D}_t} \leq R_{\mathcal{D}_m}(h) + d_{TV}(\mathcal{D}_t, \mathcal{D}_m) \leq R_{\mathcal{D}_m} + d_{TV}(\mathcal{D}^\star, \mathcal{D}_m) \\ + d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \end{aligned} \tag{17}$$

Sum over all the source domain i, we have,

$$\begin{aligned} m \cdot R_{\mathcal{D}_t}(h) \leq & R_{\mathcal{D}_1,\dots,\mathcal{D}_m}(h) + m \cdot d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \\ & + \sum_i d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \end{aligned} \tag{18}$$

Then, we have,

$$\begin{aligned} R_{\mathcal{D}_t}(h) \leq & \frac{1}{m} R_{\mathcal{D}_1,\dots,\mathcal{D}_m}(h) + d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \\ & + \frac{1}{m} \sum_i d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) \end{aligned} \tag{19}$$

Note $d_{TV}(\mathcal{D}^\star, \mathcal{D}_t) = \epsilon^\star$ and $\frac{1}{m} R_{\mathcal{D}_1,\dots,\mathcal{D}_m}(h)$ is the averaged source errors, we conclude the proof. □

A.2. Proof to Corollary 1

Since Theorem 1 is represented by the joint distribution, in order to show the insights that can motivate the benefits on controlling the semantic and label distribution, we can further provide the proof of Corollary 1.

Proof. Since $d_{TV}(\mathcal{D}^\star, \mathcal{D}_i) \leq 2\sqrt{D_{JS}(\mathcal{D}^\star \| \mathcal{D}_i)}$, plug into Eq. 3, we have

$$
\begin{aligned}
R_{\mathcal{D}_t}(h) \leq & \frac{1}{m}\sum_{i=1}^{m} R_{\mathcal{D}_i}(h) + \epsilon^\star \\
& + \frac{1}{m}\sum_{i}^{m} d_{TV}(\mathcal{D}^\star(\mathbf{x},y), \mathcal{D}_i(\mathbf{x},y)) \\
\leq & \frac{1}{m}\sum_{i=1}^{m} R_{\mathcal{D}_i}(h) + \epsilon^\star \\
& + \frac{2}{m}\sum_{i}^{m} [\sqrt{D_{JS}(\mathcal{D}^\star(\mathbf{x},y)||\mathcal{D}_i(\mathbf{x},y))}]
\end{aligned} \tag{20}
$$

Now we need to bound the third term of Eq. 20. Similar with (Zhou et al., 2021a; Shui et al., 2020), we can introduce an intermediate distribution $\mathcal{M}(\mathbf{x}) = \frac{1}{2}(\mathcal{D}^\star(\mathbf{x}) + \mathcal{D}_i(\mathbf{x}))$, then $\text{supp}(\mathcal{D}_i) \subseteq \text{supp}(\mathcal{M})$ we notice that,

$$
\begin{aligned}
& 2D_{\text{JS}}(\mathcal{D}^\star(\mathbf{x},y)\|\mathcal{D}_i(\mathbf{x},y)) = \\
& D_{\text{KL}}(\mathcal{D}^\star(\mathbf{x},y)\|\mathcal{M}(\mathbf{x},y)) + D_{\text{KL}}(\mathcal{D}_i(\mathbf{x},y)\|\mathcal{M}(\mathbf{x},y)) \\
& = D_{\text{KL}}(\mathcal{D}^\star(y)\|\mathcal{M}(y)) + \mathbb{E}_{x\sim\mathcal{D}^\star(y)} D_{\text{KL}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& + D_{\text{KL}}(\mathcal{D}_i(y)\|\mathcal{M}(y)) + \mathbb{E}_{x\sim\mathcal{D}_i(y)} D_{\text{KL}}(\mathcal{D}_i(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& = 2D_{\text{JS}}(\mathcal{D}^\star(y)\|\mathcal{D}_i(y)) + \mathbb{E}_{x\sim\mathcal{D}^\star(y)} D_{\text{KL}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& + \mathbb{E}_{x\sim\mathcal{D}_i(y)} D_{\text{KL}}(\mathcal{D}_i(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y))
\end{aligned} \tag{21}
$$

Then, we provide two bounded term for the last two KL divergence based terms,

$$
\begin{aligned}
& \mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{\text{KL}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& \leq \mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{\text{KL}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& + \mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{\text{KL}}(\mathcal{D}_i(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& = 2\mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{\text{JS}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{D}_i(\mathbf{x}|y))
\end{aligned}
$$

Similarly, we could also have,

$$
\begin{aligned}
& \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i(x)} D_{\text{KL}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{M}(\mathbf{x}|y)) \\
& \qquad \leq 2\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i(x)} D_{\text{JS}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{D}_i(\mathbf{x}|y))
\end{aligned}
$$

Plug these two terms into Eq 21, we have

$$
\begin{aligned}
D_{\text{JS}}(\mathcal{D}^\star(\mathbf{x},y)\|\mathcal{D}_i(\mathbf{x},y)) \leq & D_{\text{JS}}(\mathcal{D}^\star(y)\|\mathcal{D}_i(y)) \\
& + \mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{\text{JS}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{D}_i(\mathbf{x}|y)) \\
& + \mathbb{E}_{y\sim\mathcal{D}_i(y)} D_{\text{JS}}(\mathcal{D}^\star(\mathbf{x}|y)\|\mathcal{D}_i(\mathbf{x}|y))
\end{aligned} \tag{22}
$$

Now we have

$$
\begin{aligned}
\sqrt{\text{R.H.S. of Eq. 22}} \leq & \sqrt{D_{JS}(\mathcal{D}^\star(y)||\mathcal{D}_i(y))} \\
& + \sqrt{\mathbb{E}_{y\sim\mathcal{D}^\star(y)} D_{JS}(\mathcal{D}^\star(\mathbf{x}|y)||\mathcal{D}_i(\mathbf{x}|y))} \\
& + \sqrt{\mathbb{E}_{y\sim\mathcal{D}_i(y)} D_{JS}(\mathcal{D}^\star(\mathbf{x}|y)||\mathcal{D}_i(\mathbf{x}|y))}
\end{aligned} \tag{23}
$$

Plug Eq. 23 into Eq. 20, we conclude the proof. □

A.3. Proof to Corollary 2

Now we show the proof to Corollary 2.

Proof. First consider the risk in the testing phase, *i.e.*, the prediction loss on the target domain,

$$
\begin{aligned}
R_{\mathcal{D}_t}(h) = & \frac{1}{K}\sum_{k=1}^{K}\int_x \mathcal{D}_t(\mathbf{x}|Y=k)\mathcal{L}(h(\mathbf{x}),y) \\
\leq & \frac{1}{K}\sum_{k=1}\Big[\mathbb{E}_{x\sim\mathcal{D}^\star}(\mathbf{x}|Y=k)\mathcal{L}(h(\mathbf{x}),y) \\
& + d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_t(\mathbf{x}|Y=k))\Big]
\end{aligned} \tag{24}
$$

Similar with the proof of Theorem 1, we could bound the two items in Eq. 24,

$$
\begin{aligned}
\mathbb{E}_{x\sim\mathcal{D}^\star}\mathcal{L}(h(\mathbf{x}),y) \leq & \mathbb{E}_{x\sim\mathcal{D}_1}\mathcal{L}(h(\mathbf{x}),y) \\
& + d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_1(\mathbf{x}|Y=k)) \\
& \dots \\
\mathbb{E}_{x\sim\mathcal{D}^\star}\mathcal{L}(h(\mathbf{x}),y) \leq & \mathbb{E}_{x\sim\mathcal{D}_i}\mathcal{L}(h(\mathbf{x}),y) \\
& + d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_i(\mathbf{x}|Y=k)) \\
& \dots \\
\mathbb{E}_{x\sim\mathcal{D}^\star}\mathcal{L}(h(\mathbf{x}),y) \leq & \mathbb{E}_{x\sim\mathcal{D}_m}\mathcal{L}(h(\mathbf{x}),y) \\
& + d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_m(\mathbf{x}|Y=k))
\end{aligned} \tag{25}
$$

Sum this Eq. 25 and plug into Eq. 24, we have,

$$
\begin{aligned}
R_{\mathcal{D}_t}(h) \leq & \frac{1}{K}\sum_{k=1}^{K}\Big[\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{x\sim\mathcal{D}_i}\mathcal{L}(h) \\
& + \frac{1}{m}\sum_{i=1}^{m} d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_i(\mathbf{x}|Y=k)) \\
& + \frac{1}{m}\sum_{i=1}^{m} d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_t(\mathbf{x})|Y=k)\Big] \\
\leq & \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{x\sim\mathcal{D}_i}\hat{\mathcal{L}}^{\boldsymbol{\alpha}}_{\mathcal{D}_i}(h) + \kappa^\star \\
& + \frac{1}{K}\sum_{k=1}^{K}\Big[\frac{1}{m}\sum_{i=1}^{m} d_{TV}(\mathcal{D}^\star(\mathbf{x}|Y=k), \mathcal{D}_i(\mathbf{x}|Y=k))\Big]
\end{aligned} \tag{26}
$$

Then we could conclude the proof. □

B. Experimental Details

B.1. Datasets and Preparation

We compare our method with some baseline methods on VLCS, PACS and Office-home dataset. The VLCS dataset Torralba and Efros (2011) consists of four domains of images from *LabelMe*(L), *PASCAL-VOC2007*(V), *SUN-09*(S) and *Caltech-101*(C) with total five categories in each domain. Unlike some previous work (Li et al., 2018b; Dou et al., 2019), which adopts the *DeCAF* model (Donahue

et al., 2014) features (DeCAF6 features), we use the original dataset with images so that the model could explore the semantic features. The PACS dataset (Li et al., 2017) is a recent standard benchmark for DG which consists images from four domains: *Art* (A), *Cartoon* (C), *Photo* (P) and *Sketch* (S). Office-Home Venkateswara et al. (2017) is a more challenging dataset, which was widely investigated in recent DA and DG research. It contains images from four different domains: *Art* (Ar), *Clipart* (Cl), *Product* (Pr) and *Real World* (Rw). Images from all the domains have 65 categories. DomainNet Gulrajani and Lopez-Paz (2021) is a recent framework that provides a bundle of baselines with a standard evaluation framework.

For the PACS and VLCS set, we adopt the following pre-process pipeline: 1) for the training set, firstly resize the image to 224×224 using *RandomResizedCrop* function and then apply the *RandomHorizontalFlip* function. 2) for the testing set, only resize the image to 224×224 using using *RandomResizedCrop* function.

For the experiments results presented in Table 3, we follow the data pre-processing protocols of (Zhou et al., 2021b). For the evaluations presented in Table 5, we directly follow the data processing protocols of DomainBed Gulrajani and Lopez-Paz (2021) with the *MultipleDomainDataset* class provided therein. We report the results with training domain validation criteria.

B.2. Neural Networks model and Hyperparameters

We first implement the experiments on PACS and VLCS datasets with pre-trained AlexNet provided by PyTorch. We extract the intermediate layer feature with size 4096 to match the semantic features. The architecture of the classifier is implemented as follows,

- (Layer 0): Linear layer (in = 4096, out = 256, bias = True)
- (Layer 1): Linear layer (in = 256, out = # classes, bias = True)
- (Layer 2): Softmax (dim=-1)

The model is trained with Adam optimizer with a learning rate 2×10^{-4} with mini-batch size 64 for a total of 180 epochs.

We then conduct the experiments on PACS and Office-Home with the pre-trained ResNet-18 model as the feature extractor. With the ResNet-18 model, the output size of the feature extractor is 512. The classifier is implemented as follows,

- (Layer 0): Linear Layer (in = 512, out=256, bias = True)
- (Layer 1): Linear Layer (in = 256, out # classes, bias = True)
- (Layer 2): Softmax (dim=-1)

The intermediate layer features with size 256 are extracted to match the semantic features. The learning rate is set as 3×10^{-3} on the Office-home dataset and is set as 2×10^{-4} with ResNet-18 backbones.

classes of PACS data sets is 7, # classes of VLCS is 5 and # classes of Office-home is 65.

For the experiments conducted with DomainBed framework with ResNet-50 backbone on Office-Home dataset. The feature extractor and classifier are implemented as per the default model provided therein. We train the model with mini-batch size 24 due to the computation limitations. The model is trained with Adam optimizer with a learning rate 5×10^{-5} as per the default setting of DomainBed.

The value of λ and γ are determined by reverse validation. We set $\lambda = 0.1$, which is a common setting for regularization term of the DA (Ganin et al., 2016; Shen et al., 2018b) and DG (Zhou et al., 2021b). The value of γ, the coefficient to control the moving average temperature, is set as 0.3, which we found can have stable results.

References

Achituve, I., Maron, H., Chechik, G., 2021. Self-supervised learning for domain adaptation on point clouds, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 123–133.

Albuquerque, I., Monteiro, J., Falk, T.H., Mitliagkas, I., 2019. Adversarial target-invariant representation learning for domain generalization. arXiv preprint arXiv:1911.00804 .

Azizzadenesheli, K., Liu, A., Yang, F., Anandkumar, A., 2019. Regularized learning for domain adaptation under label shifts. arXiv preprint arXiv:1903.09734 .

Balaji, Y., Sankaranarayanan, S., Chellappa, R., 2018. Metareg: Towards domain generalization using meta-regularization, in: Advances in Neural Information Processing Systems, pp. 998–1008.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J., 2010. A theory of learning from different domains. Machine Learning 79, 151–175. URL: http://www.springerlink.com/content/q6qk230685577n52/.

Blanchard, G., Lee, G., Scott, C., 2011. Generalizing from several related classification tasks to a new unlabeled sample, in: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2011/file/b571ecea16a9824023ee1af16897a582-Paper.pdf.

Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T., 2019. Domain generalization by solving jigsaw puzzles, in: CVPR.

Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 539–546 vol. 1. doi:10.1109/CVPR.2005.202.

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

Dou, Q., de Castro, D.C., Kamnitsas, K., Glocker, B., 2019. Domain generalization via model-agnostic learning of semantic features, in: Advances in Neural Information Processing Systems, pp. 6447–6458.

Du, Y., Xu, J., Xiong, H., Qiu, Q., Zhen, X., Snoek, C.G., Shao, L., 2020. Learning to learn with variational information bottleneck for domain generalization, in: European Conference on Computer Vision, Springer. pp. 200–216.

D'Innocente, A., Caputo, B., 2018. Domain generalization with domain-specific aggregation modules, in: German Conference on Pattern Recognition, Springer. pp. 187–198.

Fang, Z., Lu, J., Liu, A., Liu, F., Zhang, G., 2021a. Learning bounds for open-set learning, in: International Conference on Machine Learning, PMLR. pp. 3122–3132.

Fang, Z., Lu, J., Liu, F., Xuan, J., Zhang, G., 2020. Open set domain adaptation: Theoretical bound and algorithm. IEEE transactions on neural networks and learning systems .

Fang, Z., Lu, J., Liu, F., Xuan, J., Zhang, G., 2021b. Open set domain adaptation: Theoretical bound and algorithm. IEEE Transactions on Neural Networks and Learning Systems 32, 4309–4322. doi:10.1109/TNNLS.2020.3017213.

Finn, C., Abbeel, P., Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org. pp. 1126–1135.

Fuglede, B., Topsoe, F., 2004. Jensen-shannon divergence and hilbert space embedding, in: International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings., pp. 31–. doi:10.1109/ISIT.2004.1365067.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., 2016. Domain-adversarial training of neural networks. The Journal of Machine Learning Research 17, 2096–2030.

Geng, C., Huang, S.j., Chen, S., 2020. Recent advances in open set recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Ghifary, M., Bastiaan Kleijn, W., Zhang, M., Balduzzi, D., 2015. Domain generalization for object recognition with multi-task autoencoders, in: Proceedings of the IEEE international conference on computer vision, pp. 2551–2559.

Gong, R., Chen, Y., Paudel, D.P., Li, Y., Chhatkuli, A., Li, W., Dai, D., Van Gool, L., 2021. Cluster, split, fuse, and update: Meta-learning for open compound domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8344–8354.

Guan, D., Huang, J., Lu, S., Xiao, A., 2021. Scale variance minimization for unsupervised domain adaptation in image segmentation. Pattern Recognition 112, 107764.

Gulrajani, I., Lopez-Paz, D., 2021. In search of lost domain generalization, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=lQdXeXDoWtI.

Li, D., Yang, Y., Song, Y.Z., Hospedales, T., 2017. Deeper, broader and artier domain generalization, in: International Conference on Computer Vision.

Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M., 2018a. Learning to generalize: Meta-learning for domain generalization, in: Thirty-Second AAAI Conference on Artificial Intelligence.

Li, H., Jialin Pan, S., Wang, S., Kot, A.C., 2018b. Domain generalization with adversarial feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5400–5409.

Li, J., Chen, E., Ding, Z., Zhu, L., Lu, K., Shen, H.T., 2020. Maximum density divergence for domain adaptation. IEEE transactions on pattern analysis and machine intelligence .

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D., 2018c. Deep domain generalization via conditional invariant adversarial networks, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 624–639.

Lin, J., 1991. Divergence measures based on the shannon entropy. IEEE Transactions on Information theory 37, 145–151.

Lipton, Z.C., Wang, Y.X., Smola, A., 2018. Detecting and correcting for label shift with black box predictors. arXiv preprint arXiv:1802.03916 .

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

Liu, F., Zhang, G., Lu, J., 2020b. Multi-source heterogeneous unsupervised domain adaptation via fuzzy-relation neural networks. IEEE Transactions on Fuzzy Systems .

Liu, H., Cao, Z., Long, M., Wang, J., 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, pp. 2927–2936.

Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A., 2021. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. arXiv preprint arXiv:2103.06030 .

Long, M., Cao, Y., Wang, J., Jordan, M., 2015. Learning transferable features with deep adaptation networks, in: International conference on machine learning, PMLR. pp. 97–105.

Long, M., Cao, Z., Wang, J., Philip, S.Y., 2017. Learning multiple tasks with multilinear relationship networks, in: Advances in neural information processing systems, pp. 1594–1603.

Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S., 2014. Transfer joint matching for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1410–1417.

Luo, Z., Zou, Y., Hoffman, J., Fei-Fei, L.F., 2017. Label efficient learning of transferable representations acrosss domains and tasks, in: Advances in Neural Information Processing Systems, pp. 165–177.

Mancini, M., 2020. Towards recognizing new semantic concepts in new visual domains. arXiv preprint arXiv:2012.09058 .

Mao, Y., Liu, W., Lin, X., 2020. Adaptive adversarial multi-task representation learning, in: International Conference on Machine Learning.

Matsuura, T., Harada, T., 2020. Domain generalization using a mixture of multiple latent domains, in: AAAI.

Maurer, A., Pontil, M., Romera-Paredes, B., 2013. Sparse coding for multitask and transfer learning, in: International Conference on Machine Learning, pp. 343–351.

Motiian, S., Jones, Q., Iranmanesh, S., Doretto, G., 2017. Few-shot adversarial domain adaptation, in: Advances in Neural Information Processing Systems, pp. 6670–6680.

Muandet, K., Balduzzi, D., Schölkopf, B., 2013. Domain generalization via invariant feature representation, in: International Conference on Machine Learning, pp. 10–18.

Panareda Busto, P., Gall, J., 2017. Open set domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 754–763.

Polyanskiy, Y., Wu, Y., 2019. Lecture notes on information theory .

Redko, I., Habrard, A., Sebban, M., 2017. Theoretical analysis of domain adaptation with optimal transport, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 737–753.

Redko, I., Morvant, E., Habrard, A., Sebban, M., Bennani, Y., 2020. A survey on domain adaptation theory. arXiv preprint arXiv:2004.11829 .

Sharifi-Noghabi, H., Asghari, H., Mehrasa, N., Ester, M., 2020. Domain generalization via semi-supervised meta learning. arXiv preprint arXiv:2009.12658 .

Shen, J., Qu, Y., Zhang, W., Yu, Y., 2018a. Wasserstein distance guided representation learning for domain adaptation, in: AAAI Conference on Artificial Intelligence.

Shen, J., Qu, Y., Zhang, W., Yu, Y., 2018b. Wasserstein distance guided representation learning for domain adaptation, in: Thirty-Second AAAI Conference on Artificial Intelligence.

Shu, Y., Cao, Z., Wang, C., Wang, J., Long, M., 2021. Open domain generalization with domain-augmented meta-learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9624–9633.

Shui, C., Abbasi, M., Robitaille, L.É., Wang, B., Gagné, C., 2019. A principled approach for learning task similarity in multitask learning. arXiv preprint arXiv:1903.09109 .

Shui, C., Chen, Q., Wen, J., Zhou, F., Gagné, C., Wang, B., 2020. Beyond $\mathcal{H}$ -divergence: Domain adaptation theory with jensen-shannon divergence. arXiv preprint arXiv:2007.15567 .

Shui, C., Wang, B., Gagné, C., 2022. On the benefits of representation regularization in invariance based domain generalization. Machine Learning , 1–21.

Torralba, A., Efros, A.A., 2011. Unbiased look at dataset bias, in: CVPR 2011, IEEE. pp. 1521–1528.

Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S., 2017. Deep hashing network for unsupervised domain adaptation, in: (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR).

Wang, M., Deng, W., 2018. Deep visual domain adaptation: A survey. Neurocomputing 312, 135–153.

Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R., 2019. Multi-similarity loss with general pair weighting for deep metric learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5022–5030.

Wen, J., Zheng, N., Yuan, J., Gong, Z., Chen, C., 2019. Bayesian uncertainty matching for unsupervised domain adaptation. arXiv preprint arXiv:1906.09693 .

Xie, S., Zheng, Z., Chen, L., Chen, C., 2018. Learning semantic representations for unsupervised domain adaptation, in: International Conference on Machine Learning, pp. 5423–5432.

Xu, Q., Zhang, R., Zhang, Y., Wang, Y., Tian, Q., 2021. A fourier-based framework for domain generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14383–14392.

Yan, S., Song, H., Li, N., Zou, L., Ren, L., 2020. Improve unsupervised domain adaptation with mixup training. arXiv preprint arXiv:2001.00677 .

Zhang, Y., Tang, H., Jia, K., Tan, M., 2019. Domain-symmetric networks for adversarial domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5031–5040.

Zhao, H., Combes, R.T.d., Zhang, K., Gordon, G.J., 2019. On learning invariant representation for domain adaptation. arXiv preprint arXiv:1901.09453 .

Zhao, S., Gong, M., Liu, T., Fu, H., Tao, D., 2020. Domain generalization via entropy regularization. Advances in Neural Information Processing Systems 33.

Zhou, F., Chaib-draa, B., Wang, B., 2021a. Multi-task learning by leveraging the semantic information. Proceedings of the AAAI Conference on Artificial Intelligence 35, 11088–11096. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17323.

Zhou, F., Jiang, Z., Shui, C., Wang, B., Chaib-draa, B., 2021b. Domain generalization via optimal transport with metric similarity learning. Neurocomputing URL: https://www.sciencedirect.com/science/article/pii/S0925231221002009, doi:https://doi.org/10.1016/j.neucom.2020.09.091.

Zhou, F., Shui, C., Abbasi, M., Robitaille, L.É., Wang, B., Gagné, C., 2021c. Task similarity estimation through adversarial multitask neural network. IEEE Transactions on Neural Networks and Learning Systems 32, 466–480. doi:10.1109/TNNLS.2020.3028022.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C., 2021d. Domain generalization: A survey. arXiv preprint arXiv:2103.02503 .

Zhou, K., Yang, Y., Hospedales, T., Xiang, T., 2020a. Deep domain-adversarial image generation for domain generalisation, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 13025–13032.

Zhou, K., Yang, Y., Hospedales, T., Xiang, T., 2020b. Learning to generate novel domains for domain generalization, in: European Conference on Computer Vision, Springer. pp. 561–578.

Zhou, K., Yang, Y., Qiao, Y., Xiang, T., 2021e. Domain generalization with mixstyle. arXiv preprint arXiv:2104.02008 .