A. Theoretical Proof of Theorem 1

A.1. Notations

Given K source domains $\mathcal{D}_s = \{D_s^1, D_s^2, ..., D_s^K\}$, we indicate that each domain D_s^k contains n_k input and labels $\{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The target domain is denoted as $\mathcal{D}_t$. Given a hypothesis $h : \mathcal{X} \to \mathcal{Y}$, where h is from the space of the candidate hypothesis $\mathcal{H}$. The expected risk of h on a domain D is defined as: $\mathcal{R}[h] = \mathbb{E}_{x\sim D}\ell[h(x), f(x)]$, where $\ell_{h,f} : x \to \ell[h(x), f(x)]$ is a convex loss-function defined for $\forall h, f \in \mathcal{H}$ and assumed to obey the triangle inequality. Under the DG set, $y = f(x)$ represents the input label. We also denote the feature extractor of the network as $\phi(\cdot) : \mathcal{X} \to \mathbb{R}^n$, which maps the input images into the n-dimentional feature space. Following [1, 4, 26, 25], for the source domains $\mathcal{D}_s = \{D_s^1, D_s^2, ..., D_s^K\}$, we define the convex hull Λ_s as a set of mixture of source domain distributions: $\Lambda_s = \{\bar{D} : \bar{D}(\cdot) = \sum_{i=1}^K \pi_i D_s^i(\cdot), \pi_i \in \Delta_K\}$, where π is non-negative coefficient in the K-dimensional simplex Δ_K. We define $\bar{D}_t \in \Lambda_s$ as the closest domain to the target domain D_t.

A.2. Definitions and Lemmas

Definition 1 [16]. *Let $\mathcal{F} = \{f \in \mathcal{H}_k : ||f||_{\mathcal{H}_k} \leq 1\}$ be a function class, where $\mathcal{H}_k$ be a RKHS with its associated kernel k. Given two different distributions of D_s and D_t, the maximum mean discrepancy (MMD) distance is:*

$$d_{\text{MMD}}(D_s, D_t) = ||\int_x k(x, \cdot) d(\phi(D_s) - \phi(D_t))||_{\mathcal{H}_k}. \tag{1}$$

Based on the MMD distance, we now introduce learning bounds for the target error where the divergence between distributions is measured by the MMD distance. We first introduce a lemma that indicates how the target error can be bounded by the empirical estimate of the MMD distance between an arbitrary pair of source and target domains.

Lemma 1 [16]. *Let $\mathcal{F} = \{f \in \mathcal{H}_k : ||f||_{\mathcal{H}_k} \leq 1\}$ denote a function class, where $\mathcal{H}_k$ be a RKHS with its associated kernel k. Let $\ell_{h,f} : x \to \ell[h(x), f(x)]$ be a convex loss-function with a parameter form $|h(x) - f(x)|^q$ for some $q > 0$, and defined $\forall h, f \in \mathcal{F}$, ℓ obeys the triangle inequality. Let S and T be two samples of size m drawn i.i.d from D_s and D_t, respectively. Then, with probability of at least $1 - \delta$ ($\delta \in (0, 1)$) for all $h \in \mathcal{F}$, the following holds:*

$$\begin{aligned}\mathcal{R}_t[h] \leq &\mathcal{R}_s[h] + d_{\text{MMD}}(D_t, D_s) + \frac{2}{m}(E_{x\sim D_s}[\sqrt{tr(K_{D_s})}] \\ &+ E_{x\sim D_t}[\sqrt{tr(K_{D_t})}]) + 2\frac{\log(\frac{2}{\sigma})}{2m} + \epsilon,\end{aligned} \tag{2}$$

where K_{D_s} and K_{D_t} are kernel functions computed on samples from D_s and D_t, respectively. ϵ is the combined error of the ideal hypothesis h^* on D_s and D_t.

Then, to investigate the effect of channel robustness to domain shifts on the generalization error bound, we define the channel-level maximum mean discrepancy (CMMD) distance to estimate the channel-level distribution gap between different domains, which is formulated as:

Definition 2. *Let n denote the number of channels in the extracted features of $\phi(\cdot)$. Given two different distribution of D_s and D_t, the channel-level maximum mean discrepancy (CMMD) between $\phi(D_s)$ and $\phi(D_t)$ is defined as:*

$$\begin{aligned}d_{\text{CMMD}}(D_s, D_t) = \frac{1}{n}\sum_{i=1}^{n} \sup_{\phi_i \in \Phi_i} ||\int_x & k(x, \cdot) d(\phi_i(D_s) \\ &- \phi_i(D_t))||_{\mathcal{H}_k},\end{aligned} \tag{3}$$

where Φ is the space of candidate hypothesis for each channel, $\phi_i(D)$ is the distribution of the i-th channel for the domain D, and $\mathcal{H}_k$ is a RKHS with its associated kernel k.

The CMMD distance could be regarded as a channel-level version of the MMD distance, which represents the maximum value of the difference in channel activation for a given two domains in the model, thus reflecting the channel robustness to domain shifts. Based on the CMMD distance and Lemma 1, we derive a generalization error boundary of the model in the multi-source domain scenario (*i.e.*, Theorem 1), and provide the detailed proof below.

A.3. Proof

Theorem 1 (Generalization risk bound). *With the previous settings and assumptions, let S^i and T be two samples of size m drawn i.i.d from D_s^i and D_t, respectively. Then, with the probability of at least $1 - \delta$ ($\delta \in (0, 1)$) for all $h \in \mathcal{F}$, the following inequality holds for the risk $\mathcal{R}_t[h]$:*

$$\begin{aligned}\mathcal{R}_t[h] \leq &\sum_{i=1}^{N} \pi_i \mathcal{R}_s^i[h] + d_{\text{CMMD}}(\bar{D}_t, D_t) \\ &+ \sup_{i,j\in[K]} d_{\text{CMMD}}(D_s^i, D_s^j) + \lambda + \epsilon,\end{aligned} \tag{4}$$

where $\lambda = 2\sqrt{\frac{\log(\frac{2}{\sigma})}{2m}} + \frac{2}{m}(\sum_{i=1}^N \pi_i E_{x\sim D_s^i}[\sqrt{tr(K_{D_s^i})}] + E_{x\sim D_t}[\sqrt{tr(K_{D_t})}])$, $K_{D_s^i}$ and K_{D_t} are kernel functions computed on samples from D_s^i and D_t, respectively. ϵ is the combined error of ideal hypothesis h^* on D_t and $\bar{D}_t$.

Proof. Consider the closest domain $\bar{D}_t$ to target domain D_t as a mixture distribution of K source domains where the mixture weight is given by π, *i.e.*, $\bar{D}_t = \sum_{i=1}^K \pi_i D_s^i(\cdot)$ with $\sum_{i=1}^K \pi_i = 1$. For a pair of source domain D_s^i and the target domain D_t, the following inequality holds:

$$d_{\text{CMMD}}(D_t, D_s^i) \leq d_{\text{CMMD}}(D_t, \bar{D}_t) + d_{\text{CMMD}}(\bar{D}_t, D_s^i). \tag{5}$$

According to Definition 2, we could derive the weighted sum of the CMMD distance between source domains and the target domain, which is formulated as:

$$\begin{aligned}\sum_{i=1}^{N} &\pi_i d_{\text{CMMD}}(D_t, D_s^i) \\ &\leq d_{\text{CMMD}}(D_t, \bar{D}_t) + \sum_{i=1}^{N} \pi_i d_{\text{CMMD}}(\bar{D}_t, D_s^i) \\ &\leq d_{\text{CMMD}}(D_t, \bar{D}_t) + \sup_{i,j \in \mathcal{H}} d_{\text{CMMD}}(D_s^i, D_s^j).\end{aligned} \tag{6}$$

Moreover, we also investigate the relationship between the MMD and CMMD distances based on Definitions 1 and 2:

$$\begin{aligned} d_{\text{MMD}}(D_s^i, D_t) &= || \int_x k(x, \cdot) d(\phi(D_s^i) - \phi(D_t)) ||_{\mathcal{H}_k} \\ &= || \int_x k(x, \cdot) d(\frac{1}{n} \sum_{i=1}^{n} (\phi_i(D_s^i) - \phi_i(D_t))) ||_{\mathcal{H}_k} \\ &\leq || \int_x k(x, \cdot) \sum_{i=1}^{n} \sup_{\phi_i \in \Phi_i} d(\phi_i(D_s^i) - \phi_i(D_t)) ||_{\mathcal{H}_k} \\ &= d_{\text{CMMD}}(D_s^i, D_t). \end{aligned} \tag{7}$$

Based on the above preparations, we now derive the generalization error bound of the model on the unseen target domain. Recalling that Lemma 1 indicates the generalization error bound between two different distributions. Considering the pair of the i-th source domain and the target domain, the following holds with the probability of at least $1 - \delta$:

$$\begin{aligned} \mathcal{R}_t[h] \leq& \mathcal{R}_s^i[h] + d_{\text{CMMD}}(D_t, D_s^i) + \frac{2}{m}(E_{x \sim D_s^i}[\sqrt{tr(K_{D_s^i})}] \\ &+ E_{x \sim D_t}[\sqrt{tr(K_{D_t})}]) + 2\frac{\log(\frac{2}{\sigma})}{2m} + \epsilon. \end{aligned} \tag{8}$$

We then generalize the above inequality to the multi-source scenario, where the ideal target domain could be expressed as a weighted combination of different source domains. We weight the generalization error of each source-target pair with π where $\sum_{i=1}^{K} \pi_i = 1$ and calculate their sum:

$$\begin{aligned} \mathcal{R}_t[h] \leq& \sum_{i=1}^{N} \pi_i \mathcal{R}_s^i[h] + \sum_{i=1}^{N} \pi_i d_{\text{CMMD}}(D_t, D_s^i) \\ &+ \frac{2}{m}(\sum_{i=1}^{N} \pi_i E_{x \sim D_s^i}[\sqrt{tr(K_{D_s^i})}] \\ &+ E_{x \sim D_t}[\sqrt{tr(K_{D_t})}]) + 2\frac{\log(\frac{2}{\sigma})}{2m} + \epsilon. \end{aligned} \tag{9}$$

By replacing the CMMD distance in Eq. (9) with the retracted CMMD distance in Eq. (6), we arrive at Theorem 1.

B. Additional Experiments

We conduct additional experiments to verify the effectiveness of our DomainDrop, including: 1) The effects of different inserted positions of DomainDrop in the network; 2) The experiments on the DomainBed benchmark.

Table 1. Effect (%) on different inserted posotions of DomainDrop. B1 − 4 represent four residual blocks of the ResNet architecture. The experiment is conducted on PACS dataset with ResNet-18 backbone. The best performance is marked as **bold**.

Position				PACS				
B1	B2	B3	B4	Art	Cartoon	Photo	Sketch	Avg.
-	-	-	-	80.31 ± 1.54	76.65 ± 0.48	95.38 ± 0.12	71.67 ± 1.49	81.00
✓	-	-	-	81.10 ± 0.76	78.88 ± 0.69	94.72 ± 0.45	81.92 ± 0.69	84.15
-	✓	-	-	80.71 ± 0.71	79.25 ± 0.44	94.85 ± 0.35	82.16 ± 1.35	84.24
-	-	✓	-	82.52 ± 0.72	79.44 ± 0.46	95.76 ± 0.16	79.35 ± 1.17	84.27
-	-	-	✓	81.15 ± 0.98	78.58 ± 0.81	95.39 ± 0.40	79.74 ± 1.47	83.72
✓	✓	-	-	81.15 ± 1.03	79.44 ± 0.30	95.99 ± 0.49	83.13 ± 0.48	84.93
✓	✓	✓	-	83.84 ± 0.70	80.02 ± 0.37	96.29 ± 0.23	83.23 ± 0.53	85.87
✓	✓	✓	✓	**84.47** ± 0.77	**80.50** ± 0.56	**96.83** ± 0.21	**84.83** ± 0.67	**86.66**

Table 2. Performance (%) comparisons with the start-of-the-art DG approaches on the DomainBed benchmark. We compare with 12 DG algorithms on the following five multi-domain datasets: VLCS [20], PACS [12], OfficeHome [21], TerraInc [2], and DomainNet [15]. The network architecture is ResNet-50. We use the validation set from source domains for the model selection.

Method	Venue	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg.
ERM [6]	ICLR'20	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
RSC [8]	ECCV'20	77.1 ± 0.5	85.2 ± 0.9	65.5 ± 0.9	46.6 ± 1.0	38.9 ± 0.5	62.7
SagNet [14]	CVPR'21	77.8 ± 0.5	86.3 ± 0.2	68.1 ± 0.1	48.6 ± 1.0	40.3 ± 0.1	64.2
SelfReg [9]	ICCV'21	77.5 ± 0.0	86.5 ± 0.3	69.4 ± 0.2	51.0 ± 0.4	44.6 ± 0.1	65.8
FISH [19]	ICLR'21	77.8 ± 0.3	85.5 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	42.7 ± 0.2	63.9
W2D [7]	CVPR'22	-	83.4 ± 0.3	63.5 ± 0.1	44.5 ± 0.5	-	-
XDED [10]	ECCV'22	74.8 ±0.0	83.8 ± 0.0	65.0 ± 0.0	42.5 ± 0.0	-	-
GVRT [13]	ECCV'22	79.0 ± 0.2	85.1 ± 0.3	70.1 ± 0.1	48.0 ± 0.2	44.1 ± 0.1	65.2
MIRO [3]	ECCV'22	79.0 ± 0.0	85.4 ± 0.4	**70.5** ± 0.4	50.4 ± 1.1	44.3 ± 0.2	65.9
PTE [13]	ECCV'22	79.0 ± 0.2	85.1 ± 0.3	70.1 ± 0.1	48.0 ± 0.2	44.1 ± 0.1	65.2
EQRM [5]	NeurIPS'22	77.8 ± 0.6	86.5 ± 0.2	67.5 ± 0.1	47.8 ± 0.6	41.0 ± 0.3	64.1
DAC-SC [11]	CVPR'23	78.7 ± 0.3	87.5 ± 0.1	70.3 ± 0.2	44.9 ± 0.1	**46.5** ± 0.3	65.6
DomainDrop	Ours	**79.8** ± 0.3	**87.9** ± 0.3	68.7 ± 0.1	**51.5** ± 0.4	44.4 ± 0.5	**66.5**

Different inserted positions of DomainDrop. We here investigate where to insert DomainDrop in the network. Given a standard ResNet with four residual blocks, we train different models by taking different blocks as candidates and randomly selecting a block to activate DomainDrop at each iteration. The results are reported in Tab. 1. The first line represents the results of the baseline model, which is trained using all source domains directly on the ResNet-18 (*i.e.*, DeepAll [27]). We observe that no matter where DomainDrop is inserted, the model consistently outperforms the baseline model by a significant margin, *e.g.*, 3.15% (84.15% vs. 81.00%) with DomainDrop in Block 1. The results indicate that our DomainDrop is effective in enhancing the robustness of channels to domain shifts at different network layers. Furthermore, we find that inserting DomainDrop into all blocks of the network leads to the highest performance, exceeding the baseline model by 5.66% (86.66% vs. 81.00%), indicating that suppressing domain-sensitive channels in all training stages will result in the best generalization ability. Based on the analysis, we insert DomainDrop into all network blocks in our all experiments.

Experiments on DomainBed. We conducted experiments on the DomainBed benchmark [6], including VLCS, PACS, OfficeHome, TerraInc, and DomainNet. The network is trained using Adam optimizer for 5000 iterations with a learning rate of $5e-5$ and batch size of 64. The experiments are repeated three times, and the averaged accuracy is reported in Tab. 2. We observe that our DomainDrop can consistently achieve better performance than ERM (a strong baseline in DomainBed) on all datasets, *e.g.*, outperforming ERM by 2.4% (87.9% vs. 85.5%) on PACS and 5.4% (51.5% vs. 46.1%) on TerraInc. The experimental results demonstrate the effectiveness of our method on various DG benchmark datasets. Moreover, DomainDrop obtained the highest average accuracy among all the compared methods, exceeding the SOTA method DAC-SC [11] by 0.9% (66.5% vs. 65.6%), indicating that our method can significantly improve the model generalization ability.

C. Analytical Experiments

We conduct experiments to analyze the effectiveness of our method, including: 1) We discuss why tackle the DG issue on feature channels; 2) We quantify the channel robustness to domain shifts in each network layer; 3) We measure the domain gap of feature maps extracted by the model; 4) We provide visual explanations of our DomainDrop.

Why tackle DG on feature channels. Different from traditional DG methods that constrain the entire network, recent methods have focused on learning domain-invariant features in middle layers via domain augmentations [23, 28] or local penalizations [18, 22]. However, recent work [4] has indicated that these methods typically perturb or penalize specific pre-defined features, *e.g.*, style statistics [28] or local textures [18], which could neglect other domain-specific features and affect model generalization. In this paper, we propose to analyze the DG issue from a novel perspective of channel robustness to domain shifts. *Our key insight is that if a channel captures domain-invariant patterns, its activations should remain stable across different domains.* As shown in Fig. 1, we observe that numerous channels exhibit limited robustness to domain shifts (*i.e.*, the red bars). The findings motivate us to focus on enhancing channel robustness to domain shifts.

Channel robustness to domain shifts. To enhance the generalization ability of the models to the unseen target domain, we wish the model to learn general and comprehensive domain-invariant features from source domains. Ideally, we hope each channel of the representations is activated by category-related information while being invariant across domains, making the whole representation sufficient for classification. Inspired by previous work [23], we exploit the averaged activation for each class in each domain to estimate the robustness of each channel to domain shifts. Specifically, for the i-th channel in the l-th middle layer, we first calculate its averaged activation in the k-the domain:

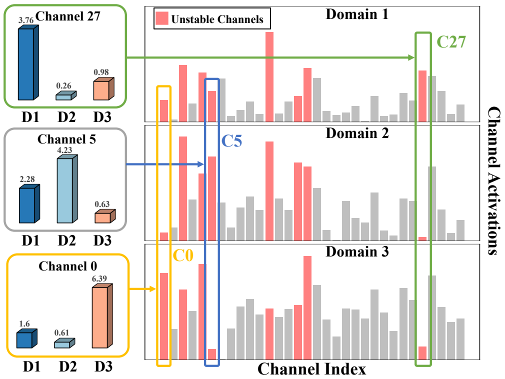

Figure 1. The activation value of the channels $1-32$ in the last block of ResNet-18 across different domains. The experiment is conducted on the PACS dataset with Art as the target domain.

Table 3. The standard deviation of channel activations for samples from different domains. Block. $1-4$ represent four residual blocks of the ResNet architecture. The lower the standard deviation, the more robust the channel is to domain shifts.

Sensitivity	Block. 1	Block. 2	Block. 3	Block. 4
Baseline	4.43	2.06	1.54	7.84
RSC [8]	4.22	1.94	1.63	7.23
I^2-Drop [18]	4.03	1.89	1.58	7.42
MixStyle [28]	4.30	2.03	1.51	7.16
FACT [24]	4.83	2.07	1.57	7.52
DomainDrop (Ours)	**3.85**	**1.56**	**1.04**	**5.94**

$$a_i^l = \frac{1}{n_k}\sum_{j=1}^{n_k} GAP(F_l(x_j))_i, \tag{10}$$

where $F_l(\cdot)$ is the feature maps in the l-th middle layer and $GAP(\cdot)$ denotes the global average pooling layer. Then we compute the standard deviation of the i-th channel activation among different domains. We present the results in Tab. 3. We observe that compared with Baseline, RSC [8] and I^2-Drop [18] present lower channel sensitivity to domain shifts in the last layer (*i.e.*, Block. 4) since they can regularize the model to learning domain-invariant features. However, since these methods are only suitable for specific layers (*i.e.*, RSC for the deepest layer and I^2-Drop for the shallowest layer), they cannot adequately counter the overfitting issue. The SOTA DG methods MixStyle [28] can increase the feature diversity at multiple layers, but it does not explicitly remove domain-specific features, thus failing to reduce channel sensitivity adequately. In contrast, the lowest standard deviation that DomainDrop achieves indicates that our method can learn more domain-invariant representations, showing the superiority of our framework.

Domain gap of extracted features maps. To investigate the influence of our framework, we also calculate the interdomain distance (across all source domains) of the feature maps extracted by the model on various datasets, includ-

Table 4. The inter-domain distribution gap (×100) of the extracted features by our method. For PACS, we take Art Painting as the target domain and the others as all source domains. For Office-Home, the target domain is Real-World and the others are source domains. For VLCS, we adopt Sun as the target domain and the others as source domains. The smaller the inter-domain distance, the better the generalization performance of the model.

Method	PACS	OfficeHome	VLCS
Baseline	17.57	11.56	16.65
DomainDrop (Ours)	**11.82**	**8.58**	**14.21**

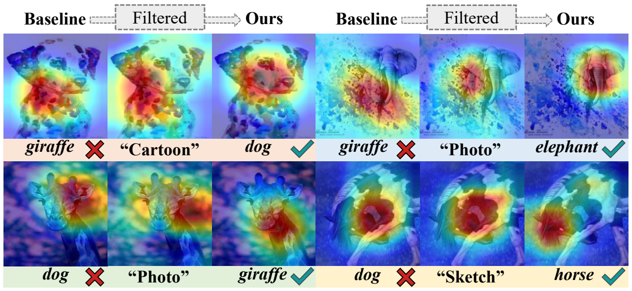

Figure 2. Visualization of attention maps of the last convolutional layer on PACS with Art Painting as the target domain. The backbone used in the experiment is ResNet-18. For each sample, the first column is the category attention map of baseline, the middle column is the domain attention map generated by domain discriminator, and the last column is the attention map of DomainDrop.

ing PACS, OfficeHome, and VLCS. Following previous DG method [23], we calcute the inter-domain gap as:

$$d = \frac{2}{K(K-1)} \sum_{k_1=1}^{K} \sum_{k_2=k_1+1}^{K} ||\overline{F}_{k_1} - \overline{F}_{k_2}||_2, \qquad (11)$$

where K is the number of source domains, $\overline{F}_{k_1}$ and $\overline{F}_{k_2}$ denote the averaged feature maps of all samples from the k_1-th and k_2-th domain, respectively. As shown in Tab. 4, we can observe that compared to the baseline, DomainDrop can effectively narrow the inter-domain gap among source domains on all datasets, indicating that our method can suppress domain-specific features and encourage the model to learn domain-invariant features during training.

Visual explanations. To provide visual evidence of the effectiveness of DomainDrop in reducing domain-specific features, we utilized GradCAM [17] to generate attention maps of the last conventional layer for both the baseline (DeepAll) and DomainDrop models. The results are presented in Fig. 2. As we can see, the baseline model captures a considerable amount of domain-specific information, as indicated by the overlap between the category attention map (column 1) and the domain attention map (column 2). On the other hand, DomainDrop can discard domain-specific features while retaining domain-invariant features, leading to more generalized attention maps that focus on representative information for object classification (column 3). For instance, in the case of the dog image, the model needs to focus on the dog's face as one of the representative features to classify, which is precisely captured by DomainDrop. In contrast, the baseline focuses on spot texture features, which results in misclassification. These results suggest that DomainDrop can effectively reduce the sensitivity of the model to domain shifts and learn more generalized features, making it a promising method for DG tasks.

References

[1] 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. 1

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018. 2

[3] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *ECCV*, 2022. 2

[4] Yu Ding, Lei Wang, Bin Liang, Shuming Liang, Yang Wang, and Fang Chen. Domain generalization by learning and removing domain-specific features. In *NeurIPS*, 2022. 1, 3

[5] Cian Eastwood, Alexander Robey, Shashank Singh, Julius von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. Probable domain generalization via quantile risk minimization. In *NeurIPS*, 2022. 2

[6] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2020. 2

[7] Zeyi Huang, Haohan Wang, Dong Huang, Yong Jae Lee, and Eric P Xing. The two dimensions of worst-case training and their integrated effect for out-of-domain generalization. In *CVPR*, 2022. 2

[8] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. 2, 3

[9] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *ICCV*, 2021. 2

[10] Kyungmoon Lee, Sungyeon Kim, and Suha Kwak. Cross-domain ensemble distillation for domain generalization. In *ECCV*, 2022. 2

[11] Sangrok Lee, Jongseong Bae, and Ha Young Kim. Decompose, adjust, compose: Effective normalization by playing with frequency for domain generalization. In *CVPR*, 2023. 2, 3

[12] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 2

[13] Seonwoo Min, Nokyung Park, Siwon Kim, Seunghyun Park, and Jinkyu Kim. Grounding visual representations with texts for domain generalization. In *ECCV*, 2022. 2

[14] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021. 2

[15] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2

[16] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020. 1

[17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 4

[18] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. Informative dropout for robust representation learning: A shape-bias perspective. In *ICML*, 2020. 3

[19] Yuge Shi, Jeffrey Seely, Philip Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *ICLR*, 2021. 2

[20] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2

[21] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 2

[22] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 3

[23] Yue Wang, Lei Qi, Yinghuan Shi, and Yang Gao. Feature-based style randomization for domain generalization. *TCSVT*, 2022. 3, 4

[24] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *CVPR*, 2021. 3

[25] Xingchen Zhao, Chang Liu, Anthony Sicilia, Seong Jae Hwang, and Yun Fu. Test-time fourier style calibration for domain generalization. In *IJCAI*, 2022. 1

[26] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *TPAMI*, 2022. 1

[27] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020. 2

[28] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 3