Supplementary for DgCD: Domain-guided Channel Dropout for Domain Generalization

Seonggyeom Kim*
Byeongtae Park*
Hanyang University
Seoul, South Korea
{canino,byeongtae}@hanyang.ac.kr

Harim Lee
Hanyang University
Seoul, South Korea
hrimlee@hanyang.ac.kr

Dong-Kyu Chae
Hanyang University
Seoul, South Korea
dongkyu@hanyang.ac.kr

ACM Reference Format:
Seonggyeom Kim, Byeongtae Park, Harim Lee, and Dong-Kyu Chae. 2024. Supplementary for DgCD: Domain-guided Channel Dropout for Domain Generalization. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3627673.3679539

1 MORE DETAILS OF DGCD

1.1 The Upper Bound of Generalization Risk

Given multiple source domains, the generalization risk for an unseen target domain can be represented as the sum of the risk over the source domains and the discrepancy among the source domains [1]. This can be defined based on the $\mathcal{H}$-divergence [2], which measures the difference between two distributions, and is formulated as follows:

$$d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_U) = 2 \sup_{F \in \mathcal{H}} [\Pr_{x \sim \mathcal{D}_S}[F(x) = 1] - \Pr_{x \sim \mathcal{D}_U}[F(x) = 1]], \tag{1}$$

where $F : \mathcal{X} \rightarrow \{0, 1\}$ is a classifier with an input space $\mathcal{X}$. 'sup' represents the supremum, which is the smallest upper bound of a given set. Under the assumption that all source domains share the same set of labels, the convex hull Λ_S of $\mathcal{D}_S$, which is a set of mixture of source distributions, can be defined as [1]:

$$\Lambda_S = \left\{ \sum_{k=1}^{K} \pi_k \mathcal{D}_S^k | \pi \in \Delta_{K-1} \right\}, \tag{2}$$

where each π_k denotes a non-negative coefficient for a $(K-1)$-th dimensional simplex Δ_{K-1}.

Under the assumption that the target unseen domain $\mathcal{D}_U$ shares the same set of labels with the source domains, the ideal target $\bar{\mathcal{D}}_U$ is associated with the convex hull Λ_S, *i.e.*, $\bar{\mathcal{D}}_U \in \Lambda_S$. Grounded by this assumption, the generalization risk for $\mathcal{D}_U$ is bounded by:

$$R_U[F] \leq \sum_{k=1}^{K} \pi_k R_S^k[F] + \boldsymbol{\tau} + \boldsymbol{\zeta} + \min\{\mathbb{E}_{\mathcal{D}_u}[|F_{S_\pi} - F_U|], \mathbb{E}_{\mathcal{D}_u}[|F_U - F_{S_\pi}|]\}, \tag{3}$$

where $\boldsymbol{\tau}$ indicates $d_{\mathcal{H}}[\bar{\mathcal{D}}_U, \mathcal{D}_U]$, *i.e.*, the $\mathcal{H}$-divergence between the ideal unseen domain and the real unseen domain. $\boldsymbol{\zeta}$ represents $\sup_{D_S', D_S'' \in \Lambda_S} d_{\mathcal{H}\Delta\mathcal{H}}(D_S', D_S'')$, *i.e.*, the largest $\mathcal{H}$-divergence between any pair of the source domains. The last term $\min\{\cdot\}$ corresponds to the number of mismatches between the labeling functions F_S and F_U.

The upper bound can be simplified by taking several assumptions. First, in the DG scenarios where the covariate shift assumption holds [6], the mismatch between F_S and F_U reduces to zero (as all domains have the same labeling function). Next, under the assumption that the unseen domain $\mathcal{D}_U$ can be represented as a mixture of source domains $\mathcal{D}_S$ with weights π_k [14], $\mathcal{D}_U$ is equal to $\bar{\mathcal{D}}_U$, thus making $\boldsymbol{\tau}$ zero ($\mathcal{D}_U \in \Lambda_S, \mathcal{D}_U \in \bar{\mathcal{D}}_U$). In this way, Eq. (3) can be simplified as follows:

$$R_U[F] \leq \sum_{k=1}^{K} \pi_k R_S^k[F] + \boldsymbol{\zeta} \tag{2}$$

As a result, to minimize the above bound, DgCD aims to minimize $\sum_{k=1}^{K} \pi_k R_S^k[F]$ by ERM and minimize $\boldsymbol{\zeta}$ by finding out and dropping channels that yield higher discrepancy.

1.2 Channel Selection for Dropout

In our channel dropout strategy, we compute the probability of each channel being dropped as follows:

$$\mathbf{P}_i = \frac{\mathbf{S}_i}{\sum_{j=1}^{C} \mathbf{S}_{i,j}}, \tag{4}$$

where $\mathbf{P}_i$ represents the probability distribution of each channel being dropped in the i-th sample, and $\mathbf{S}_{i,c}$ denotes each channel's discrepancy. The sum of the elements in each $\mathbf{P}_i$ is 1. Among the methods for selecting channels to be dropped based on probability, naive WRS [7] could be a suitable candidate. However, it has a high time complexity of $O(C \times (r \times C))$. In contrast, the WRS algorithm adopted in this paper [9, 15] has a time complexity of $O(C)$, which significantly reduces computational costs. Algorithm 1 shows how we create a mask based on the WRS algorithm.

In addition to WRS, there are other alternative algorithms for probabilistically dropping channels. For instance, one method involves constructing a multinomial distribution with the probabilities of channels and performing sampling with replacement to select channel indices [20]. Another method involves calculating a removal score (gradient) for each neuron and extracting percentiles to select neurons with scores above a certain percentile [16]. However, we chose WRS, which is used in the most recent and our key paper (DomainDrop [9]). Experimentally, we tried other alternatives, but WRS showed the most stable performance.

*Both authors contributed equally to this research.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679539

Algorithm 1 Weighted Random Selection (WRS)

Input: S (channel-wise discrepancy), C (# channels), r (dropout ratio)
Output: $\mathbf{M}_i$ (mask for channel dropout in i-th sample)

Initialize empty set $\mathbf{K}_i$
Draw a random sample $\mathbf{R}_i$ from Uniform$(0, 1)$ of the same size as $\mathbf{S}_i$
Compute $\mathbf{K}_i = \mathbf{R}_i^{1/\Phi(\mathbf{S}_i)}$, where Φ is the min-max scaler
Sort $\{\mathbf{K}_{i,1}, \ldots, \mathbf{K}_{i,C}\}$ in descending order
Select the top $r \times C$ elements: $\mathbf{K}_{i,\text{top}} \leftarrow \text{Top}(\{\mathbf{K}_{i,1}, \ldots, \mathbf{K}_{i,C}\}, r \times C)$
Initialize $\mathbf{M}_i$ as a binary mask of size C with all elements set to 1
for all $c = 1$ to C **do**
  if $\mathbf{K}_{i,c} \in \mathbf{K}_{i,\text{top}}$ **then**
    Set $\mathbf{M}_{i,c}$ to 0
  end if
end for
Normalize $\mathbf{M}_i$ to have a mean of $(1 - r)$: $\hat{\mathbf{M}}_i = \mathbf{M}_i \times \frac{C}{\sum \mathbf{M}_i}$

1.3 Details of Distribution Discrepancy

In Section 4.4 of the main paper, we introduced an interesting work [19] suggesting that minimizing the KL-divergence between a source domain distribution and a prior distribution reduces the discrepancy in unseen domains. Based on this study, we defined the discrepancy as the KL-divergence between distributions $\mathcal{T}$ and $\mathcal{E}$, omitting the prior distribution from Eq. (16) of the main paper. The following statements elaborate more details about the KL-divergence between $\mathcal{T}$ and $\mathcal{E}$ without introducing the prior distribution.

Assumption 1. *We have sufficient data and $\mathcal{T}$ and $\mathcal{E}$ follow a Gaussian distribution.*

Given this assumption, the distributions of standardized data, $\mathcal{T}$ and $\mathcal{E}$, can be expressed as follows:

$$\mathcal{T} = \mathcal{N}(\mu_{\mathcal{T}}, \sigma^2_{\mathcal{T}}),\ \mu_{\mathcal{T}} = \frac{1}{N}\sum_{i=1}^{N} x_i,\ \sigma^2_{\mathcal{T}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\mathcal{T}})^2 \quad (5)$$

$$\mathcal{E}_k = \mathcal{N}(\mu_{\mathcal{E}_k}, \sigma^2_{\mathcal{E}_k}),\ \mu_{\mathcal{E}_k} = \frac{1}{N_k}\sum_{i=1}^{N_k} x_i,\ \sigma^2_{\mathcal{E}_k} = \frac{1}{N_k}\sum_{i=1}^{N_k}(x_i - \mu_{\mathcal{E}})^2$$
$$\mathcal{E} = \mathcal{N}(\mu_{\mathcal{E}}, \sigma^2_{\mathcal{E}}),\ \mu_{\mathcal{E}} = \sum_{k=1}^{K}\frac{1}{K}\mu_{\mathcal{E}_k},\ \sigma^2_{\mathcal{E}} = \sum_{k=1}^{K}\frac{1}{K}\left(\sigma^2_{\mathcal{E}_k} + \mu^2_{\mathcal{E}_k}\right) - \mu^2_{\mathcal{E}} \quad (6)$$

When represented as a probability density function, they can be re-written as $\mathcal{T}(x)$ and $\mathcal{E}(x)$:

$$\mathcal{T}(x) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{T}}}}\exp\left(-\frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}}\right),$$
$$\mathcal{E}(x) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{E}}}}\exp\left(-\frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}}\right) \quad (7)$$

Then, the discrepancy we defined can be expressed as follows:

$$D_{KL}(\mathcal{T}||\mathcal{E}) = \int \mathcal{T}(x)\log\frac{\mathcal{T}(x)}{\mathcal{E}(x)}\,dx \quad (8)$$

This KL-divergence can be re-written by breaking it down as follows:

$$\mathcal{T}(x) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{T}}}}\exp\left(-\frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}}\right),$$
$$\mathcal{E}(x) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{E}}}}\exp\left(-\frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}}\right),$$
$$\frac{\mathcal{T}(x)}{\mathcal{E}(x)} = \frac{\frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{T}}}}\exp\left(-\frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}}\right)}{\frac{1}{\sqrt{2\pi\sigma^2_{\mathcal{E}}}}\exp\left(-\frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}}\right)}$$
$$= \frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}}\exp\left(\frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}} - \frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}}\right),$$
$$\log\frac{\mathcal{T}(x)}{\mathcal{E}(x)} = \log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}} + \frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}} - \frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}} \quad (9)$$

$$D_{KL}(\mathcal{T}||\mathcal{E}) = \int \mathcal{T}(x)\left(\log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}} + \frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}} - \frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}}\right)dx \quad (10)$$

The above Eq. (10) can be simplified by:

$$\int \mathcal{T}(x)\log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}}\,dx = \log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}},$$
$$\int \mathcal{T}(x)\frac{(x-\mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}}\,dx = \frac{\sigma^2_{\mathcal{T}} + (\mu_{\mathcal{T}} - \mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}},$$
$$\int \mathcal{T}(x)\cdot -\frac{(x-\mu_{\mathcal{T}})^2}{2\sigma^2_{\mathcal{T}}}\,dx = -\frac{1}{2} \quad (11)$$

$$D_{KL}(\mathcal{T}||\mathcal{E}) = \log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}} + \frac{\sigma^2_{\mathcal{T}} + (\mu_{\mathcal{T}} - \mu_{\mathcal{E}})^2}{2\sigma^2_{\mathcal{E}}} - \frac{1}{2} \quad (12)$$

The KL divergence from the standard normal distribution, the prior distribution, is expressed as follows:

$$D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) = \frac{1}{2}\log\sigma^2_{\mathcal{T}} - \frac{1}{2} + \frac{\sigma^2_{\mathcal{T}} + \mu^2_{\mathcal{T}}}{2},$$
$$D_{KL}(\mathcal{E}||\mathcal{N}(0,1)) = \frac{1}{2}\log\sigma^2_{\mathcal{E}} - \frac{1}{2} + \frac{\sigma^2_{\mathcal{E}} + \mu^2_{\mathcal{E}}}{2} \quad (13)$$

Next, we aim to verify that the following relationship holds:

$$D_{KL}(\mathcal{T}||\mathcal{E}) = D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1)) \quad (14)$$

We validate Eq. (14) in two idealized scenarios (**Cases 1** and **2**), and then consider a more realistic case (**Case 3**).

Case 1. ***Idealized Distributions.*** *Suppose both distributions, $\mathcal{T}$ and $\mathcal{E}$, are standardized with a mean of zero and a standard deviation of one, i.e., $\mu_{\mathcal{T}} = \mu_{\mathcal{E}} = 0$, $\sigma_{\mathcal{T}} = \sigma_{\mathcal{E}} = 1$.*

$$\begin{aligned}
&D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1)) \\
&= \log\frac{\sigma_{\mathcal{T}}}{\sigma_{\mathcal{E}}} + \frac{(\sigma_{\mathcal{T}}^2 - \sigma_{\mathcal{E}}^2) + (\mu_{\mathcal{T}}^2 - \mu_{\mathcal{E}}^2)}{2} \\
&= \log\frac{1}{1} + \frac{1 - 1 + (0-0)}{2} \\
&= 0\,, \\
&D_{KL}(\mathcal{T}||\mathcal{E}) \\
&= \log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}} + \frac{\sigma_{\mathcal{T}}^2 + (\mu_{\mathcal{T}} - \mu_{\mathcal{E}})^2}{2\sigma_{\mathcal{E}}^2} - \frac{1}{2} \\
&= \log\frac{1}{1} + \frac{1 + (0-0)^2}{2\cdot 1} - \frac{1}{2} \\
&= 0
\end{aligned} \quad (15)$$

Case 2. ***Identical Mean and Standard Deviation.*** *Suppose $\mathcal{T}$ and $\mathcal{E}$ share the same mean and standard deviation, i.e., $\mu_{\mathcal{T}} = \mu_{\mathcal{E}} = \mu$, $\sigma_{\mathcal{T}} = \sigma_{\mathcal{E}} = \sigma$.*

$$\begin{aligned}
&D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1)) \\
&= \log\frac{\sigma_{\mathcal{T}}}{\sigma_{\mathcal{E}}} + \frac{(\sigma_{\mathcal{T}}^2 - \sigma_{\mathcal{E}}^2) + (\mu_{\mathcal{T}}^2 - \mu_{\mathcal{E}}^2)}{2} \\
&= \log\frac{\sigma}{\sigma} + \frac{(\sigma^2 - \sigma^2) + (\mu^2 - \mu^2)}{2} \\
&= 0\,, \\
&D_{KL}(\mathcal{T}||\mathcal{E}) \\
&= \log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}} + \frac{\sigma_{\mathcal{T}}^2 + (\mu_{\mathcal{T}} - \mu_{\mathcal{E}})^2}{2\sigma_{\mathcal{E}}^2} - \frac{1}{2} \\
&= \log\frac{\sigma}{\sigma} + \frac{\sigma^2 + (\mu - \mu)^2}{2\sigma^2} - \frac{1}{2} \\
&= 0
\end{aligned} \quad (16)$$

Case 3. ***Real-world Scenarios.*** *In real-world applications, $\mathcal{T}$ and $\mathcal{E}$ may not have exactly the same distributions. However, with a sufficiently large dataset, their means and variances will be quite similar, i.e., $\mu_{\mathcal{T}} \approx \mu_{\mathcal{E}}$ and $\sigma_{\mathcal{T}} \approx \sigma_{\mathcal{E}}$.*

$$\begin{aligned}
&D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1)) \\
&= \log\frac{\sigma_{\mathcal{T}}}{\sigma_{\mathcal{E}}} + \frac{(\sigma_{\mathcal{T}}^2 - \sigma_{\mathcal{E}}^2) + (\mu_{\mathcal{T}}^2 - \mu_{\mathcal{E}}^2)}{2} \\
&\approx \log\frac{\sigma_{\mathcal{E}}}{\sigma_{\mathcal{T}}} + \frac{\sigma_{\mathcal{T}}^2 + (\mu_{\mathcal{T}} - \mu_{\mathcal{E}})^2}{2\sigma_{\mathcal{E}}^2} - \frac{1}{2} \\
&= D_{KL}(\mathcal{T}||\mathcal{E})
\end{aligned} \quad (17)$$

Based on **Cases 1** and **2**, we have shown that **Case 3** holds by the *syllogism*. We now introduce an error term $\boldsymbol{\epsilon}$ and express Eq. (14) as an inequality, considering the need for approximation in real-world scenarios:

$$\begin{aligned}
&D_{KL}(\mathcal{T}||\mathcal{E}) - D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1)) \approx 0\,, \\
&||D_{KL}(\mathcal{T}||\mathcal{E})| - |D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1))|| \le |\boldsymbol{\epsilon}|\,.
\end{aligned} \quad (18)$$

For simplicity, we replace the first term ($D_{KL}(\mathcal{T}||\mathcal{E})$) with **A**, the second term with **B**, and the last term ($\boldsymbol{\epsilon}$) with **C**, as follows:

$$||\mathbf{A}| - |\mathbf{B}|| \le |\mathbf{C}|\,. \quad (19)$$

The above inequality can be re-written using the properties of absolute value and reverse triangle inequalities, as follows:

$$\begin{aligned}
& -|\mathbf{C}| \le |\mathbf{A}| - |\mathbf{B}| \le |\mathbf{C}| \\
\Rightarrow\ & -|\mathbf{C}| \le |\mathbf{A}| - |\mathbf{B}| \\
\Rightarrow\ & 0 \le |\mathbf{C}| + |\mathbf{A}| - |\mathbf{B}| \\
\Rightarrow\ & -|\mathbf{A}| \le |\mathbf{C}| - |\mathbf{B}| \\
\Rightarrow\ & -|\mathbf{A}| \le |\mathbf{C}| - |\mathbf{B}| \le |\mathbf{A}| \\
\Rightarrow\ & ||\mathbf{C}| - |\mathbf{B}|| \le |\mathbf{A}|.
\end{aligned} \quad (20)$$

Finally, $||\mathbf{C}| - |\mathbf{B}|| \le |\mathbf{A}|$ can be again re-written by:

$$||\boldsymbol{\epsilon}| - |D_{KL}(\mathcal{T}||\mathcal{N}(0,1)) - D_{KL}(\mathcal{E}||\mathcal{N}(0,1))|| \le |D_{KL}(\mathcal{T}||\mathcal{E})|\,, \quad (21)$$

where $D_{KL}(\mathcal{T}||\mathcal{E})$ represents the upper bound on the difference between $D_{KL}(\mathcal{T}||\mathcal{N}(0,1))$ and $D_{KL}(\mathcal{E}||\mathcal{N}(0,1))$. Therefore, the term $D_{KL}(\mathcal{T}||\mathcal{E})$ can be used to quantify the discrepancy between two distributions, taking into account differences in their means and standard deviations. If the distributions $\mathcal{T}$ and $\mathcal{E}$ are identical, then $D_{KL}(\mathcal{T}||\mathcal{E}) = 0$. Conversely, $D_{KL}(\mathcal{T}||\mathcal{E})$ will increase if there are significant differences between the two distributions, particularly in their means and standard deviations across subsets of $\mathcal{E}$. As a result, the $D_{KL}(\mathcal{T}||\mathcal{E})$ term enables us to quantify the distribution discrepancy, particularly under the situation where we deal with sufficient amount of data and we try to approximate the two distributions to be as identical as possible.

The introduction of the $D_{KL}(\mathcal{T}||\mathcal{E})$ term was based on the assumption that a given data follows a normal distribution, which may not hold for many real-world scenarios. Therefore, further research is needed for other types of distributions. Nevertheless, we empirically observed that minimizing $D_{KL}(\mathcal{T}||\mathcal{E})$ successfully reduced the discrepancy between source domains as well as the discrepancy between unseen domains and source domains (see Section 5.4.1

Table 1: Hyperparameter values used for each dataset.

Hyperparameter	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet
learning rate	1-e5	2-e5	1-e6	1-e5	5-e5
weight decay	1-e6	5-e6	5-e6	1-e6	1-e6
max step	5000	5000	5000	5000	20000
frequency of evaluation	100	100	100	100	500
ρ	0.01	0.05	0.05	0.1	0.01
r	0.33	0.3	0.33	0.33	0.33

of the main paper and Section 2.3 of this material). These results suggest that $D_{KL}(\mathcal{T}||\mathcal{E})$ may be an important factor in domain generalization, and we plan to explore more robust generalization methods by closely analyzing it in our future research.

2 DETAILS OF THE EXPERIMENTAL SETTINGS

Training and Evaluation protocol. We basically follow the training and evaluation protocols presented in DomainBed [8]. Our implementation is built upon the code[1] released by the authors of MIRO [4]. For training, we select one domain as the unseen target domain and consider the rest as source domains. We split each source domain 8:2 (training/validation) and measure the performance on the target domain at the training step with the highest validation accuracy for model selection.

In addition, for the experiments introduced in Section 5.4 of the main paper, we applied DgCD on top of the code[2] and hyperparameter settings of DomainDrop [9] for a fair comparison: the batch size is set to 64, the maximum number of epochs to 50, the initial learning rate to 0.002 and is decreased by 0.1 after 80% of the total epochs. We used an SGD optimizer with a momentum of 0.9 and weight decay of 0.0005. The split for training and validation was set to 7:3. The hyperparameters unique for DomainDrop and PLACE [10][3] were used as presented in their official code.

Hyperparameters. We follow the hyperparameter search protocol proposed by the authors of DNA [5]: For the hyperparameters of DgCD, the dropout ratio r is searched in [0.25, 0.3, 0.33, 0.4], and the update rate ρ is searched in [0.01, ..., 0.1]. Other hyperparameters, such as the learning rate, were searched in [1e-5, 2e-5, 3e-5, 4e-5, 5e-5], and the weight decay was searched in [1e-6, 3e-6, 5e-6]. Rather than searching independently for each unseen domain, we use the hyperparameters found in the first random data split. Table 1 summarizes the hyperparameter values used for each dataset. There are slight differences between datasets, but most share the same values. The model is optimized using Adam [17] optimizer. We de-activate the ResNet dropout flag specified in DomainBed. The number of samples drawn from each source domain is set to 32: if the number of source domains is 3, the total batch size is 96, which is the default setting of DomainBed.

[1]https://github.com/kakaobrain/miro
[2]https://github.com/lingeringlight/DomainDrop
[3]https://github.com/lingeringlight/PLACEdropout

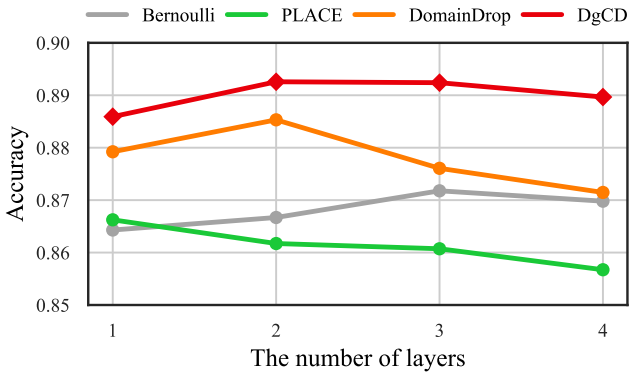


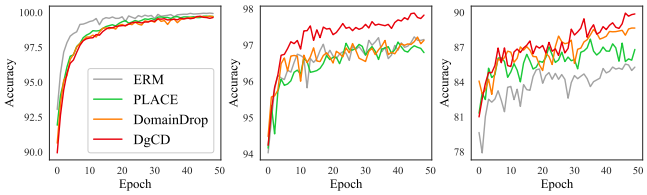
Figure 1: Performance depending on the number of randomly selected layers. PACS is used here.

2.1 The Number of Dropout Layers

Figure 1 shows that DgCD is not significantly affected by the number of randomly selected layers, while the existing dropout-based approaches PLACE [10] and DomainDrop [9] tend to decrease in performance as the number of layers increases. This indicates that DgCD, as a statistical approach, is stable even as the number of discarded features increases because it effectively reduces domain discrepancies that hinder generalization. In contrast, PLACE and DomainDrop indicate that they may lead to the wrong decision of discarding important features related to the label. The traditional dropout approach (Bernoulli) is stable to the number of layers because it discards features completely randomly by the Bernoulli distribution, albeit with lower accuracy.

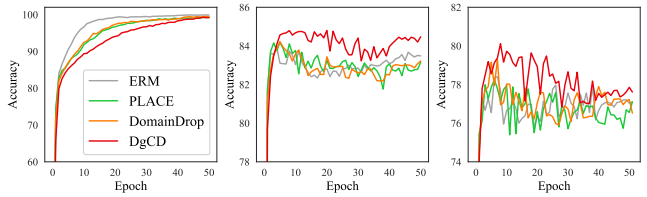
2.2 Accuracy for Training, Validation, and Test Sets

For another quantitative evaluation of covariate shift, we measured training, validation, and test accuracy. Figures 2, 3, and 4 (a), (b), and (c) show the respective results. ERM showed the highest training accuracy in (a), but it exhibited the lowest test accuracy in (c), indicating its weakness to domain shift. PLACE and DomainDrop showed higher test accuracy than ERM in (c), suggesting a somewhat higher robustness to domain shift than ERM, but similar validation accuracy to ERM in (b). DgCD achieved the highest accuracy in both validation and test sets. This confirms that DgCD



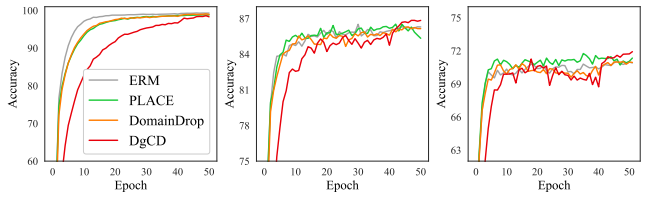
(a) Training (b) Validation (c) Test

Figure 2: Training, validation and test accuracy on PACS.



(a) Training (b) Validation (c) Test

Figure 3: Training, validation and test accuracy on VLCS.



(a) Training (b) Validation (c) Test

Figure 4: Training, validation and test accuracy on Office-Home.

is robust to covariate and domain shift, outperforming existing dropout-based DG methods.

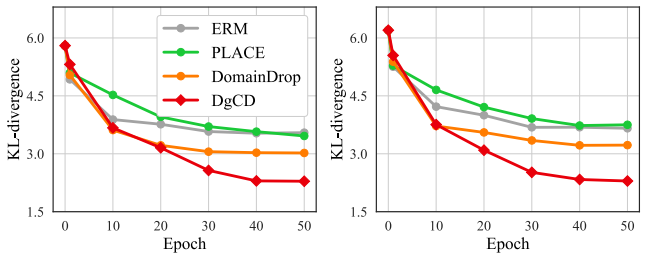
2.3 Single-Source Domain Generalization

In the main paper, we evaluate DgCD's performance on the single-source DG task using the PACS. Here, we use the ImageNet Robustness benchmarks: we used Tiny ImageNet [18] as the single source domain, and its shifted versions, Tiny ImageNet-R [11] and Tiny ImageNet-C [12] as the test unseen domains. Tiny ImageNet-R (Renditions) includes various domains such as cartoons, deviantart, patterns, embroidery, graphics and plastic objects, generated by style changes on the ImageNet images. Tiny ImageNet-C (Corruption) comprises generated corruptions such as Gaussian noise and blur. Following [3, 21], we used *top-1 accuracy* for evaluation on Tiny ImageNet-R and *mean corruption error* (mCE) [13] for Tiny ImageNet-C. Both metrics are computed based on the top-1 classification performance, but the first metric indicates better generalization when it is higher, while the second metric indicates better performance when it is lower.

When training with Tiny ImageNet, we set the batch size to 96, and applied our algorithm by splitting the input batch into three pieces and reducing their distribution discrepancies. Table 2 reports the results. On the unseen test sets, Tiny ImageNet-R and Tiny ImageNet-C, DgCD outperforms the compared baselines (DomainDrop is not compared because it requires domain labels).

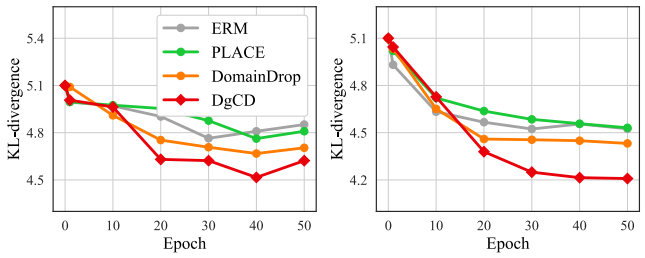
Table 2: Tiny ImageNet robustness benchmarks. We show the single domain generalization performance on Tiny ImageNet-C and Tiny ImageNet-R.

Method	Tiny ImageNet (source, top-1 acc.↑)	Tiny ImageNet-R (target, top-1 acc.↑)	Tiny ImageNet-C (target, mCE ↓)
ERM	**54.4**	12.7	76.2
Bernoulli	51.8	13.5	73.8
PLACE	51.6	13.5	73.5
DgCD	53.0	**13.7**	**73.1**



(a) Source - Source (b) Source - Target

Figure 5: (a): The avg. KL-divergence between pairs of the source domains in VLCS, derived by the output embedding of the ResNet backbone. (b): The avg. KL-divergence between the source domains and the unseen target domain.



(a) Source - Source (b) Source - Target

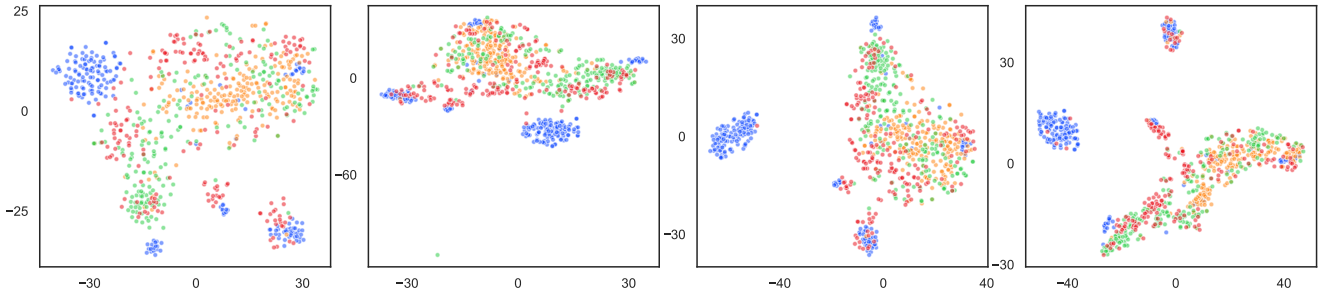
Figure 6: (a): The avg. KL-divergence between pairs of the source domains in OfficeHome, derived by the output embedding of the ResNet backbone. (b): The avg. KL-divergence between the source domains and the unseen target domain.

2.4 Domain Discrepancy

Figures 5 and 6 show the results of measuring domain discrepancy using KL-divergence in VLCS and OfficeHome, respectively. Similar to the results shown in our main paper for PACS, our DgCD demonstrated the lowest KL-divergence.

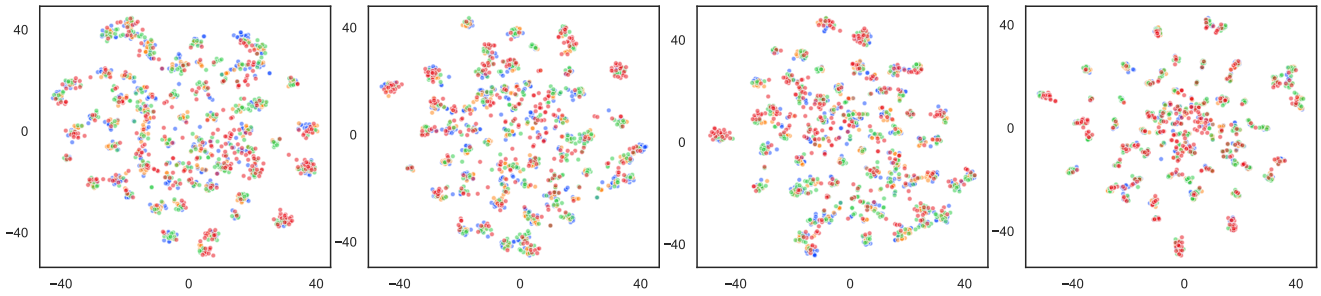
2.5 Robustness to Covariate Shift

Figures 7 and 8 visualize the embeddings for VLCS and OfficeHome using T-SNE. In VLCS, which has 4 classes, any method failed to form clear clusters. In contrast, in OfficeHome, which has 65 classes, the clusters formed by our DgCD were the most distinct.



(a) ERM (b) PLACE (c) DomainDrop (d) Ours

Figure 7: T-SNE visualization of the last pooling layer output of the ResNet, where the red points correspond to the unseen domain (Pascal) and the other colored points represent the source domains. The four clusters formed by each model correspond to each class label in VLCS.



(a) ERM (b) PLACE (c) DomainDrop (d) Ours

Figure 8: T-SNE visualization of the last pooling layer output of the ResNet, where the red points correspond to the unseen domain (Art) and the other colored points represent the source domains. The 65 clusters formed by each model correspond to each class label in OfficeHome.

REFERENCES

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

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79 (2010), 151–175.

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

[4] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. 2022. Domain Generalization by Mutual-Information Regularization with Pre-trained Models. In *European Conference on Computer Vision*. 440–457.

[5] Xu Chu, Yujie Jin, Wenwu Zhu, Yasha Wang, Xin Wang, Shanghang Zhang, and Hong Mei. 2022. DNA: Domain generalization with diversified neural averaging. In *International Conference on Machine Learning*. PMLR, 4010–4034.

[6] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. 2010. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 129–136.

[7] Pavlos S Efraimidis and Paul G Spirakis. 2006. Weighted random sampling with a reservoir. *Information processing letters* 97, 5 (2006), 181–185.

[8] Ishaan Gulrajani and David Lopez-Paz. 2020. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*.

[9] Jintao Guo, Lei Qi, and Yinghuan Shi. 2023. Domaindrop: Suppressing domain-sensitive channels for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19114–19124.

[10] Jintao Guo, Lei Qi, Yinghuan Shi, and Yang Gao. 2023. PLACE Dropout: A Progressive Layer-wise and Channel-wise Dropout for Domain Generalization. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 3 (2023), 1–23.

[11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*. 8340–8349.

[12] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261* (2019).

[13] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261* (2019).

[14] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. 2018. Algorithms and theory for multiple-source adaptation. *Advances in neural information processing systems* 31 (2018).

[15] Saihui Hou and Zilei Wang. 2019. Weighted channel dropout for regularization of deep convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8425–8432.

[16] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging Improves Cross-Domain Generalization. In *European Conference on Computer Vision*. 124–140.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Ya Le and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7, 7 (2015), 3.

[19] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. 2020. Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in neural information processing systems* 33 (2020), 3118–3129.

[20] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. 2020. Informative dropout for robust representation learning: A shape-bias perspective. In *International Conference on Machine Learning*. PMLR, 8828–8839.

[21] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. 2023. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3769–3778.