DIRECT-EFFECT RISK MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of out-of-distribution (o.o.d.) generalization where spurious correlations of attributes vary across training and test domains. This is known as the problem of *correlation shift* and has posed concerns on the reliability of machine learning. In this work, we introduce the concepts of *direct* and *indirect effects* from causal inference to the domain generalization problem. Under mild conditions, we show that models that learn direct effects provably minimize the worst-case risk across correlation-shifted domains. To eliminate the indirect effects, our algorithm consists of two stages: in the first stage, we learn an indirect-effect representation by minimizing the prediction error of domain labels using the representation and the class label; in the second stage, we remove the indirect effects learned in the first stage by matching each data with another data of similar indirect-effect representation but of different class label. Experiments on 5 correlation-shifted datasets and the DomainBed benchmark verify the effectiveness of our approach.

1 INTRODUCTION

Machine learning has achieved huge success in many fields, yet they mostly rely on the independent and identically distributed (i.i.d) assumption. When it comes to an out-of-distribution (o.o.d.) test domains, machine learning models usually suffer from a sharp performance drop (Beery et al., 2018; Arjovsky et al., 2019; Nagarajan et al., 2020). The o.o.d. data typically come in the form of *correlation shift*, where spurious correlations of attributes vary between training and test domains, or *diversity shift*, where the shifted test distribution keeps the semantic content of the data unchanged while altering the data style. The focus of this work is on the former setting known as correlation shift. That is, given stable causality and spurious correlations between attributes, how to disentangle the stable causality and the spurious correlations from the training data. Figure 1 shows the performance gain of our method on the correlation shift datasets.

Much effort has been devoted to learning representations that are invariant across training environments, where many works have introduced the tools from causality to address the o.o.d. generalization problems. When the data are of high dimension and multiple attributes are entangled, it is challenging to identify invariant causality across domains. Many methods have been designed to resolve the issue. Representative methods include incorporating invariance constraints by designing new loss functions (Arjovsky et al., 2019; Krueger et al., 2021; Bellot & van der Schaar, 2020), learning latent semantic features in causal graphs by VAE (Liu et al., 2021; Lu et al., 2021), and eliminating selection bias by matching (Mahajan et al., 2021; Wang et al., 2022). However, these methods, despite their theoretical guarantees, fail to show empirical improvement over Empirical Risk Minimization (ERM) as verified by the DomainBed benchmark (Gulrajani & Lopez-Paz, 2020; Vedantam et al., 2021).

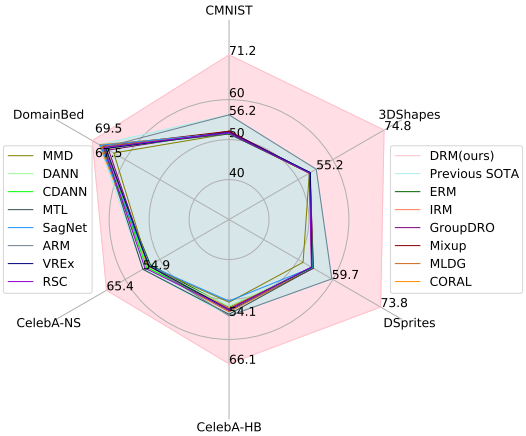


Figure 1: Test accuracy of o.o.d. algorithms on 5 correlation-shifted datasets and the DomainBed benchmark (avg). The pink region represents the performance of our method, while the light blue region represents the previously best-known results (implemented by DomainBed using *training-domain validation*) on each dataset.

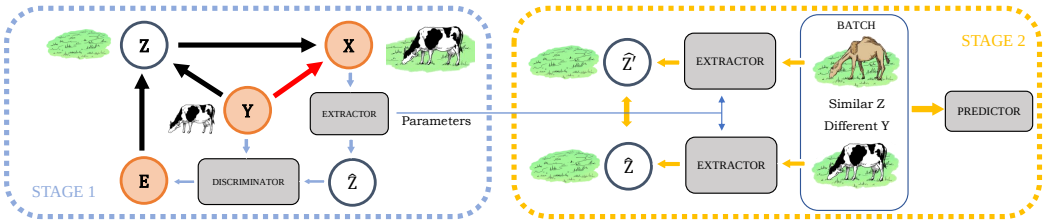


Figure 2: Description of our two-stage approach. In Stage 1, we jointly learn a discriminator and an (indirect-effect) extractor by predicting the domain labels. In Stage 2, the extractor in Stage 1 is used to construct a balanced batch of samples with a similar indirect-effect representation but different class labels, and a predictor is trained on the balanced batch to predict the class labels. The red and black arrows form the graphical model of the correlation shift problem (data generation process).

This paper is the first attempt to use the tool of *direct* and *indirect effects* from causal inference to analyze the correlation shift problem. We show that under certain conditions, models that learn direct effects minimize the worst-case risk across domain-shifted domains. To learn the direct effects, we propose a two-stage approach: in the first stage, we use an extractor to infer the indirect-effect representation Z from the data X such that Z can predict the domain label E through a discriminator head (see the blue box in Figure 2). In the second stage, we construct a *balanced batch* by augmenting the original batch with data of the same indirect-effect representation Z but of a different class label Y. We test our approach on the DomainBed benchmark. On the correlation shift dataset Colored MNIST, our model obtains an average accuracy of 71.2% over three domain generalization problems. While the information-theoretic best accuracy on the Colored MNIST dataset is 75%, our method achieves an accuracy as high as 69.7% on the most difficult "-90%" environment. Moreover, the results of our model on the diversity shift datasets are comparable to the state-of-the-art. Our main contributions are as follows:

- Theoretically, we present a framework to analyze the correlation shift problem based on direct/indirect causal effects. We demonstrate that under mild conditions, models learning direct effects provably minimize the worst-case risk across correlation-shifted domains.
- Algorithmically, inspired by our theoretical analysis, we propose a new two-stage approach to improve o.o.d. generalization. The algorithm consists of two stages: in the first stage, it learns an indirect-effect representation by minimizing the prediction error of domain label using the representation and the class label; in the second stage, the method constructs a balanced batch by augmenting the original batch with data of the same indirect-effect representation but of a different class label.
- Experimentally, our method outperforms baselines by a large margin on the correlation-shifted datasets. For example, on the Colored MNIST dataset, our approach achieves up to 15% absolute improvement over the state-of-the-art in terms of average accuracy over three domains. On the CelebA datasets, our algorithm achieves up to 11% absolute improvement over the state-of-the-art in terms of average accuracy over three domains.

2 Preliminaries

Notations. In this paper, we will use *capital* letters such as X, Y and Z to represent random variables, *lower-case* letters such as x and z to represent realization of random variables, and letters with *hat* such as $\hat{Z}$ to represent inferred variables by the model. We use the *calligraphic capital* letter $\mathcal{E}$ to represent the set of environments, and by *lower-case* letter e the domain label. $X \perp\!\!\!\perp Y$ means that random variables X and Y are independent. We use $\mathbb{P}^e$ to denote the distribution of variables on environment e, and use $\mathbb{P}^e_B$ to denote its corresponding *balanced distribution* (see Definition 4). We add the superscript e to a variable such as x^e to indicate that the variable is sampled from the distribution of the environment e, and (x^e_i, y^e_i, e) refers to an instance sampled from $\mathbb{P}^e$. We denote by $\mathcal{H}$ the hypothesis class of models, and by $h : \mathcal{X} \to \mathcal{Y}$ the predictor. $R^e(h)$ refers to the risk of predictor h on environment e. *Environment* and *domain* are of the same concept, and we use them interchangeably throughout the paper.

Direct effect and indirect effect. Direct and indirect effects are important concepts in causal inference. Their formal definitions were given in (Pearl, 2001):

Definition 1 (Definitions 5 and 7 of Pearl (2001), Average Natural Direct (NDE) and Indirect Effects (NIE)). *The average natural direct effect and indirect effect of event $Y = y$ on a response variable X w.r.t. the reference point y^* are defined as*

$$\textit{(Direct effect)} \quad NDE(y, y^*; X) = \mathbb{E}(X|do(Y = y), do(Z = Z_{y^*})) - \mathbb{E}(X|do(Y = y^*)),$$

$$\textit{(Indirect effect)} \quad NIE(y, y^*; X) = \mathbb{E}(X|do(Y = y^*), do(Z = Z_y)) - \mathbb{E}(X|do(Y = y^*)),$$

where Z stands for all parents of X except Y, and Z_y denotes the value of Z when $Y = y$. The do-operation assigns a value to a variable while ensuring that all variables except its descendants are not changed. The relationship between variables is also not changed by the do-operation.

Definition 1 states that, if Y changes from y^* to y and Z is fixed, then the change in X is the direct effect of Y on X. Similarly, fixing Y but changing Z from the value taken at $Y = y^*$ to the value taken at $Y = y$, the change in X is the indirect effect of Y on X. We define the total effect of the change in Y on X as follows:

Definition 2 (Theorem 3 of Pearl (2001), Total Effect). *The total effect of event $Y = y$ on variable X w.r.t. the reference point y^* is defined as $TE(y, y^*; X) = \mathbb{E}(X_y - X_{y^*}) = NIE(y, y^*; X) + NDE(y, y^*; X)$, where the second equality holds when $NDE(y, y^*; X) = -NDE(y^*, y; X)$.*

Data generation process. In the correlation shift problem, the process of generating the data X can be depicted by Figure 2. Consider a binary classification problem of cows and camels. We assume that the animal category and background are the two attributes that contribute to the generation of an image. Our goal is to predict the animal category Y from image X, and the background is denoted by Z. The image X is the result of the total effect of the two attributes. We assume that the value of Y is changed from "cow" to "camel" during the data generation process. So the animal in the image X is changed. Meanwhile, the cow is more likely to be on the grass, while the camel is more likely to be in the desert. Therefore, Z may change from "grass" to "desert" as Y changes, changing the background in the image X.

Label inference process. Figure 3 shows the label inference process, an inverse process of the data generation process. The model can either infer labels through a direct-effect pathway $X \to Y$ (based on the animal in the image) or through an indirect-effect pathway $X \to Z \to Y$ (based on the background in the image). The latter is not stable because the relationship between Y and Z may change as the domain changes. Thus cutting off the correlation between Y and Z during training would force the model to learn the direct effects and significantly improve domain generalization performance.

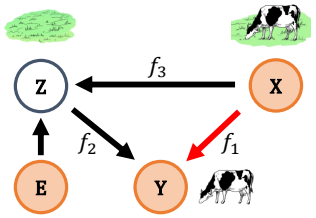


Figure 3: The causal graph of label inference process.

Why is our causal graph general? A general causal graph of correlation shift can reduce to our causal graph. Consider a general Directed Acyclic Graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, where $\mathbf{V}$ is the set of vertexs and $\mathcal{E}$ is the set of edges. Vertexs in this graph include input X and label Y. The indirect pathways between X and Y can be divided into three categories. For pathway category $P_1 : X \to \mathbf{V}_1 \leftarrow Y$, $\mathbf{V}_1$ is unobservable so it does not affect the correlation between X and Y. Thus this kind of pathway is not in our consideration. The pathway category $P_2 : X \leftarrow \mathbf{V}_2 \to Y$ and $P_3 : X \to \mathbf{V}_3 \to Y$ are unstable because $\mathbf{V}_2$ and $\mathbf{V}_3$ may be disturbed by different environments. We merge vertexs in $\mathbf{V}_2$ and $\mathbf{V}_3$ into the vertex Z in our causal graph and introduce environment vertex E to control this unstable pathway.

3 Domain Generalization Benefits from Direct Effects

We consider a standard domain generalization setting, where the data come from different environments $e \in \mathcal{E}_{all}$. Assume that we have the training data collected from a finite subset of training environments $\mathcal{E}_{train}$, where $\mathcal{E}_{train} \subset \mathcal{E}_{all}$. For every environment $e \in \mathcal{E}_{train}$, the training dataset $D^e = \{(x_i^e, y_i^e, e)\}_{i=1}^{N_e}$ is sampled from the distribution $\mathbb{P}^e(X^e, Y^e) = \mathbb{P}(X, Y \mid E = e)$, where X is the instance (e.g., an image), Y is the class label, E is the domain label, and N_e is the number of training data in environment e. The goal of domain generalization is to train a model with data from training environments $\mathcal{E}_{train}$ that generalizes well to all environments $e \in \mathcal{E}_{all}$. Our goal is to find a predictor $h^* : \mathcal{X} \to \mathcal{Y}$ in the hypothesis class $\mathcal{H}$ such that the worst-case risk is minimized:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \max_{e \in \mathcal{E}_{\text{all}}} R^e(h), \tag{1}$$

where $R^e(h)$ is the risk of predictor h in environment e. In this paper, we consider the following correlation shift model (see Figure 3):

Definition 3 (Correlation Shift Model)**.** *For any environment e, the class label Y^e can be inferred by a model through two pathways, $X \xrightarrow{f_1} Y$ and $X \xrightarrow{f_3} Z \xrightarrow{f_2} Y$. The correlation shift model is defined as follows:*

$$Y^e = f_1(X^e) + f_2(Z^e) + \varepsilon_1, \quad \mathbb{E}\left[\varepsilon_1\right] = 0, \quad \varepsilon_1 \perp\!\!\!\perp X^e, \quad \varepsilon_1 \perp\!\!\!\perp Z^e, \tag{2}$$

where ε_1 is a random noise. The first pathway f_1 is invariant across all environments, while the second is affected by the environment variable E through $Z^e = f_3(X^e, e)$. We assume that $\forall f : \mathcal{X} \to \mathcal{Y}$, $\exists\, e \in \mathcal{E}_{all}$ such that $f_2\left(f_3(x, e)\right) = f(x)$.

The correlation shift model claims that f_1 is the reversed mapping of the direct causal effect, while $f_2 \circ f_3$ is the reversed mapping of the indirect causal effect in Definition 1. Equation 2 then follows from Definition 2. Definition 3 states that varying e makes the composition $f_2 \circ f_3$ a rich function class to express arbitrary function. Thus, the indirect-effect pathway $X \to Z \to Y$ is unstable w.r.t. the varying e's. For example, in the classification problem of cows and camels, f_3 extracts the background Z from an image X and f_2 predicts the class label Y according to the background Z. Definition 2 implies that there exists at least a test environment where $f_2 \circ f_3$ differs significantly from that of the training environment. That is, the correlation between the background and the class label may be reversed: most backgrounds of cow images are desert while most backgrounds of camel images are grass. The assumption makes the domain generalization problem non-trivial; otherwise, ERM is optimal when the test and training environments have the same correlation between the background and the class label.

Theorem 1. *The predictor $h^* = f_1$ that makes use of direct causal effect achieves the global minimum of Equation 1.*

We defer the proof to the appendix. To enable the model to learn the direct effects in the data, it is desirable to cut off the pathway between Z and Y so that they are independent. To this end, we consider a balanced distribution over each domain, defined as follows:

Definition 4 (Balanced Distribution)**.** *The balanced distribution of $\mathbb{P}^e(X, Y, Z)$ is defined as $\mathbb{P}^e_B(X, Y, Z) = \mathbb{P}^e(X, Y | do(Z = z))\mathbb{P}^e(Z = z) = \mathbb{P}^e(X|Y, Z)\mathbb{P}^e(Z)\mathbb{P}^e(Y)$, where $\mathbb{P}^e(Y)$ is the marginal distribution of $\mathbb{P}^e(X, Y, Z)$, $Y \perp\!\!\!\perp Z$. The do-operation is defined in Definition 1.*

The balanced distribution shares the same marginal distributions of Z and Y with the original distribution. But the marginal distribution of X is different. We make the following assumptions about the change in the marginal distribution of X.

Assumption 1. *Consider any training environments $e^i_S \in \mathcal{E}_{train}$ and the test environment e_T with support $\mathcal{X}$, with distributions $\mathbb{P}^{e^i_S}$ and $\mathbb{P}^{e_T}$, respectively. Let $\mathbb{P}^{e^i_T}_B$ be the corresponding balanced distribution of $\mathbb{P}^{e^i_T}$. $\mathcal{X}$ can be divided into two subsets $\mathcal{X}_1$ and $\mathcal{X}_2$, where $\mathcal{X}_1 = \left\{X : \mathbb{P}^{e_S}(X) > \mathbb{P}^{e^i_T}(X)\right\}$ and $\mathcal{X}_2 = \left\{X : \mathbb{P}^{e_S}(X) \leq \mathbb{P}^{e^i_T}(X)\right\}$. We then assume that $\forall X \in \mathcal{X}_1$, $\mathbb{P}^{e_S}_B(X) \leq \mathbb{P}^{e_S}(X)$, and $\forall X \in \mathcal{X}_2$, $\mathbb{P}^{e_S}_B(X) \leq \mathbb{P}^{e^i_T}(X)$.*

This assumption states that the probability density of X in the balanced distribution will not exceed the maximum of that of the training and test distributions. For example, on the CMNIST dataset, we observe that Assumption 1 indeed holds true: if the training domain is the "$+90\%$" domain and the test domain is the "-90%" domain, the probability of red images with label 1 on the training distribution is greater than that on the balanced distribution and the probability of green images with label 1 on the test distribution is greater than that on the balanced distribution. This is consistent with Assumption 1. With that, we have the following theorem.

Theorem 2. *Let $\mathcal{H}$ be a hypothesis space of VC-dimension d, and denote by $d_{\mathcal{H}}(\mathbb{P}^{e^i_S}, \mathbb{P}^{e_T}) := 2\sup_{\eta \in \mathcal{H}} |\Pr_{x \sim \mathbb{P}^{e_T}}[\eta(x) = 1] - \Pr_{x \sim \mathbb{P}^{e^i_S}}[\eta(x) = 1]|$ the $\mathcal{H}$-divergence between $\mathbb{P}^{e^i_S}$ and $\mathbb{P}^{e_T}$. Assume that we have N_S training environments $\mathcal{E}_{train} = \left\{e^i_S \mid i = 1, 2, \ldots, N_S\right\}$. We uniformly draw i.i.d. samples of size m from the balanced distribution of N_S training environments $\{\mathbb{P}^{e^i_S}_B \mid i = 1, 2, \ldots, N_S\}$. Suppose that $\inf_{h \in \mathcal{H}}[R^{e_T}(h) + \frac{1}{N_S}\sum_{i=1}^{N_S} \hat{R}^{e^i_S}_B(h)] \leq \lambda$, $\max_i d_{\mathcal{H}}(\mathbb{P}^{e^i_S}, \mathbb{P}^{e_T}) = \epsilon$, and Assumption 1 holds. Then with probability at least $1 - \delta$, for every $h \in \mathcal{H}$, we have $R^{e_T}(h) \leq \frac{1}{N_S}\sum_{i=1}^{N_S} \hat{R}^{e^i_S}_B(h) + \sqrt{\frac{4}{m}(d \log \frac{2em}{d} + \log \frac{4}{\delta})} + \epsilon + \lambda$, where $\hat{R}^{e^i_S}_B(h)$ is the empirical risk of h on the balanced distribution of training environment e^i_S.*

4 ALGORITHMIC DESIGN FOR DOMAIN GENERALIZATION

Inspired by our theoretical analysis, we propose a two-stage approach for domain generalization. In the first stage, we extract the indirect-effect representation Z by learning a discriminator head to predict the domain labels. In the second stage, we create a balanced batch to cut off the pathway between the representation Z and the label Y, on which we train our model. However, implementation of the procedure is challenging: 1) Z is not observable and needs to be recovered from X, Y and E; 2) a perfectly balanced batch may not exist; 3) creating a balanced distribution by the matching method would naturally change the distribution of Y in the batch and conflict Definition 4. To overcome these challenges, we describe our approach in the following two sections.

4.1 RECOVERING THE INDIRECT EFFECTS

Since the variable Z on the indirect-effect pathway is not observable, we extract a representation $\hat{Z}$ of the indirect effect from X by learning a discriminator head in the first stage. From Figure 2, we observe that given the indirect-effect representation Z and the class label Y, the other variables are independent of the domain label E. Hence the discriminator head is able to extract $\hat{Z}$ from X to predict the domain label E. Specifically, assume that the dataset is sampled from N_S training domains. We set up an extractor $G(\cdot;\Theta_G) : \mathcal{X} \to \mathcal{Z}$ and a discriminator head $D(\cdot,\cdot;\Theta_D) : \mathcal{Z} \times \mathcal{Y} \to [0,1]^{N_S}$ that outputs the probability that a sample belongs to each training domain, and update the parameters of both models by minimizing the prediction error of domain label e: $\Theta_G^*, \Theta_D^* := \operatorname{argmin}_{\Theta_G,\Theta_D} \mathbb{E}_{x,y,e}\text{CrossEntropy}(D(G(x;\Theta_G), y; \Theta_D), e)$, where Θ_G and Θ_D stand for the parameters of the extractor G and the discriminator head D, respectively, and (x, y, e) is a training sample. We use the learned extractor to obtain the representation $\hat{z}_i^e = G(x_i^e; \Theta_G^*)$ for every instance (x_i^e, y_i^e, e).

Many methods learned domain discriminators by a minimax problem (Ganin et al., 2016; Li et al., 2018b; Albuquerque et al., 2019). These methods extracted features that could maximize the domain discriminator error. In our approach, on the other hand, the representation vector Z is obtained by minimizing the domain discrimination error. This makes our model easier to optimize and more stable than a minimax game.

4.2 ELIMINATING THE INDIRECT EFFECTS

In the second stage, we remove the indirect effects from the data based on the representation $\hat{Z}$. We start by defining the balanced batch.

Definition 5 (Balanced Batch)**.** *For any sample $(x_i^e, y_i^e, e, \hat{z}_i^e)$ in a balanced batch, there exists a corresponding sample $(x_j^e, y_j^e, e, \hat{z}_j^e)$ with probability P, such that $\hat{z}_i^e = \hat{z}_j^e$, $y_i^e \neq y_j^e$, and $\mathbb{P}_{Batch}(Y) = \mathbb{P}_D(Y)$, where $\mathbb{P}_{Batch}(Y)$ and $\mathbb{P}_D(Y)$ are marginal distributions of Y in the batch and in the training set, respectively.*

Ideally, Y follows a uniform distribution in the training set, and for each sample x_i, we can find a corresponding sample x_j with the same indirect-effect representation $\hat{z}_i^e = \hat{z}_j^e$. However, the above approach does not necessarily create a balanced batch. There are two reasons: 1) we cannot always find exactly equal $\hat{z}$ as in the ideal case; 2) if the label Y in the training set is not uniformly distributed, the procedure mentioned above might change the distribution of Y in the batch. This is inconsistent with Definition 5.

To resolve the first problem, for each sample $(x_i^e, y_i^e, e, \hat{z}_i^e)$, we search for another sample $(x_j^e, y_j^e, e, \hat{z}_j^e)$ such that $\hat{z}_j^e$ is the nearest neighbor of $\hat{z}_i^e$. As for the second problem, we include the matched sample into the batch with a probability that depends on the proportion of each class of samples in the training set. Qualitatively, we match each sample with a small number of samples for the majority class and a large number of samples for the minority class. The following theorem gives exact description of such probabilities.

Theorem 3. *Suppose Y has m classes. Let the proportion of each class of sample in the training set be $\omega = (\omega_1, \omega_2, \ldots, \omega_m)$. For any sample $(x_i^e, y_i^e, e, \hat{z}_i^e)$ in a batch, if it belongs to class k and we include its matched sample $(x_j^e, y_j^e, e, \hat{z}_j^e)$ into the batch with probability $\frac{(1-\omega_k)/\omega_k}{\max_i(1-\omega_i)/\omega_i}$, then we have $\forall k \in \{1, 2, \ldots, m\}, \mathbb{E}(\mathrm{N}_{\mathrm{Batch}}(Y = k)) \propto \omega_k$, where $\mathrm{N}_{\mathrm{Batch}}(Y = k)$ is the number of samples belonging to class k in the balanced batch.*

Algorithm 1 Direct-Effect Risk Minimization (DRM)

Input: Dataset D; initial predictor f_{θ_0}; training set class distribution $\omega = (\omega_1, \omega_2, \ldots, \omega_m)$; training steps T; learning rate ϵ;
Output: Predictor f_{θ_T};

$\Theta_G^*, \Theta_D^* \leftarrow \text{argmin}_{\Theta_G, \Theta_D} \mathbb{E}_{x,y,e}\text{CrossEntropy}(D(G(x;\Theta_G), y; \Theta_D), e)$;
$t \leftarrow 0$;
while $t \leq T$ **do**
  Sample a batch $\{(x_i^e, y_i^e)\}_{i=1}^{\text{batchsize}}$ from D; $B \leftarrow \{\}$;
  for (x^e, y^e) in batch **do**
    $\hat{z}^e \leftarrow G(x^e; \Theta_G^*)$;
    Search for the sample $(x^{e\prime}, y^{e\prime})$ with the closest $\hat{z}^{e\prime}$ (Euclidean distance) to $\hat{z}^e$;
    Add $(x^{e\prime}, y^{e\prime})$ to B with probability $\frac{(1-\omega_{y^e})/\omega_{y^e}}{\max_i(1-\omega_i)/\omega_i}$;
  end for
  Add B to the original batch $\{(x_i^e, y_i^e)\}_{i=1}^{batchsize}$ to form a new batch;
  Run ERM or other algorithms on the new batch and update f_{θ_t};
  $t \leftarrow t + 1$;
end while

Algorithm 1 describes our algorithm Direct-Effect Risk Minimization (DRM): in a balanced batch, each sample pair has the same indirect effect Z but different labels Y's. So when the balanced batch is generated, the difference in X all comes from the difference in Y, which is transmitted through the direct-effect pathway $Y \to X$. Since the indirect effects are eliminated, we force the model to learn the direct effects between Y and X.

5 EXPERIMENTS

We compare DRM with 14 baseline methods, including: ERM (Vapnik, 1998), IRM (Arjovsky et al., 2019), GroupDRO (Sagawa et al., 2019), Mixup (Zhang et al., 2018; Xu et al., 2020; Yan et al., 2020; Wang et al., 2020), MLDG (Li et al., 2018a), CORAL (Sun & Saenko, 2016), MMD (Li et al., 2018c), DANN (Ganin et al., 2016), CDANN (Li et al., 2018d), MTL (Blanchard et al., 2011), SagNet (Nam et al., 2019), ARM (Zhang et al., 2020), VREx (Krueger et al., 2021) and RSC (Huang et al., 2020), which appeared in the DomainBed benchmark (Gulrajani & Lopez-Paz, 2020). We strictly follow the protocol of DomainBed by conducting random searches for all hyperparameters in all stages. Following Ye et al. (2022), we divide the datasets into two categories: datasets dominated by *correlation shift* and datasets dominated by *diversity shift*. We evaluate the performance of our approach on both datasets. We choose the *training-domain validation method* in DomainBed as our model selection method. We also balance the validation set using the method in Section 4.2.

5.1 CORRELATION SHIFT

In the correlation-shifted datasets, there is spurious correlation between the class label and the features such as the color or the background of the images. Definition 3 indicates that learning the test environment is difficult since there might be correlation flip between the training and test environments. We show that the performance of i.i.d algorithms such as ERM will significantly drop in this case, while our approach achieves improved performance in these difficult environments. We evaluate our approach on the correlation-shifted dataset CMNIST from DomainBed. Moreover, in order to perform a comprehensive evaluation, we also conduct experiments on the 3DShapes dataset, DSprites dataset, and the CelebA dataset, which are common correlation shift datasets used to evaluate domain generalization methods (Wiles et al., 2021; Ye et al., 2022). We use the DomainBed benchmark to evaluate algorithms on these datasets. For comparison, we run the above mentioned 14 methods on these datasets using the codes provided by DomainBed.

Colored MNIST (Arjovsky et al., 2019). Colored MNIST is a handwritten digit classification dataset (LeCun, 1998). It creates spurious correlation between colors and digits by artificially coloring the digits by red or green. The correlations between the color and the label in three environments are $+90\%$, $+80\%$ and -90%, respectively. For example, in the "$+90\%$" environment, 90% images with label 1 are dyed red, while 90% images with label 0 are dyed green. In addition, the dataset

Table 1: Experimental results on the correlation-shifted datasets, where the experiments are run by following the DomainBed setting.

Algorithm	CMNIST		3DShapes		DSprites		CelebA-HB		CelebA-NS	
	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg
ERM	10.0 ± 0.1	51.5 ± 0.1	10.1 ± 0.1	53.3 ± 0.1	13.8 ± 0.5	54.0 ± 0.1	16.8 ± 1.2	52.0 ± 0.5	21.1 ± 0.4	52.7 ± 0.5
IRM	10.2 ± 0.3	52.0 ± 0.1	10.0 ± 0.0	53.2 ± 0.1	14.5 ± 0.3	54.0 ± 0.1	20.4 ± 2.1	52.1 ± 0.7	21.5 ± 0.9	53.2 ± 0.4
GroupDRO	10.0 ± 0.2	52.1 ± 0.0	10.5 ± 0.4	53.4 ± 0.1	15.0 ± 0.4	54.4 ± 0.2	18.3 ± 1.5	52.8 ± 0.9	21.2 ± 0.2	53.3 ± 0.2
Mixup	10.1 ± 0.1	52.1 ± 0.2	10.2 ± 0.1	53.4 ± 0.2	14.0 ± 0.3	53.9 ± 0.0	17.9 ± 3.4	52.4 ± 1.1	22.2 ± 1.5	53.7 ± 0.7
MLDG	9.8 ± 0.1	51.5 ± 0.1	10.1 ± 0.1	53.5 ± 0.1	14.3 ± 0.3	54.2 ± 0.1	20.0 ± 2.1	53.0 ± 0.7	22.7 ± 1.7	53.7 ± 0.6
CORAL	9.9 ± 0.1	51.5 ± 0.1	10.0 ± 0.0	53.3 ± 0.1	13.8 ± 0.2	53.9 ± 0.2	17.7 ± 1.6	52.4 ± 0.6	22.1 ± 1.1	53.4 ± 0.4
MMD	9.9 ± 0.3	51.5 ± 0.2	10.0 ± 0.1	53.2 ± 0.1	14.4 ± 0.0	51.4 ± 2.1	17.4 ± 1.8	50.7 ± 0.5	22.5 ± 0.6	53.3 ± 0.1
DANN	10.0 ± 0.0	51.5 ± 0.3	10.0 ± 0.0	53.3 ± 0.0	14.7 ± 0.3	54.1 ± 0.3	16.9 ± 1.7	51.7 ± 0.3	21.8 ± 1.5	53.7 ± 0.8
CDANN	10.2 ± 0.1	51.7 ± 0.1	10.0 ± 0.0	53.3 ± 0.1	14.4 ± 0.2	54.0 ± 0.1	18.6 ± 2.6	52.5 ± 0.6	22.5 ± 1.2	53.9 ± 0.4
MTL	10.5 ± 0.1	51.4 ± 0.1	10.1 ± 0.0	53.4 ± 0.1	14.8 ± 0.5	54.3 ± 0.1	23.5 ± 1.4	53.7 ± 0.6	27.6 ± 1.2	54.9 ± 0.3
SagNet	10.3 ± 0.1	51.7 ± 0.0	10.1 ± 0.1	53.4 ± 0.1	13.6 ± 0.1	54.0 ± 0.0	14.9 ± 0.9	50.4 ± 0.3	22.0 ± 0.6	53.1 ± 0.2
ARM	10.2 ± 0.0	56.2 ± 0.2	10.0 ± 0.0	55.2 ± 0.3	14.5 ± 0.6	59.7 ± 0.4	22.8 ± 2.3	54.1 ± 0.6	21.1 ± 1.4	53.0 ± 0.5
VREx	10.2 ± 0.0	51.8 ± 0.1	10.8 ± 0.3	53.5 ± 0.1	13.8 ± 0.3	53.9 ± 0.1	19.2 ± 1.9	52.5 ± 0.7	20.3 ± 0.4	53.2 ± 0.3
RSC	10.0 ± 0.2	51.7 ± 0.2	10.1 ± 0.1	53.2 ± 0.1	13.3 ± 0.2	53.8 ± 0.1	18.9 ± 1.1	52.5 ± 0.5	23.7 ± 0.8	54.3 ± 0.5
DRM(ours)	**69.7** ± 1.5	**71.2** ± 0.6	**74.5** ± 0.2	**74.8** ± 0.1	**73.3** ± 0.5	**73.8** ± 0.2	**61.0** ± 4.9	**66.1** ± 0.6	**59.9** ± 2.6	**65.4** ± 1.2

randomly flips 25% of the class labels, which results in 75% correlation between shape and digital labels, lower than that between color and labels. Thus, an i.i.d. learning approach like ERM prefers to learn correlations between colors and labels.

3DShapes (Burgess & Kim, 2018). To demonstrate that our approach can eliminate spurious correlations between attributes, we run our algorithm on the 3DShapes dataset. The 3DShapes is a dataset with six attributes, among which we choose the "floor hue" and "orientation" to form the spurious correlation. Specifically, we divide the orientation of the graph into two categories. Our goal is to predict which category the orientation belongs to. We use the same construction as Colored MNIST to build three environments and add label noise.

DSprites (Matthey et al., 2017). We also evaluate our DRM algorithm on the DSprites dataset, which has six attributes. In this paper, "Position X" and "Position Y" are chosen to form spurious correlations in the DSprites. We argue that these two attributes are similar, and thus it is challenging to identify the invariant features across all environments.

CelebA-HB and CelebA-NS (Liu et al., 2015). We introduce the CelebA dataset to test the performance of our approach. CelebA is a large-scale face attribute dataset with 40 attribute annotations, e.g., eyeglasses, wearing hat and bangs. Any two attributes can form a correlation shift dataset, one of which acts as the label to be predicted and the other is used to create spurious correlations. In this paper, for *CelebA-HB*, "No Beard" is the label and there exists spurious correlation between the attribute "No Beard" and the attribute "Wearing Hat". For *CelebA-NS*, "Smiling" is the label and the correlation between "Wearing Necktie" and "Smiling" is unstable. Unlike above mentioned datasets, correlation shift on CelebA comes from non-random sampling, which is called selection bias in causal inference.

Results. Table 1 shows the performance of our approach under correlation shift. Under the DomainBed protocol, both ERM and the domain generalization algorithms officially reported by DomainBed do not perform well for correlation shift because they all suffer from a sharp performance drop when the test environment has reversed correlation with the training environment. For Colored MNIST, on which the information-theoretic best accuracy is 75% due to the 25% noise, the accuracy of ERM and other domain generalization algorithms are no more than 10.5% on the most difficult "-90%" environment, which is far lower than random guess. In contrast, our DRM approach achieves 69.7% accuracy, almost 60% higher than the other algorithms. At the same time, our approach does not hurt performance on the "$+90\%$" and "$+80\%$" domains. On average, our approach outperforms ERM by 20% and outperforms the best previous approach by 15%. For the other datasets, the results show the same pattern. Our approach is substantially ahead of the other methods by about 50% on the most difficult domain and bring significant performance improvement of more than 15% for the correlation shift problems.

5.2 DIVERSITY SHIFT

For the diversity shift datasets, the support sets of data on different environments have no overlap, and the test distribution keeps the semantic content of the data unchanged while altering the data style. For instance, in the PACS dataset, the training environment and the test environment can be

Table 2: Experimental results on the DomainBed benchmark.

DomainBed	Algorithm	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg
Official report	ERM	51.5 ± 0.1	98.0 ± 0.0	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	66.6
	IRM	52.0 ± 0.1	97.7 ± 0.1	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	65.4
	GroupDRO	52.1 ± 0.0	98.0 ± 0.0	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	64.8
	Mixup	52.1 ± 0.2	98.0 ± 0.1	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	66.7
	MLDG	51.5 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	66.7
	CORAL	51.5 ± 0.1	98.0 ± 0.1	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	67.5
	MMD	51.5 ± 0.2	97.9 ± 0.0	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	63.3
	DANN	51.5 ± 0.3	97.8 ± 0.1	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1	66.1
	CDANN	51.7 ± 0.1	97.9 ± 0.1	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	45.8 ± 1.6	38.3 ± 0.3	65.6
	MTL	51.4 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.6 ± 0.5	66.4 ± 0.5	45.6 ± 1.2	40.6 ± 0.1	66.2
	SagNet	51.7 ± 0.0	98.0 ± 0.0	77.8 ± 0.5	86.3 ± 0.2	68.1 ± 0.1	48.6 ± 1.0	40.3 ± 0.1	67.2
	ARM	56.2 ± 0.2	98.2 ± 0.1	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	66.1
	VREx	51.8 ± 0.1	97.9 ± 0.1	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9	65.6
	RSC	51.7 ± 0.2	97.6 ± 0.1	77.1 ± 0.5	85.2 ± 0.9	65.5 ± 0.9	46.6 ± 1.0	38.9 ± 0.5	66.1
Codes by authors	Fish	51.6 ± 0.1	98.0 ± 0.0	77.8 ± 0.3	85.5 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	42.7 ± 0.2	67.1
	Fishr	52.0 ± 0.2	97.8 ± 0.0	77.8 ± 0.1	85.5 ± 0.4	67.8 ± 0.1	47.4 ± 1.6	41.7 ± 0.0	67.1
	ANDmask	51.3 ± 0.2	97.6 ± 0.1	78.1 ± 0.9	84.4 ± 0.9	65.6 ± 0.4	44.6 ± 0.3	37.2 ± 0.6	65.5
	SANDmask	51.8 ± 0.2	97.4 ± 0.1	77.4 ± 0.2	84.6 ± 0.9	65.8 ± 0.4	42.9 ± 1.7	32.1 ± 0.6	64.6
	SelfReg	52.1 ± 0.2	98.0 ± 0.1	77.8 ± 0.9	85.6 ± 0.4	67.9 ± 0.7	47.0 ± 0.3	42.8 ± 0.0	67.3
	CausIRL$_C$	51.7 ± 0.1	97.9 ± 0.1	77.5 ± 0.6	85.8 ± 0.1	68.6 ± 0.3	47.3 ± 0.8	41.9 ± 0.1	67.3
	CausIRL$_M$	51.6 ± 0.1	97.9 ± 0.0	77.6 ± 0.4	84.0 ± 0.8	65.7 ± 0.6	46.3 ± 0.9	40.3 ± 0.2	66.2
Reported by authors	mDSDI	52.2 ± 0.2	98.0 ± 0.1	79.0 ± 0.3	86.2 ± 0.2	69.2 ± 0.4	48.1 ± 1.4	42.8 ± 0.1	67.9
	SWAD	-	-	79.1 ± 0.1	88.1 ± 0.1	70.6 ± 0.2	50.0 ± 0.3	46.5 ± 0.1	-
	T3A	-	-	80.0 ± 0.2	85.3 ± 0.6	68.3 ± 0.1	47.0 ± 0.6	-	-
	DRM (ours)	**71.2** ± 0.6	97.6 ± 0.1	77.9 ± 0.5	84.8 ± 0.5	65.7 ± 0.6	48.2 ± 0.2	41.0 ± 0.2	**69.5**

photos and cartoons, respectively. We test our approach on the diversity shift datasets. Following DomainBed, we present results on RotatedMNIST (Ghifary et al., 2015), VLCS (Fang et al., 2013), PACS (Li et al., 2017), Office-Home (Venkateswara et al., 2017), Terra Incognita (Beery et al., 2018) and DomainNet (Peng et al., 2019). In addition to the 14 methods we mentioned above, we also reports the results of other recent methods, including: Fish (Shi et al., 2021), Fishr (Rame et al., 2022), AND-mask (Parascandolo et al., 2020), SAND-mask (Shahtalebi et al., 2021), SelfReg (Kim et al., 2021), CausIRL (Chevalley et al., 2022), mDSDI (Bui et al., 2021), SWAD (Cha et al., 2021), and T3A (Iwasawa & Matsuo, 2021).

Results. We report the experimental results in Table 2. It shows that DRM does not hurt model performance for diversity shift. We observe that performance of ERM and other domain generalization algorithms are similar. The performance of our DRM approach is comparable with others on the diversity shift datasets, and improves by $\sim 2\%$ on average concerning the DomainBed benchmark.

5.3 Visual Explanation

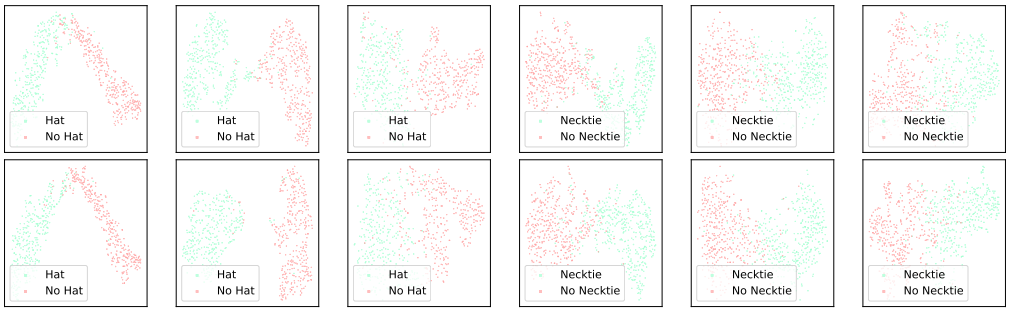
In this section, we choose the *CelebA-HB* dataset and the *CelebA-NS* dataset to visualize the results. For *CelebA-HB*, the indirect effect is the pathway between the spurious feature and label, which is "Wearing Hat" and "No Beard", respectively. For *CelebA-NS*, the indirect effect is the pathway between "Wearing Necktie" and "Smiling".

Analysis of indirect effect representation. Our DRM approach recovers the indirect effect in the first stage. Thus the quality of indirect effect representation Z is important. We use t-SNE (Van der Maaten & Hinton, 2008) to reduce the dimension of Z extracted in the first stage to 2 and show the result in Figure 4. We can observe that data points with the same spurious feature show a clustering effect, which means that Z is an appropriate representation of the spurious feature.

Attention map. In Figure 5, we present the attention maps of the last convolution layer for ERM (the second row) and DRM (the third row). The model trained by ERM focuses on the spurious feature "Wearing Hat" and "Wearing Necktie", while the model trained by DRM focuses on the stable feature "No Beard" and "Smiling".

6 Related Works

Domain generalization with causality. A large body of works has introduced tools from causality inference to the domain generalization problem. Causality has been shown to be robust across



(a) Test Env 0 (b) Test Env 1 (c) Test Env 2 (d) Test Env 0 (e) Test Env 1 (f) Test Env 2

Figure 4: Visualization of 2-D t-SNE result of Z. Different colors represent different spurious features, "Wearing Hat" vs. "No Hat", or "Wearing Neckline" vs. "No Neckline". (a)(b)(c) are for the *CelebA-HB* dataset and (d)(e)(f) are for the *CelebA-NS* dataset.

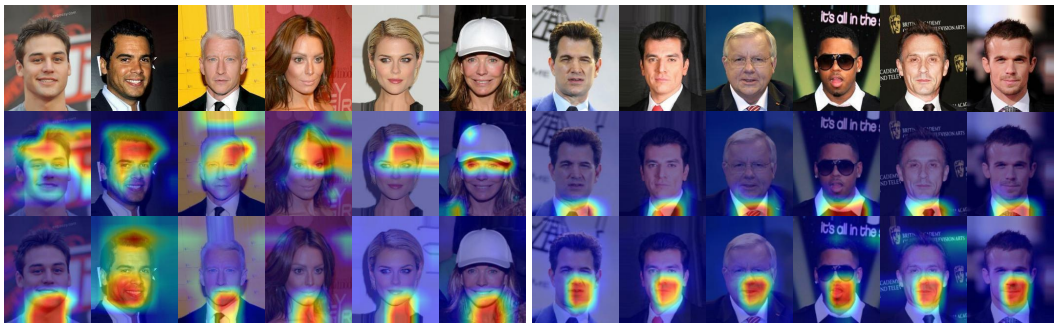


Figure 5: Attention map. The first row is the original images, the second row and the third row are attention maps of ERM and our method, respectively. **The left half** is on the *CelebA-HB* dataset, where the stable causality is "No Beard" and the spurious correlation is "Wearing Hat". **The right half** is on the *CelebA-NS* dataset, where the stable casualty is "Smiling" and the spurious correlation is "Wearing Necktie".

domains (Peters et al., 2016), and some works discussed which causal factors can be extracted (Schölkopf et al., 2012; 2021) and the connection between causality and generalization (Christiansen et al., 2020). Pearl (2001) gave a definition of natural direct effects. Although the concept has been introduced in early works (Zhang & Bareinboim, 2018; Qi et al., 2020; Heskes et al., 2020), no work analyzed domain generalization using this framework.

Matching based methods. Matching is a common approach that aims to eliminate selection bias in causal inference by matching comparable instances (Rosenbaum & Rubin, 1983). Mahajan et al. (2021) proposed an unsupervised matching algorithm and Wang et al. (2022) introduced the propensity score matching method to balance the mini-batch. Our method also uses a mini-batch balancing approach in the second stage. However, we propose a new definition of balanced distribution, which makes the marginal distribution of Y unchanged. Moreover, we extract the indirect-effect representation in the first stage by learning a domain discriminator, which is different from above approaches and helps our approach to achieve better performance.

7 Conclusion

In this paper, we introduce the concept of direct and indirect effects from causal inference to the domain generalization problems. We prove that a model learns direct effects is optimal for correlation shift. We propose a domain generalization method to extract the indirect-effect representation and remove the indirect effects during training. Experimental results show that our approach achieves the state-of-the-art performance.

ETHICS STATEMENT

We trained our model on a publicly available face dataset CelebA (Liu et al., 2015) to evaluate the domain generalization performance of our method. The dataset contains many attributes of faces and the model may reflect the biases carried in the dataset. Our method aims to improve the generalization ability of machine learning models. Models with good generalization performance can operate robustly in many senarios in reality and benefit the society.

REPRODUCIBILITY

Our architectures and hyperparameters follow DomainBed benchmark and more implementation details are given in Appendix C. All datasets in this paper are public and available. We plan to release our code as open source.

REFERENCES

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020a.

Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2020b.

Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450, 2021.

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *European Conference on Computer Vision*, pp. 472–489. Springer, 2018.

Alexis Bellot and Mihaela van der Schaar. Accounting for unobserved confounding in domain generalization. *arXiv preprint arXiv:2007.10653*, 2020.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34:21189–21201, 2021.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Mathieu Chevalley, Charlotte Bunne, Andreas Krause, and Stefan Bauer. Invariant causal mechanisms through distribution matching. *arXiv preprint arXiv:2206.11646*, 2022.

Rune Christiansen, Niklas Pfister, Martin Emil Jakobsen, Nicola Gnecco, and Jonas Peters. A causal framework for distribution generalization, 2020. URL `https://arxiv.org/abs/2006.07433`.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1657–1664, 2013.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. cvpr. 2016. *arXiv preprint arXiv:1512.03385*, 2016.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789, 2020.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pp. 124–140. Springer, 2020.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pp. 4069–4077. PMLR, 2021.

Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9619–9628, 2021.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018b.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018c.

Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in Neural Information Processing Systems*, 33:3118–3129, 2020.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018d.

Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pp. 3915–3924. PMLR, 2019.

Chang Liu, Xinwei Sun, Jindong Wang, Haoyue Tang, Tao Li, Tao Qin, Wei Chen, and Tie-Yan Liu. Learning causal semantic representation for out-of-distribution prediction. *Advances in Neural Information Processing Systems*, 34:6155–6170, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8046–8056, 2022.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pp. 7313–7324. PMLR, 2021.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5715–5725, 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2020.

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *arXiv preprint arXiv:1910.11645*, 2(7):8, 2019.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

Judea Pearl. Direct and indirect effects. *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 373, 2001.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning*, pp. 7728–7738. PMLR, 2020.

Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10860–10869, 2020.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020.

Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 18347–18377. PMLR, 2022.

Alexander Robey, George J Pappas, and Hamed Hassani. Model-based domain generalization. *Advances in Neural Information Processing Systems*, 34:20210–20229, 2021.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2020.

Jongbin Ryu, Gitaek Kwon, Ming-Hsuan Yang, and Jongwoo Lim. Generalized convolutional forest networks for domain generalization and visual recognition. In *International conference on learning representations*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Soroosh Shahtalebi, Jean-Christophe Gagnon-Audet, Touraj Laleh, Mojtaba Faramarzi, Kartik Ahuja, and Irina Rish. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization, 2021. URL `https://arxiv.org/abs/2106.02266`.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vladimir Vapnik. Statistical learning theory wiley. *New York*, 1(624):2, 1998.

Ramakrishna Vedantam, David Lopez-Paz, and David J Schwab. An empirical investigation of domain generalization with empirical risk minimizers. *Advances in Neural Information Processing Systems*, 34:28131–28143, 2021.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

Xinyi Wang, Michael Saxon, Jiachen Li, Hongyang Zhang, Kun Zhang, and William Yang Wang. Causal balancing for domain generalization. *arXiv preprint arXiv:2206.05263*, 2022.

Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3622–3626. IEEE, 2020.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6502–6509, 2020.

Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

Fu-En Yang, Yuan-Chia Cheng, Zu-Yun Shiau, and Yu-Chiang Frank Wang. Adversarial teacher-student representation learning for domain generalization. *Advances in Neural Information Processing Systems*, 34:19448–19460, 2021.

Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:23519–23531, 2021.

Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7947–7958, 2022.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2100–2110, 2019.

Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pp. 12356–12367. PMLR, 2021a.

Guojun Zhang, Han Zhao, Yaoliang Yu, and Pascal Poupart. Quantifying and improving transferability in domain generalization. *Advances in Neural Information Processing Systems*, 34: 10957–10970, 2021b.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 8:9, 2020.

Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33: 16096–16107, 2020.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13025–13032, 2020.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A OTHER RELATED WORKS

Learning invariant features. To enable classifiers to generalize across domains, a very intuitive idea is to force the model to learn a representation with cross-domain invariance (Motiian et al., 2017; Wald et al., 2021), which is usually implemented by adding a regularization term to the loss (Li et al., 2020; Zhao et al., 2020; Robey et al., 2021). Representative methods include using maximum mean discrepancy as a divergence measure (Muandet et al., 2013), seeking a data representation such that the predictors using the representation are invariant (IRM) (Arjovsky et al., 2019), encouraging the training risks in different domains to be similar (Krueger et al., 2021), or using a correlation matrix to construct a new loss function (Lv et al., 2022). Some works studied the conditions under which invariance can guarantee domain generalization from a theoretical point of view (Ye et al., 2021), while other works quantified transferability of feature embeddings learned by domain generalization models (Zhang et al., 2021b). Ahuja et al. (2020b) analyzed different finite sample and asymptotic behavior of ERM and IRM, and Ahuja et al. (2020a) expanded IRM from a game theory perspective. However, existing works have demonstrated that it is difficult to achieve good generalization performance by relying only on the constraint of cross-domain invariance (Mahajan et al., 2021; Ahuja et al., 2021), and some other works claimed that many of these approaches still fail to capture the invariance (Kamath et al., 2021; Rosenfeld et al., 2020). Empirical work has also questioned the effectiveness of these methods (Gulrajani & Lopez-Paz, 2020). On average, ERM outperforms the other methods (Vedantam et al., 2021).

Data augmentation. Data augmentation is a kind of valid methods to impove the generalization ability of models. Domain Randomization bridge the simulated environment and the real world by generating rich enough data, which can benefit the o.o.d. generalization (Tobin et al., 2017). Some works performed domain adversarial training to generate data across environments (Shankar et al., 2018; Volpi et al., 2018). Generative models and transformation models such as CycleGAN (Zhu et al., 2017) are also used to perform data augmentation (Zhou et al., 2020; Qiao et al., 2020; Yue et al., 2019).

Other approaches. In addition to the above approaches, many works have been done to enhance the performance of o.o.d. generalization in a variety of different ways. A great deal of work has been done to improve generalization performance by analyzing and designing new neural network structures (Li et al., 2017; Zhang et al., 2021a; Ryu et al., 2019). Meta-learning is another helpful direction, and many approaches based on it have emerged (Li et al., 2018a; Balaji et al., 2018; Dou et al., 2019; Li et al., 2019). In addition, Yang et al. (2021) proposed an approach of Adversarial Teacher-Student Representation Learning to derive generalizable representations; Zhou et al. (2021) proposed an approach based on feature statistics mixing across source domains; Piratla et al. (2020) joint learned common components and domain-specific components by modifying the last classification layer; Carlucci et al. (2019) combined supervised with self-supervised learning to improve the generalization performance of the model by solving the Jigsaw puzzles.

B PROOFS

In this section, we give full proofs of the main theorems in the paper.

B.1 PROOF OF THEOREM 1

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

$$\begin{aligned} R^e(h) =& \mathbb{E}^e\left[\left(Y^e - h\left(X^e\right)\right)^2\right] \\ =& \mathbb{E}^e\left[\left(Y^e - f_1\left(X^e\right) + f_1\left(X^e\right) - h\left(X^e\right)\right)^2\right] \\ =& \mathbb{E}^e\left[\left(f_2\left(Z^e\right)+\varepsilon_1\right)^2\right]+\mathbb{E}^e\left[\left(f_1\left(X^e\right)-h\left(X^e\right)\right)^2\right]+2\mathbb{E}^e\left[\left(Y^e-f_1\left(X^e\right)\right)\left(f_1\left(X^e\right)-h\left(X^e\right)\right)\right]. \end{aligned}$$

Let $Var\left[\varepsilon_1\right] = \sigma_1^2$, $Var_X\left[f_2(Z^e)\right] == \sigma_2^2$, then we have

$$R^e(h) = \sigma_1^2 + \sigma_2^2 + E^e\left[\left(f_1\left(X^e\right) - h\left(X^e\right)\right)^2\right] + 2E^e\left[f_2\left(f_3\left(X^e, e\right)\right)\left(f_1\left(X^e\right) - h\left(X^e\right)\right)\right].$$

According to Definition 3, there must exist an environment e^* that satisfies

$$f_2\left(f_3(x,e)\right) = f_1\left(x\right) - h\left(x\right),$$

which means that

$$E^{e^*}\left[f_2\left(f_3\left(X^{e^*},e^*\right)\right)\left(f_1\left(X^{e^*}\right) - h\left(X^{e^*}\right)\right)\right] \geq 0.$$

Hence,

$$\max_{e\in\mathcal{E}_{\text{all}}} R^e(h) \geq \sigma_1^2 + \sigma_2^2.$$

The equivalence holds if and only if

$$h^* = f_1.$$

□

B.2 PROOF OF LEMMA 1

Lemma 1. *If Assumption 1 holds between e_T and e_S, then we have $d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}_B^{e_S}] \leq d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}^{e_S}]$, where $d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}_B^{e}] = 2\sup_{\eta\in\mathcal{H}}|\Pr_{x\sim\mathbb{P}^{e_T}}[\eta(x)=1] - \Pr_{x\sim\mathbb{P}_B^{e}}[\eta(x)=1]|$.*

Proof. Suppose that $\Pr(E=e_T) = \Pr(E=e_S) = \frac{1}{2}$. Then we have

$$\begin{aligned}
d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}^{e_S}] &= 2\sup_{\eta\in\mathcal{H}}|\Pr_{x\sim\mathbb{P}^{e_T}}[\eta(x)=1] - \Pr_{x\sim\mathbb{P}^{e_S}}[\eta(x)=1]| \\
&= 4\sup_{\eta\in\mathcal{H}}|\frac{1}{2}\Pr_{x\sim\mathbb{P}^{e_T}}[\eta(x)=1] - \frac{1}{2}\Pr_{x\sim\mathbb{P}^{e_S}}[\eta(x)=1]| \\
&= 4\sup_{\eta\in\mathcal{H}}|\frac{1}{2}\Pr_{x\sim\mathbb{P}^{e_T}}[\eta(x)=1] + \frac{1}{2}\Pr_{x\sim\mathbb{P}^{e_S}}[\eta(x)=0] - \frac{1}{2}|
\end{aligned}$$

Consider a binary classification problem using $\eta : \mathcal{X} \to \{0,1\}$ to classify the domains e_S and e_T. We assume that $f(x) = 1$ if x is from domain e_T and $f(x) = 0$ if x is from domain e_S, where $f : \mathcal{X} \to \{0,1\}$ is the labeling function. Then we have

$$\begin{aligned}
d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}^{e_S}] &= 4\sup_{\eta\in\mathcal{H}}|\int_{\mathcal{X}} \Pr[\eta(x) = f(x) \mid X=x]\Pr(X=x) - \frac{1}{2}| \\
&= 4\sup_{\eta\in\mathcal{H}}|\int_{\mathcal{X}} ((1-\eta(x))\Pr(E=e_S \mid X=x) \\
&\quad + \eta(x)\Pr(E=e_T \mid X=x))\Pr(X=x) - \frac{1}{2}|.
\end{aligned}$$

Since

$$\begin{aligned}
\Pr(X=x) &= \frac{1}{2}(\mathbb{P}^{e_S}(x) + \mathbb{P}^{e_T}(x)), \\
\Pr(E=e_S \mid X=x) &= \frac{\Pr(X=x \mid E=e_S)\Pr(E=e_S)}{\Pr(X=x)} = \frac{\mathbb{P}^{e_S}(x)}{\mathbb{P}^{e_S}(x) + \mathbb{P}^{e_T}(x)}, \\
\Pr(E=e_T \mid X=x) &= \frac{\Pr(X=x \mid E=e_T)\Pr(E=e_T)}{\Pr(X=x)} = \frac{\mathbb{P}^{e_T}(x)}{\mathbb{P}^{e_S}(x) + \mathbb{P}^{e_T}(x)},
\end{aligned}$$

then

$$\begin{aligned}
d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}^{e_S}] &= 4\sup_{\eta\in\mathcal{H}}|\frac{1}{2}\int_{\mathcal{X}} ((1-\eta(x))\mathbb{P}^{e_S}(x) + \eta(x)\mathbb{P}^{e_T}(x)) - \frac{1}{2}| \\
&= 4\sup_{\eta\in\mathcal{H}}(\frac{1}{2}\int_{\mathcal{X}} ((1-\eta(x))\mathbb{P}^{e_S}(x) + \eta(x)\mathbb{P}^{e_T}(x)) - \frac{1}{2}) \\
&= 2(\int_{\mathcal{X}} \max\left\{\mathbb{P}^{e_S}(x), \mathbb{P}^{e_T}(x)\right\} - 1).
\end{aligned}$$

Similarly, we have

$$d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}_B^{e_S}] = 2(\int_{\mathcal{X}} \max\left\{\mathbb{P}_B^{e_S}(x), \mathbb{P}^{e_T}(x)\right\} - 1).$$

Since Assumption 1 holds,

$$\forall x \in \mathcal{X}, \max\{\mathbb{P}_B^{e_S}(x), \mathbb{P}^{e_T}(x)\} \leq \max\{\mathbb{P}^{e_S}(x), \mathbb{P}^{e_T}(x)\}$$

Hence we have

$$d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}_B^{e_S}] \leq d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}^{e_S}].$$

□

B.3 PROOF OF THEOREM 2

Proof. For a case with single training environment e_S and a single test environment e_T, it has been shown by Ben-David et al. (2006) that with probability at least $1-\delta$, for every $h \in \mathcal{H}$, we have

$$R^{e_T}(h) \leq \hat{R}^{e_S}(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + d_{\mathcal{H}}(\mathbb{P}^{e_S}, \mathbb{P}^{e_T}) + \lambda$$

We consider an unseen environment $e_T \in \mathcal{E}_{test}$ and all training environments $\mathcal{E}_{train} = \left\{e_S^i \mid i = 1, 2, \ldots, N_S\right\}$. We can define an environment $\bar{e}_S$ whose distribution is

$$\bar{\mathbb{P}}_B^{e_S} = \frac{1}{N_S}\sum_{i=1}^{N_S} \mathbb{P}_B^{e_S^i}.$$

Thus $R^{e_T}(h)$ can be bounded as

$$\begin{aligned} R^{e_T}(h) &\leq \frac{1}{N_S}\sum_{i=1}^{N_S} \hat{R}_B^{e_S^i}(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + d_{\mathcal{H}}(\bar{\mathbb{P}}_B^{e_S}, \mathbb{P}^{e_T}) + \lambda \\ &\leq \frac{1}{N_S}\sum_{i=1}^{N_S} \hat{R}_B^{e_S^i}(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + \frac{1}{N_S}\sum_{i=1}^{N_S} d_{\mathcal{H}}(\mathbb{P}_B^{e_S^i}, \mathbb{P}^{e_T}) + \lambda. \end{aligned}$$

Since Assumption 1 holds between all e_S^i and e_T, according to Lemma 1, $\forall e_S^i \in \mathcal{E}_{train}$, $d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}_B^{e_S^i}] \leq d_{\mathcal{H}}[\mathbb{P}^{e_T}, \mathbb{P}^{e_S^i}]$. Then we have

$$\begin{aligned} R^{e_T}(h) &\leq \frac{1}{N_S}\sum_{i=1}^{N_S} \hat{R}_B^{e_S^i}(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + \frac{1}{N_S}\sum_{i=1}^{N_S} d_{\mathcal{H}}(\mathbb{P}^{e_S^i}, \mathbb{P}^{e_T}) + \lambda \\ &\leq \frac{1}{N_S}\sum_{i=1}^{N_S} \hat{R}_B^{e_S^i}(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + \epsilon + \lambda. \end{aligned}$$

□

B.4 PROOF FOR THEOREM 3

Proof. We assume that for any sample in a batch, if it belongs to class k, we include its matched sample into the batch with the probability $\mathbb{P}_k$.

Let Bs be the size of batch before balancing. Without loss of generality, we assume that we will match n samples with probability $\mathbb{P}_k$ for each sample in the original batch. Since original batch are sampled i.i.d. from the training set distribution, in the original samples, we have

$$\mathbb{E}(\mathrm{N}_1(Y=j)) = \omega_j \times Bs,$$

where $\mathrm{N}_1(Y=j)$ is the number of samples who belong to class j in the original batch.

In the balancing process, each sample searches for a matching sample who has a different label from the training set. Assume that $i \neq j$, the probability of $Y=j$ among the matched samples of the sample with label $Y=i$ is $\frac{\omega_j\omega_i}{1-\omega_i}\mathbb{P}_i$. Then the expected number of samples whose $Y=j$ in the set of matched samples is

$$\mathbb{E}(\mathrm{N}_2(Y=j)) = \sum_{i\in\{1,2,\ldots,j-1,j+1,\ldots,m\}} \frac{\omega_j\omega_i}{1-\omega_i}\mathbb{P}_i \times n \times Bs.$$

Then we have

$$\mathbb{E}(\mathrm{N}(Y=j)) = \omega_j \times Bs \times (1 + \sum_{i \in \{1,2,\ldots,j-1,j+1,\ldots,m\}} \frac{\omega_i}{1-\omega_i} \mathbb{P}_i \times n),$$

where $\mathrm{N}(Y=j) = \mathrm{N}_1(Y=j) + \mathrm{N}_2(Y=j)$.

If $\mathbb{P}_i = \frac{(1-\omega_i)/\omega_i}{\max_k (1-\omega_k)/\omega_k}$, then

$$\mathbb{E}(\mathrm{N}(Y=j)) = \omega_j \times Bs \times (1 + \frac{n(m-1)}{\max_k (1-\omega_k)/\omega_k}) = c \times \omega_j,$$

where $c = (1 + \frac{n(m-1)}{\max_k (1-\omega_k)/\omega_k}) \times Bs$ is a constant. □

C Extra Experimental Results

C.1 Hyperparameter search

We search hyperparameters with the same distribution as DomainBed. On DomainBed benchmark, some distribution of hyperparameters are related to image size. To avoid human intervention, we resize images of *CelebA-HB* and *CelebA-NS* to 224×224, which is the standard size on DomainBed benchmark. To save computing resource and time, we reduce the number of search to eight for all diversity shift datasets and small-image (smaller than 224×224) correlation shift datasets. DRM will augment the batch, so we reduce the batchsize for big-image datasets to avoid GPU memory overflow. The two stages of DRM use the same model: Resnet-50 (He et al., 2016) for big-image datasets and MNIST-CNN for small-image datasets, which is consistent with DomainBed. And hyperparameters distribution of two stages are the same, which is also consistent with DomainBed. In stage 1, we divide the data from train environments into the training set and the validation set in a ratio of 8:2. We choose the model which perform best on the validation set.

C.2 Full results

Table 3: The result for CMNIST

Algorithm	**+90%**	**+80%**	**-90%**	**Avg**
ERM	71.7 ± 0.1	72.9 ± 0.2	10.0 ± 0.1	51.5
IRM	72.5 ± 0.1	73.3 ± 0.5	10.2 ± 0.3	52.0
GroupDRO	73.1 ± 0.3	73.2 ± 0.2	10.0 ± 0.2	52.1
Mixup	72.7 ± 0.4	73.4 ± 0.1	10.1 ± 0.1	52.1
MLDG	71.5 ± 0.2	73.1 ± 0.2	9.8 ± 0.1	51.5
CORAL	71.6 ± 0.3	73.1 ± 0.1	9.9 ± 0.1	51.5
MMD	71.4 ± 0.2	73.1 ± 0.2	9.9 ± 0.3	51.5
DANN	71.4 ± 0.9	73.1 ± 0.1	10.0 ± 0.0	51.5
CDANN	72.0 ± 0.2	73.0 ± 0.2	10.2 ± 0.1	51.7
MTL	70.9 ± 0.2	72.8 ± 0.3	10.5 ± 0.1	51.4
SagNet	71.8 ± 0.2	73.0 ± 0.2	10.3 ± 0.1	51.7
ARM	82.0 ± 0.5	76.5 ± 0.3	10.2 ± 0.0	56.2
VREx	72.4 ± 0.3	72.9 ± 0.4	10.2 ± 0.0	51.8
RSC	71.9 ± 0.3	73.1 ± 0.2	10.0 ± 0.2	51.7
DRM (ours)	71.4 ± 0.3	72.4 ± 0.4	$\mathbf{69.7 \pm 1.5}$	**71.2**

Table 4: The results for 3DShapes

Algorithm	+90%	+80%	-90%	Avg
ERM	74.3 ± 0.4	75.5 ± 0.2	10.1 ± 0.1	53.3
IRM	74.2 ± 0.2	75.4 ± 0.1	10.0 ± 0.0	53.2
GroupDRO	74.6 ± 0.1	75.1 ± 0.1	10.5 ± 0.4	53.4
Mixup	74.6 ± 0.4	75.4 ± 0.2	10.2 ± 0.1	53.4
MLDG	75.0 ± 0.2	75.4 ± 0.1	10.1 ± 0.1	53.5
CORAL	74.6 ± 0.2	75.2 ± 0.2	10.0 ± 0.0	53.3
MMD	74.6 ± 0.1	75.2 ± 0.1	10.0 ± 0.1	53.2
DANN	74.6 ± 0.1	75.2 ± 0.1	10.0 ± 0.0	53.3
CDANN	74.4 ± 0.4	75.3 ± 0.0	10.0 ± 0.0	53.3
MTL	74.7 ± 0.2	75.4 ± 0.2	10.1 ± 0.0	53.4
SagNet	74.9 ± 0.1	75.1 ± 0.1	10.1 ± 0.1	53.4
ARM	81.8 ± 0.3	73.9 ± 0.8	10.0 ± 0.0	55.2
VREx	74.6 ± 0.4	75.2 ± 0.0	10.8 ± 0.3	53.5
RSC	74.4 ± 0.2	75.1 ± 0.1	10.1 ± 0.1	53.2
DRM (ours)	74.5 ± 0.2	75.1 ± 0.1	**74.8 ± 0.1**	**74.8**

Table 5: The results for DSprites

Algorithm	+90%	+80%	-90%	Avg
ERM	73.5 ± 0.2	74.8 ± 0.0	13.8 ± 0.5	54.0
IRM	73.5 ± 0.2	74.2 ± 0.1	14.5 ± 0.3	54.0
GroupDRO	73.8 ± 0.2	74.4 ± 0.1	15.0 ± 0.4	54.4
Mixup	73.4 ± 0.1	74.3 ± 0.2	14.0 ± 0.3	53.9
MLDG	73.9 ± 0.3	74.4 ± 0.3	14.3 ± 0.3	54.2
CORAL	73.4 ± 0.3	74.5 ± 0.1	13.8 ± 0.2	53.9
MMD	65.8 ± 6.4	74.2 ± 0.1	14.4 ± 0.0	51.4
DANN	73.7 ± 0.5	73.9 ± 0.2	14.7 ± 0.3	54.1
CDANN	73.7 ± 0.3	74.0 ± 0.1	14.4 ± 0.2	54.0
MTL	73.3 ± 0.0	74.8 ± 0.2	14.8 ± 0.5	54.3
SagNet	73.5 ± 0.1	74.8 ± 0.1	13.6 ± 0.1	54.0
ARM	86.9 ± 0.4	77.6 ± 0.4	14.5 ± 0.6	59.7
VREx	73.4 ± 0.2	74.6 ± 0.1	13.8 ± 0.3	53.9
RSC	73.7 ± 0.3	74.5 ± 0.2	13.3 ± 0.2	53.8
DRM (ours)	73.7 ± 0.2	74.4 ± 0.1	**73.3** ± 0.5	**73.8**

Table 6: The results for CelebA-HB

Algorithm	+90%	+80%	-90%	Avg
ERM	67.3 ± 0.2	71.8 ± 0.3	16.8 ± 1.2	52.0
IRM	65.9 ± 0.7	70.0 ± 0.7	20.4 ± 2.1	52.1
GroupDRO	68.3 ± 1.2	71.9 ± 0.2	18.3 ± 1.5	52.8
Mixup	68.4 ± 0.6	70.8 ± 0.6	17.9 ± 3.4	52.4
MLDG	68.3 ± 0.2	70.6 ± 0.1	20.0 ± 2.1	53.0
CORAL	68.3 ± 0.5	71.2 ± 0.2	17.7 ± 1.6	52.4
MMD	64.5 ± 0.6	70.1 ± 0.6	17.4 ± 1.8	50.7
DANN	67.0 ± 1.0	71.2 ± 1.5	16.9 ± 1.7	51.7
CDANN	66.8 ± 0.8	72.1 ± 0.4	18.6 ± 2.5	52.5
MTL	65.9 ± 0.9	71.6 ± 0.1	23.5 ± 1.4	53.7
SagNet	64.6 ± 1.0	71.6 ± 0.4	14.9 ± 0.6	50.4
ARM	66.7 ± 1.1	72.8 ± 0.1	22.8 ± 2.2	54.1
VREx	66.9 ± 0.3	71.4 ± 0.2	19.2 ± 1.8	52.5
RSC	67.4 ± 0.4	71.1 ± 0.9	18.9 ± 1.1	52.5
DRM (ours)	68.1 ± 1.0	69.3 ± 1.4	**61.0 ± 4.9**	**66.1**

Table 7: The results for CelebA-NS

Algorithm	+90%	+80%	-90%	Avg
ERM	67.8 ± 1.0	69.1 ± 0.6	21.1 ± 0.4	52.7
IRM	68.2 ± 0.3	69.8 ± 0.1	21.5 ± 0.9	53.2
GroupDRO	68.0 ± 0.4	70.6 ± 0.3	21.2 ± 0.2	53.3
Mixup	68.7 ± 0.6	70.2 ± 0.7	22.2 ± 1.5	53.7
MLDG	67.7 ± 0.3	70.8 ± 0.1	22.7 ± 1.7	53.7
CORAL	68.2 ± 0.5	69.8 ± 0.1	22.1 ± 1.1	53.4
MMD	68.1 ± 0.3	69.4 ± 0.4	22.5 ± 0.6	53.3
DANN	69.3 ± 0.6	69.9 ± 0.6	21.8 ± 1.5	53.7
CDANN	68.0 ± 0.5	71.1 ± 0.4	22.5 ± 1.2	53.9
MTL	67.7 ± 0.4	69.5 ± 0.3	27.6 ± 1.2	54.9
SagNet	67.8 ± 0.3	69.5 ± 0.2	22.0 ± 0.6	53.1
ARM	67.7 ± 0.2	70.3 ± 0.3	21.1 ± 1.4	53.0
VREx	69.7 ± 0.4	69.7 ± 0.3	20.3 ± 0.4	53.2
RSC	68.9 ± 0.8	70.3 ± 0.2	23.7 ± 0.8	54.3
DRM (ours)	67.7 ± 1.1	68.6 ± 0.3	**59.9 ± 2.6**	**65.4**

Table 8: The result for RMNIST

Algorithm	0	15	30	45	60	75	Avg
ERM	95.9 ± 0.1	98.9 ± 0.0	98.8 ± 0.0	98.9 ± 0.0	98.9 ± 0.0	96.4 ± 0.0	98.0
IRM	95.5 ± 0.1	98.8 ± 0.2	98.7 ± 0.1	98.6 ± 0.1	98.7 ± 0.0	95.9 ± 0.2	97.7
GroupDRO	95.6 ± 0.1	98.9 ± 0.1	98.9 ± 0.1	99.0 ± 0.0	98.9 ± 0.0	96.5 ± 0.2	98.0
Mixup	95.8 ± 0.3	98.9 ± 0.0	98.9 ± 0.0	98.9 ± 0.0	98.8 ± 0.1	96.5 ± 0.3	98.0
MLDG	95.8 ± 0.1	98.9 ± 0.1	99.0 ± 0.0	98.9 ± 0.1	99.0 ± 0.0	95.8 ± 0.3	97.9
CORAL	95.8 ± 0.3	98.8 ± 0.0	98.9 ± 0.0	99.0 ± 0.0	98.9 ± 0.1	96.4 ± 0.2	98.0
MMD	95.6 ± 0.1	98.9 ± 0.1	99.0 ± 0.0	99.0 ± 0.0	98.9 ± 0.0	96.0 ± 0.2	97.9
DANN	95.0 ± 0.5	98.9 ± 0.1	99.0 ± 0.0	90.0 ± 0.1	98.9 ± 0.0	96.3 ± 0.2	97.8
CDANN	95.7 ± 0.2	98.8 ± 0.0	98.9 ± 0.1	98.9 ± 0.1	98.9 ± 0.1	96.1 ± 0.3	97.9
MTL	95.6 ± 0.1	99.0 ± 0.1	99.0 ± 0.0	98.9 ± 0.1	99.0 ± 0.1	95.8 ± 0.2	97.9
SagNet	95.9 ± 0.3	98.9 ± 0.1	99.0 ± 0.1	99.1 ± 0.0	99.0 ± 0.1	96.3 ± 0.1	98.0
ARM	96.7 ± 0.2	99.1 ± 0.0	99.0 ± 0.0	99.0 ± 0.1	99.1 ± 0.1	96.5 ± 0.4	98.2
VREx	95.9 ± 0.2	99.0 ± 0.1	98.9 ± 0.1	98.9 ± 0.1	98.7 ± 0.1	96.2 ± 0.2	97.9
RSC	94.8 ± 0.5	98.7 ± 0.1	98.8 ± 0.1	98.8 ± 0.0	98.9 ± 0.1	95.9 ± 0.2	97.6
DRM(ours)	94.5 ± 0.6	98.6 ± 0.1	98.8 ± 0.1	99.1 ± 0.0	98.9 ± 0.0	96.0 ± 0.3	97.6

Table 9: The result for VLCS

Algorithm	C	L	S	V	Avg
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4
MLDG	97.4 ± 0.2	65.2 ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2
CORAL	98.3 ± 0.1	66.1 ± 1.2	73.4 ± 0.3	77.5 ± 1.2	78.8
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.5
DANN	99.0 ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6
CDANN	97.1 ± 0.3	65.1 ± 1.2	70.7 ± 0.8	77.1 ± 1.5	77.5
MTL	97.8 ± 0.4	64.3 ± 0.3	71.5 ± 0.7	75.3 ± 1.7	77.2
SagNet	97.9 ± 0.4	64.5 ± 0.5	71.4 ± 1.3	77.5 ± 0.5	77.8
ARM	98.7 ± 0.2	63.6 ± 0.7	71.3 ± 1.2	76.7 ± 0.6	77.6
VREx	98.4 ± 0.3	64.4 ± 1.4	74.1 ± 0.4	76.2 ± 1.3	78.3
RSC	97.9 ± 0.2	64.4 ± 1.4	74.1 ± 0.4	76.2 ± 1.3	77.1
DRM(ours)	97.9 ± 0.2	65.1 ± 0.7	71.5 ± 0.9	77.1 ± 1.7	77.9

Table 10: The result for PACS

Algorithm	A	C	P	S	Avg
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
CORAL	88.3 ± 0.2	80 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80 ± 0.4	86.3
ARM	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1
VREx	86 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
RSC	85.4 ± 0.8	79.7 ± 1.8	97.6 ± 0.3	78.2 ± 1.2	85.2
DRM(ours)	85.0 ± 0.9	80.0 ± 0.5	96.7 ± 0.6	77.5 ± 1.2	84.8

Table 11: The result for OFFICEHOME

Algorithm	A	C	P	R	Avg
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
Mixup	62.4 ± 0.8	54.8 ± 0.6	76.9 ± 0.3	78.3 ± 0.2	68.1
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
CORAL	65.3 ± 0.4	54.4 ± 0.5	76.5 ± 0.1	78.4 ± 0.5	68.7
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4
SagNet	63.4 ± 0.2	54.8 ± 0.4	75.8 ± 0.4	78.3 ± 0.3	68.1
ARM	58.9 ± 0.8	51.0 ± 0.5	74.1 ± 0.1	75.2 ± 0.3	64.8
VREx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
RSC	60.7 ± 1.4	51.4 ± 0.3	74.8 ± 1.1	75.1 ± 1.3	65.5
DRM(ours)	60.4 ± 0.6	52.5 ± 0.5	74.2 ± 0.6	75.5 ± 0.9	65.7

Table 12: The result for TERRAINCOGNITA

Algorithm	L100	L38	L43	L46	Avg
ERM	49.8 ± 4.4	42.1 ± 1.4	56.9 ± 1.8	35.7 ± 3.9	46.1
IRM	54.6 ± 1.3	39.8 ± 1.9	56.2 ± 1.8	39.6 ± 0.8	47.6
GroupDRO	41.2 ± 0.7	38.6 ± 2.1	56.7 ± 0.9	36.4 ± 2.1	43.2
Mixup	59.6 ± 2.0	42.2 ± 1.4	55.9 ± 0.8	33.9 ± 1.4	47.9
MLDG	54.2 ± 3.0	44.3 ± 1.1	55.6 ± 0.3	36.9 ± 2.2	47.7
CORAL	51.6 ± 2.4	42.2 ± 1.0	57.0 ± 1.0	39.8 ± 2.9	47.6
MMD	41.9 ± 3.0	34.8 ± 1.0	57.0 ± 1.9	35.2 ± 1.8	42.2
DANN	51.1 ± 3.5	40.6 ± 0.6	57.4 ± 0.5	37.7 ± 1.8	46.7
CDANN	47.0 ± 1.9	41.3 ± 4.8	54.9 ± 1.7	39.8 ± 2.3	45.8
MTL	49.3 ± 1.2	39.6 ± 6.3	55.6 ± 1.1	37.8 ± 0.8	45.6
SagNet	53.0 ± 2.9	43.0 ± 2.5	57.9 ± 0.6	40.4 ± 1.3	48.6
ARM	49.3 ± 0.7	38.3 ± 2.4	55.8 ± 0.8	38.7 ± 1.3	45.5
VREx	48.2 ± 4.3	41.7 ± 1.3	56.8 ± 0.8	38.7 ± 3.1	46.4
RSC	50.2 ± 2.2	39.2 ± 1.4	56.3 ± 1.4	40.8 ± 0.6	46.6
DRM(ours)	52.8 ± 3.6	42.7 ± 1.3	56.3 ± 1.2	41.1 ± 2.0	48.2

Table 13: The result for DOMAINNET

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1 ± 0.3	18.8 ± 0.3	16.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9
IRM	48.5 ± 2.8	15.0 ± 1.5	38.3 ± 4.3	10.9 ± 0.5	48.2 ± 5.2	42.3 ± 3.1	33.9
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.3
Mixup	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.1	48.2 ± 0.5	39.2
MLDG	59.1 ± 0.2	19.1 ± 0.3	45.8 ± 0.7	13.4 ± 0.3	59.6 ± 0.2	50.2 ± 0.4	41.2
CORAL	59.2 ± 0.1	19.7 ± 0.2	46.6 ± 0.3	13.4 ± 0.3	59.8 ± 0.2	50.1 ± 0.6	41.5
MMD	32.1 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.8 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3
CDANN	54.6 ± 0.4	17.3 ± 0.1	43.7 ± 0.9	12.1 ± 0.7	56.2 ± 0.4	45.9 ± 0.5	38.3
MTL	57.9 ± 0.5	18.5 ± 0.4	46.0 ± 0.1	12.5 ± 0.1	59.5 ± 0.3	49.2 ± 0.1	40.6
SagNet	57.7 ± 0.3	19.0 ± 0.2	45.3 ± 0.3	12.7 ± 0.5	58.1 ± 0.5	48.8 ± 0.2	40.3
ARM	49.7 ± 0.3	16.3 ± 0.5	40.9 ± 1.1	9.4 ± 0.1	53.4 ± 0.4	43.5 ± 0.4	35.5
VREx	47.3 ± 3.5	16.0 ± 1.5	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.6
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.2 ± 0.2	55.7 ± 0.7	47.8 ± 0.9	38.9
DRM(ours)	58.5 ± 0.5	19.5 ± 0.4	45.4 ± 0.1	13.8 ± 0.6	59.0 ± 1.0	49.9 ± 0.7	41.0