Cross Contrasting Feature Perturbation for Domain Generalization

Chenming Li[1] Daoan Zhang[1] Wenjian Huang[1] Jianguo Zhang[1,2,*]
[1]Research Institute of Trustworthy Autonomous Systems and Department of Computer Science
and Engineering, Southern University of Science and Technology, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
12132339@mail.sustech.edu.cn, 12032503@mail.sustech.edu.cn, {huangwj, zhangjg}@sustech.edu.cn,

Abstract

Domain generalization (DG) aims to learn a robust model from source domains that generalize well on unseen target domains. Recent studies focus on generating novel domain samples or features to diversify distributions complementary to source domains. Yet, these approaches can hardly deal with the restriction that the samples synthesized from various domains can cause semantic distortion. In this paper, we propose an online one-stage Cross Contrasting Feature Perturbation (CCFP) framework to simulate domain shift by generating perturbed features in the latent space while regularizing the model prediction against domain shift. Different from the previous fixed synthesizing strategy, we design modules with learnable feature perturbations and semantic consistency constraints. In contrast to prior work, our method does not use any generative-based models or domain labels. We conduct extensive experiments on a standard DomainBed benchmark with a strict evaluation protocol for a fair comparison. Comprehensive experiments show that our method outperforms the previous state-of-the-art, and quantitative analyses illustrate that our approach can alleviate the domain shift problem in out-of-distribution (OOD) scenarios. *https://github.com/hackmebroo/CCFP*

1. Introduction

Deep Neural Networks have achieved remarkable success on a number of computer vision tasks[27, 65]. These models rely on the $i.i.d$ assumption[52], $i.e.$, the training data and testing data are identically and independently distributed. However, in real-world scenarios, the assumption does not always hold due to the $domain\ shift$ problem[5]. For instance, it is hard for a model trained on photographs to adapt to sketches.

Domain adaptation (DA) methods[13, 50, 61] can be employed to handle the out-of-distribution (OOD) issue in the settings where unlabeled target data is available. Although DA can perform well on known target domains, it still fails in practical situations where target domains are not accessible during training. Domain generalization (DG) [59] aims to deal with such problems. The goal of domain generalization is to learn a generalized model from multiple different but related source domains ($i.e.$ diverse training datasets with the same label space) that can perform well on arbitrary unseen target domains. To realize this goal, most deep learning models are trained to minimize the average loss over the training set, which is known as the Empirical Risk Minimization (ERM) principle[53]. However, ERM-based network provably fails to OOD scenarios[38, 11, 21, 64].

One line of work[49, 45, 46] improves the generalization capability of a model by optimizing the worst-domain risk over the set of possible domains, which are created by perturbing samples in the image level or using generative-based model ($i.e.$ VAE[25] or GAN[17]) to generate fictitious samples. Despite the performance promoted by creating samples in the image level on an offline basis to approximate the $worst\ case$ over the entire family of domains, it is hard to generate "fictitious" samples in the input space without losing semantic discriminative information[43]. Moreover, the offline two-stage data perturbation training procedure is nontrivial since both training a generative-based model and inferring them to obtain perturbed samples are challenging tasks.

Another line of work perturbs features in the latent space[55, 70] by tuning the scaling and shifting parameters after instance normalization. Another study[32] extends it and leverages the uncertainty associated with feature statistics perturbation. However, these methods all rely on a fixed perturbation strategy (linear interpolation or random perturbation) which limits the domain transportation from synthesized features to original features. Besides, although the instance normalization-based feature perturbation can change the information of intermediate features which is specific to domains, they still fail to preserve the semantic invari-

[*] Corresponding author.

ant, as the instance normalization may dilute discriminative information that is relevant to task objectives[39]. The performance of feature synthesis methods can be undetermined on account of semantic inconsistency[35].

As is mentioned above, the data perturbation based methods can hardly generate the fictitious samples in the input space, and the feature perturbation based methods limits the diversity of the synthesized features and fail to preserve the semantic consistency. To address both of these issues, we propose to enforce a domain-aware adaptive feature perturbation in the latent space following the worst-case optimization objective and explicitly constrain the semantic consistency to preserve the class discriminative information.

Practically, the desideratum for the worst-case DG problem is to simulate the realistic domain shift by maximizing the domain discrepancy and minimizing the class discriminative characteristics between the source domain distribution and the fictitious target domain distribution. To this end, we design an adaptive online one-stage $Cross\ Contrasting\ Feature\ Perturbation$ (CCFP) framework. An illustration of CCFP is shown in Figure 1. Our CCFP consists of two sub-network, one is used to extract the original features which represent the online estimate of the source distribution, and the other is used to perturb features in the latent space to create semantic invariant fictitious target distribution. In order to preserve the class discriminative information of the perturbed features, we regularize the predictions between the two sub-networks.

A key component of our framework is the feature perturbation. As pointed out in the research field of style transfer[10, 22], the feature statistics carry the information primarily referring to domain-specific but are less relevant to class discriminative. Based on this, we design a $learnable\ domain\ perturbation$ (LDP) module which can generate learnable perturbation of features to enlarge the domain transportation from the original ones. Note that the LDP only adds learnable scaling and shifting parameters on feature statistics without adopting domain labels or additional generative models.

Another critical point of CCFP is the measurement of domain discrepancy. Different from existing study measure the domain discrepancy in the last layer[58, 51], we propose to measure the domain discrepancy from the intermediate features to align with the observation that the shallow layers of the network learn low-level features (such as color and edges) which are more domain aware but less semantic relevant[60]. Additionally, Gatys et al.[15] show that Gram matrices of latent features can be used to encode stylistic attributes like textures and patterns. Motivated by this, we develop a novel Gram-matrices-based metric to represent the domain-specific information from the intermediate activations. We maximize the dissimilarity between the intermediate features' Gram matrices to simulate the domain shift.

We validate the effectiveness of CCFP on a standard DG benchmark called Domainbed[18]. Comprehensive experiment results show that our method surpasses previous methods and achieves state-of-the-art.

In summary, our contributions are three-fold:

- We propose a novel online one-stage cross contrasting feature perturbation framework (CCFP) for worst-case domain generalization problem, which can generate perturbed features while regularizing semantic consistency.
- We develop a learnable domain perturbation (LDP) module and an effective domain-aware Gram-matrices-based metric to measure domain discrepancy, which are useful for DG and integrated into the above CCFP framework. Additionally, our algorithm does not use any generative-based models and domain labels.
- Comprehensive experiments show that our method achieves state-of-the-art performance on diverse DG benchmarks under strict evaluation protocols of DomainBed[18].

2. Related Work

Domain generalization (DG) aims to learn generalized representations from multiple source domains that can generalize well on arbitrary unseen target domains. For example, in the PACS dataset[30], the task is to extract category-related knowledge, and the domains correspond to different artistic styles like art-painting, cartoon, photo, and sketch. The model will use three of four datasets to train and use the rest dataset to test. Various methods have been proposed in the DG literature that can be roughly classified into three lines: learning the domain invariant representation[51, 16, 37, 36, 57], meta-learning techniques[47, 3, 9, 29] and data perturbation based methods[56, 43, 71]. Our work is most relevant to the last line.

Data perturbation: Data perturbation in the input space can create diverse images to alleviate the spurious correlations[46] and improve the model generalization. Volpi et al.[56] proposed an adversarial data augmentation and learned an ensemble model for stable training. Bai et al.[2] decomposed feature representation and semantic augmentation approach for OoD generalization. Qiao et al.[43] extended it to create "fictitious" populations with large domain transportation. Zhou et al.[71] employed a data generator to synthesize data from pseudo-novel domains to augment the source domains. Different from these methods, we propose a latent space feature perturbation instead of perturbing raw data in the input space and require no domain labels or any generative-based models.

Feature perturbation: Unlike most data perturbation methods that adopt transformations in the input space, some approaches perturb features in the latent space. Li et al.[31] show that even perturbing the feature embedding with Gaussian noise during training leads to a comparable performance. Manifold Mixup [55] adopts linear interpolation from image level to feature level. Recent works show that linear interpolation on feature statistics of two instances [70] can synthesize samples to improve model generalization. Nuriel et al.[41] randomly swaps statistics of different samples from the same batch. [32] extends it and leverages the uncertainty associated with feature perturbations. These methods are based on a fixed perturbation strategy and lack a constraint to preserve semantics. In our work, the LDP modules can generate learnable perturbations to enlarge the domain transportation while the CCFP framework can explicitly preserve the semantic consistency.

3. Method

3.1. General Formulation

We formulate domain generalization in a classification setting from the input features $x \in \mathcal{X}$ to the predicting labels $y \in \mathcal{Y}$. Given a model family Θ and training data drawn from some distribution. The goal is to find a model $\theta \in \Theta$ that generalizes well to unseen target distribution P_{tar}. DG can be formulated as the following problem:

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim P_{tar}}[\ell(\theta;(x,y))] \tag{1}$$

where $\mathbb{E}[\cdot]$ is the expectation, $\ell(\cdot,\cdot)$ is the loss function.

The challenge for DG is that the target domain distribution P_{tar} is not available. An alternative approach to solve Eq.(1) is to merge all the data from source domains and learn the model by minimizing the training error across the pooled data. This is known as the Empirical Risk Minimization (ERM) principle:

$$\hat{\theta}_{ERM} := \min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim P_{src}}[\ell(\theta;(x,y))], \tag{2}$$

where P_{src} is the empirical distribution over the training data. Since the ERM-based methods provably lack robustness on OOD scenarios[38, 11], a number of work[28, 20, 45, 46] formulated DG as a worst-case problem leveraging distributionally robust optimization and adversarial training:

$$\hat{\theta}_{worst-case} := \min_{\theta \in \Theta} \sup_{P:D(P,P_{src})\leq\rho} \mathbb{E}_P[\ell(\theta;(x,y))] \tag{3}$$

Here $D(\cdot,\cdot)$ is a distance metric on the space of probability distributions. The solution to Eq.(3) aims to achieve a good performance against the domain shifts while the fictitious target distributions P are distance ρ away from the source domain distribution P_{src}. To solve Eq.(3), previous work expects to create fictitious distributions P by perturbing training samples in the input space or using generative-based models and updating the model with respect to these fictitious worst-case target distributions.

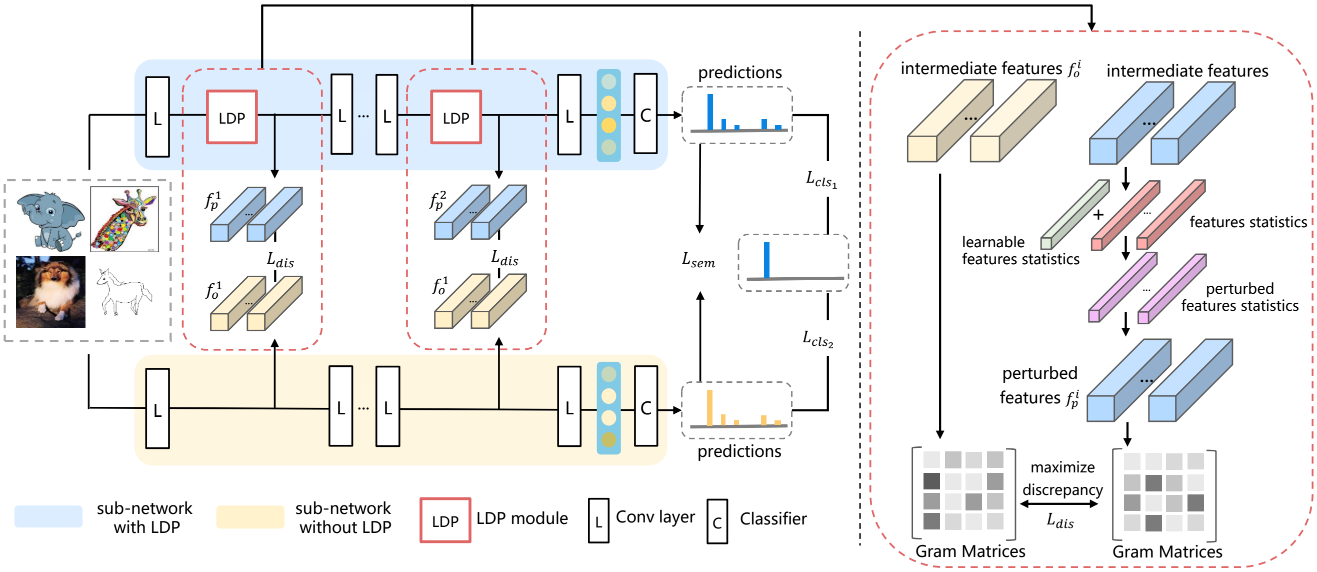
However, perturbing samples in the image level may introduce class distortions detrimental to model training which may cause the performance decline. Moreover, the offline two-stage training procedure requires significant computational resources since training the generative model and using it to obtain additional samples are both challenging tasks[60, 34].

In this regard, we propose an online one-stage *cross contrasting feature perturbation* (CCFP) framework (Sec.3.2) to obtain perturbed representation distribution P^l with learnable feature statistics (Sec.3.3) in latent space without using any generative-based model. Further, to preserve the semantic discriminative information of the perturbed features, we utilize an explicit semantic constraint to encourage the model to predict consistent semantic representations. As it is demanded to determine the source domain distribution and the fictitious target domain distribution according to Eq.3, we utilize a dual stream network as it is illustrated in Figure 1.

It is noteworthy that the distance metric is essential to the worst-case DG problem since it is used to measure the dissimilarity between the source domain distribution and the fictitious target domain distribution. The ideal goal of the metric is to create a fictitious target distribution with a large domain discrepancy from the source distribution as well as retain semantic discriminative information. Previous work directly boosts the dissimilarity in the high-level semantic space[56] (usually the output of the last layer), thus failing to preserve the semantic discriminative information. To satisfy the goal, we propose a domain-aware Gram-matrices-based metric to boost the dissimilarity in the whole latent space except for the high-level semantic space (Sec.3.4). Further, to better preserve the semantic discriminative information, we utilize a regularization loss to explicitly constrain the semantic consistency between the source domain and the fictitious target domain in Sec.3.5.

3.2. Cross Contrasting Feature Perturbation Framework (CCFP)

Since our goal is to simulate the realistic domain shifts in the latent space. To determine the source domain distribution and create the fictitious target domain distribution, we employ two sub-networks to extract features from the same images. As illustrated in Figure 1, one is used to extract the original features which represent the online estimation of source distribution in latent space. The other is used to gen-



(a) Overall architecture

(b) Learnable domain perturbation

Figure 1. **An overview of our proposed CCFP.** Our framework consists of two sub-networks. (a) The bottom network is a pre-trained backbone, and the top network is the same pre-trained backbone equipped with LDP modules (red boxes). The two sub-networks have similar architecture (except for the LDP modules) but do not share parameters. The steps of feature perturbation and the calculation of $\mathcal{L}_{dis}$ are shown in (b).

erate perturbed features which represent the fictitious target distribution by using our learnable domain perturbation modules in the intermediate layers.

From the practical perspective, it is intractable to select an appropriate magnitude of the domain shift ρ. In this regard, we consider Eq.3 as the following Lagrangian relaxation with penalty parameter γ:

$$\hat{\theta} := \min_{\theta \in \Theta} \sup_{P^l} \{\mathbb{E}_{P^l}[\ell(\theta;(x,y))] - \gamma D(P^l, P^l_{src})\} \quad (4)$$

Here P^l_{src} is the source domain distribution in the latent space, and the P^l is the fictitious target distribution in the latent space. Taking the dual reformulation Eq.4, we can obtain an *min-max* optimization objective that maximizes the domain discrepancy between the source distribution and the fictitious target distribution while minimizing the target risk.

During the *min-max* optimization, each iteration can be divided into two steps. For the maximization step, a batch of images will be fed into both sub-networks and are used to calculate the domain discrepancy loss, denoted by $\mathcal{L}_{dis}$ detailed in Eq.8, only the parameters of LDP blocks will be updated at this stage. For the minimization step, the same batch of images will be fed into the model again and are used to calculate the classification loss (cross-entropy loss) and semantic consistency loss, denoted by $\mathcal{L}_{sem}$ detailed in Eq.9, all the parameters of two sub-networks will be updated at this stage.

3.3. Learnable Domain Perturbation Module (LDP)

The key point of our CCFP framework is how to create domain-aware feature perturbation. As perturbing parameters for an affine transformation of intermediate features after normalization can change their characteristics which primarily refer to domain-specific information but are less relevant to category-related information[10, 22], Huang et al.[22] propose the adaptive instance normalization (AdaIN), which replaces the feature statistics of the input features x with the feature statistics of a style image's features x_s to achieve style transfer. Let $x \in \mathbb{R}^{B\times C\times H\times W}$ be a batch of features, the AdaIN can be formulated as:

$$AdaIN(x) = \sigma(x_s)\frac{x-\mu(x)}{\sigma(x)} + \mu(x_s) \quad (5)$$

where $\mu(x) \in \mathbb{R}^{B\times C}$ and $\sigma(x) \in \mathbb{R}^{B\times C}$ are the mean and standard deviation respectively. However, in DG scenarios, the feature statistics of target domain images are not available. Previous work[70, 32] utilizes linear interpolation or uncertainty modeling to diversify the feature statistics, but both of them limit the domain transportation from synthesized features to original features. To address this, we design a learnable domain perturbation (LDP) module (The red box in Figure 1) to generate perturbed intermediate features:

$$LDP(x) = (\sigma(x) + \gamma)\frac{x-\mu(x)}{\sigma(x)} + \mu(x) + \beta \quad (6)$$

Here we only add learnable parameters γ and β to the features' original scaling $\sigma(x)$ and shifting $\mu(x)$ statistics. Different from prior works based on a fixed perturbation strategy, the LDP module can enlarge the domain discrepancy between the original and the perturbed features.

3.4. Gram-based Domain Discrepancy Metric

The worst-case optimization objective for DG is to guarantee model performance against fictitious target distribution within a certain distance from the source distribution. Considering the essential desideratum of DG that enables the model to generalize well to the unseen domain, the ideal distance metric is domain-specific and class-discriminative agnostic. Inspired by the well-known observation[66] that the shallow layers learn low-level features which are task-irrelevant, we build an effective domain discrepancy metric applied to the shallow layers. Specifically, Gatys et al.[15] shows that Gram matrices can encode stylistic attributes like textures and patterns that are less relevant to task objectives but can be used to depict the individual domain information. Therefore, we utilize the Gram-matrices-based metric to depict the domain discrepancy.

Specifically, we denote the network as the following:

$$c(x) = g \circ f^n \circ f^{n-1} \circ \cdots \circ f^1(x) \tag{7}$$

Here g is the classifier, $f = f^n \circ f^{n-1} \circ \cdots \circ f^1(x)$ is the feature extractor, and n denotes the number of shallow layers. In our method, we use a set of Gram matrices $\{G^1, G^2, \cdots, G^K\}$ from a set of shallow layers $\{f^1, \cdots, f^K\}$ in the network to describe the domain-specific characteristics. The domain discrepancy loss can be formulated as:

$$\mathcal{L}_{dis} = -\sum_{i=1}^{K} ||G(f_o^i(\mathbf{x})) - G(f_p^i(\mathbf{x}))||_F \tag{8}$$

where f_o and f_p are two feature extractors (original and perturbed) respectively. K is the number of shallow layers to calculate the loss $\mathcal{L}_{dis}$, $G(\cdot)$ is the Gram matrix, and the $||\cdot||_F$ denotes the Frobenius norm.

3.5. Explicit Semantic Consistency Constraint

To better preserve the semantic consistency, we minimize the L2-norm between the final classifier predictions of two sub-networks. The semantic consistency loss can be formulated as:

$$L_{sem} = ||g_o(f_o(\mathbf{x})) - g_p(f_p(\mathbf{x}))||_2^2 \tag{9}$$

Here g_o and g_p are two classifiers (original and perturbed respectively). The final loss is given by:

$$\mathcal{L}_{final} = \mathcal{L}_{cls_1} + \mathcal{L}_{cls_2} + \lambda_{dis}\mathcal{L}_{dis} + \lambda_{sem}\mathcal{L}_{sem} \tag{10}$$

Algorithm 1 : Cross Contrasting Feature Perturbation

```
Input:$\mathcal{S}_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, batch size $B$, learning rate $\eta$, Adam optimizer, initial $\lambda_{dis}, \lambda_{sem}$
Initial: Parameters of CCFP $i.e.$ parameters $\theta_0$, $\theta_1$, $\phi_0$, $\phi_1$, $(\gamma_k, \beta_k; k = 1 \cdots K)$ for feature extractor $f_o$, $f_p$, classifier $g_o, g_p$ and LDP modules $P^1, P^2 \cdots P^K$ (K is defined in Eq.8).
repeat
  Minimization Stage:
  for $i = 1, \cdots, B$ do
    $\mathcal{L}_{cls_1}^i = \ell(g_o(f_o(\mathbf{x}_i)), y_i)$
    $\mathcal{L}_{cls_2}^i = \ell(g_p(f_p(\mathbf{x}_i)), y_i)$
    $\mathcal{L}_{sem}^i = \lambda_{sem}||f_o(\mathbf{x}_i) - f_p(\mathbf{x}_i)||_2^2$
  end for
  $\theta_0, \phi_0 \leftarrow \text{Adam}(\frac{1}{B}\sum_{i=1}^B \mathcal{L}_{cls_1}^i + \mathcal{L}_{sem}^i, \theta_0, \phi_0, \eta)$
  $\theta_1, \phi_1, \gamma_k, \beta_k \leftarrow \text{Adam}(\frac{1}{B}\sum_{i=1}^B \mathcal{L}_{cls_2}^i + \mathcal{L}_{sem}^i, \theta_1, \phi_1, \gamma_k, \beta_k, \eta)$
  Maximization Stage:
  for $i = 1, \cdots, B$ do
    $\mathcal{L}_{dis}^i = \lambda_{dis}\sum_{k=1}^K ||G(f_o^k(\mathbf{x}_i)) - G(f_p^k(P^k(\mathbf{x}_i)))||_F$
  end for
  $\gamma_k, \beta_k \leftarrow \text{Adam}(\frac{1}{B}\sum_{i=1}^B \mathcal{L}_{spe}^j, \gamma_k, \beta_k, \eta)$
until $\theta_0$, $\theta_1$, $\phi_0$, $\phi_1$ are converged
```

The λ_{dis} and the λ_{sem} are used to control the strength of the domain discrepancy loss $\mathcal{L}_{dis}$ and the semantic consistency loss $\mathcal{L}_{sem}$.

The optimization algorithm is designed in Algorithm 3.5. During the inference, we only use the sub-network (the top network in the Figure 1) which is learned from the perturbed features to predict the final results, as the diverse latent features help mitigate the domain shift. Additionally, the LDP modules will also be used to prevent variations in the normalization statistics that could otherwise cause model collapse. Although the statistics shift can be alleviated by randomly applying LDP during training, the LDP modules can also be used as a test-time augmentation technology to boost the performance, we will discuss it in the Appendix.

4. Experiments

4.1. DomainBed Benchmark

We conduct comprehensive experiments on the DomainBed benchmark[18]. DomainBed includes seven multi-domain image classification tasks: Colored MNIST[1], Rotated MNIST[16], PACS[30], VLCS[12], Office-Home[54], Terra Incognita[4], and DomainNet[42].

Colored MNIST[1] is a variant of MNIST consisting of 70,000 examples of dimension (2, 28, 28) and 2 classes. The dataset contains a disjoint set of colored dig-

Algorithm	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg
ERM[52]	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[1]	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[46]	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[62]	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[29]	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[50]	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[33]	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[14]	51.5 ± 0.2	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[33]	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[6]	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[40]	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[67]	**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
V-REx[26]	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[23]	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
AND-mask[24]	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
SAND-mask[24]	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
Fish[48]	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[44]	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
CCFP (ours)	51.9 ± 0.1	97.8 ± 0.1	**78.9** ± 0.3	**86.6** ± 0.2	**68.9** ± 0.1	**48.6** ± 0.4	41.2 ± 0.0	**67.7**

Table 1. DomainBed with Training-domain model selection. We highlighted the best results using **bold** font.

its where domain $d \in \{90\%, 80\%, 10\%\}$ is the correlation strength between color and label across domains. **Rotated MNIST**[16] is a variant of MNIST consisting of 70,000 examples of dimension (1, 28, 28) and 10 classes. The dataset contains digits rotated by d degrees where domain $d \in \{0, 15, 30, 45, 60, 75\}$. **PACS**[30] includes domains $d \in \{$art, cartoons, photos, sketches$\}$ with 9,991 examples of dimension (3, 224, 224) and 7 classes. **VLCS**[12] includes domains $d \in\{$Caltech101, LabelMe, SUN09, VOC2007$\}$ with 10,729 examples of dimension (3, 224, 224) and 5 classes. **Office-Home**[54] includes domains $d \in \{$atr, clipart, product, real$\}$ with 15,588 examples of dimension (3, 224, 224) and 65 classes. **Terra Incognita**[4] contains photographs of wild animals taken by camera traps at locations $d \in \{$L100, L38, L43, L46$\}$ with 24,788 examples of dimension (3, 224, 224) and 10 classes. **DomainNet**[42] includes domains $d \in \{$clipart, infograph, painting, quickdraw, real, sketch$\}$ with 586,575 examples of dimension (3, 224, 224) and 345 classes.

For a fair comparison, the DomainBed benchmark[18] presents an evaluation protocol about dataset splits, model selection on the validation set, and hyperparameter (HP) search, which is detailed below.

Dataset splits. The data from source domains are split into training subsets (80%) and validation subsets (20%) (used on Training-domain validation set model selection). The data from the target domain are split into testing subsets (80%) and validation subsets (20%) (used on Test-domain validation set model selection). We repeat the entire experiment three times using different seeds and report the mean and standard error over all the repetitions.

Model selection methods. There are three model selection methods in [18]. (i) Training-domain validation set. (ii) Leave-one-out cross-validation. (iii) Test-domain validation set (oracle). We choose the Training-domain model selection that assumes the training and test examples follow similar distributions. The best-performing model in the validation set is selected as the final model, and its test domain performance is reported as the final performance. The results of the oracle model selection are shown in Appendix.

Model architectures. Following DomainBed, we use Conv-Net (detail in Appendix D.1 in [18]) as the backbone for Colored MNIST and Rotated MNIST and use ResNet-50[19] for the rest datasets. For the classifier, we only use one linear layer. We insert the LDP modules after the first Conv, Max Pooling, and 1,2,3-th ConvBlock, and we further perform an ablation study for the effects of different inserted positions. When using Conv-Net as our backbone, we insert the LDP modules at the position after the first three Batch Normalization layers.

Hyperparameter (HP) search. We run a random search of 20 trials over the hyperparameter distribution given by DomainBed. Our CCFP relies on two additional hyperparameters λ_{spe} and λ_{sem}, and we set the range of search such as $[0.1, 10]$ for both of them, more details about the range of hyperparameter search will be discussed in Appendix.

Implementation details. We implement our algorithm using the codebase of DomainBed in PyTorch, using

ResNet-50 pre-trained on the ImageNet[8] and fine-tuning on each dataset. Note that our evaluation setting follows the standard evaluation protocol given by DomainBed[18].

4.2. Results

Comparison with domain generalization methods on Domainbed benchmark. Comprehensive experiments show that CCFP achieves significant performance gain against previous methods on most of the benchmark datasets and obtains comparable performance on three of seven datasets. Table 1 summarizes the results on DomainBed using the Training-domain model selection method. Our CCFP outperforms all previous approaches on the averaged result.

To further validate the generalization of CCFP, we conduct experiments under another commonly used baseline SWAD[7] as our backbone, which is a unique model selection mechanism. For a fair comparison, we only summarize the methods based on SWAD. The performance comparison with other existing approaches that adopted SWAD is provided in Tables 2-4. Our CCFP achieves significant performance gain in all experiments against previous best results.

Algorithm	A	C	P	R	Avg.
SWAD[7]	66.1	57.7	78.4	80.2	70.6
PCL[63]	67.3	**59.9**	78.7	80.7	71.6
CCFP (ours)	**68.0**	58.6	**79.7**	**81.9**	**72.1**

Table 2. Comparison with SWAD-based state-of-the-art methods on OfficeHome benchmark. A: art, C: clipart, P: product, R: real, Avg.: average.

Algorithm	C	L	S	V	Avg.
SWAD[7]	98.8	63.3	**75.3**	79.2	79.1
PCL[63]	**99.0**	63.6	73.8	75.6	78.0
CCFP (ours)	98.9	**64.1**	74.9	**79.9**	**79.4**

Table 3. Comparison with SWAD-based state-of-the-art methods on VLCS benchmark. C: Caltech101, L: LabelMe, S: SUN09, V: VOC2007, Avg.: average.

Algorithm	L100	L38	L43	L46	Avg.
SWAD[7]	55.4	44.9	59.7	39.9	50.0
PCL[63]	58.7	46.3	60.0	43.6	52.1
CCFP (ours)	**59.9**	**47.6**	**60.8**	**43.8**	**53.0**

Table 4. Comparison with SWAD-based state-of-the-art methods on TerraIncognita benchmark. L100: Location 100, L38: Location 38, L43: Location 43, L46: Location 46, Avg.: average.

Comparison with previous feature perturbation methods. To reveal the performance gain by using learnable parameters to perturb feature statistics, we conduct experiments to compare with two previous features perturbation methods Mixstyle[70] and DSU[32]. Since both of them use their own experiment settings. For a fair comparison, we rerun their results on the DomainBed experiment benchmark. Table 5 shows that our CCFP achieves a substantial improvement in performance compared to the previous feature perturbation methods more experiment results are shown in Appendix.

Algorithm	A	C	P	S	Avg.
ERM	81.6	78.7	95.5	78.7	83.6
Mixstyle[70]	84.0	79.9	94.3	**81.6**	84.9
DSU[32]	81.9	79.6	95.0	79.6	84.1
CCFP (ours)	**87.5**	**81.3**	**96.4**	81.4	**86.6**

Table 5. Comparison with previous feature perturbation methods on PACS benchmark. Comparison with SWAD-based state-of-the-art methods on PACS benchmark.

5. Ablation Study

Effects of the explicit semantic regularization. To validate the effectiveness of the semantic regularization, we conduct experiments without using the semantic consistency loss in Eq.10. Table 6 shows that semantic regularization can achieve performance gain on most target domains and the average accuracy. In particular, we can find that without semantic regularization, our method can still significantly outperform ERM.

Algorithm	A	C	P	S	Avg.
ERM	81.6	78.7	95.5	78.7	83.6
CCFP(w/o) L_{sem}	83.6	**83.9**	**96.4**	80.3	86.0
CCFP (ours)	**87.5**	81.3	**96.4**	**81.4**	**86.6**

Table 6. Comparison with result without using L_{sem} on PACS benchmark.

To further validate the essential to regularize the semantic consistency after perturbing features in the latent space, we enforce the semantic regularization on previous feature perturbation methods. Note that both Mixstyle and DSU use one single network to generate the perturbed features, which is unable to calculate the L_{sem} in Eq.10. To address this, we implement the two methods in our CCFP framework. Similar to our approach, we use one sub-network to extract the original features and use the other sub-network to generate the perturbed features by using Mixstyle and DSU feature perturbation methods. Further, we constrain the consistency between the predictions of the two sub-networks. During the inference, we only use the perturbed sub-network to produce the final predictions which are the same as our approach. Since Mixstyle and DSU are non-parametric, we

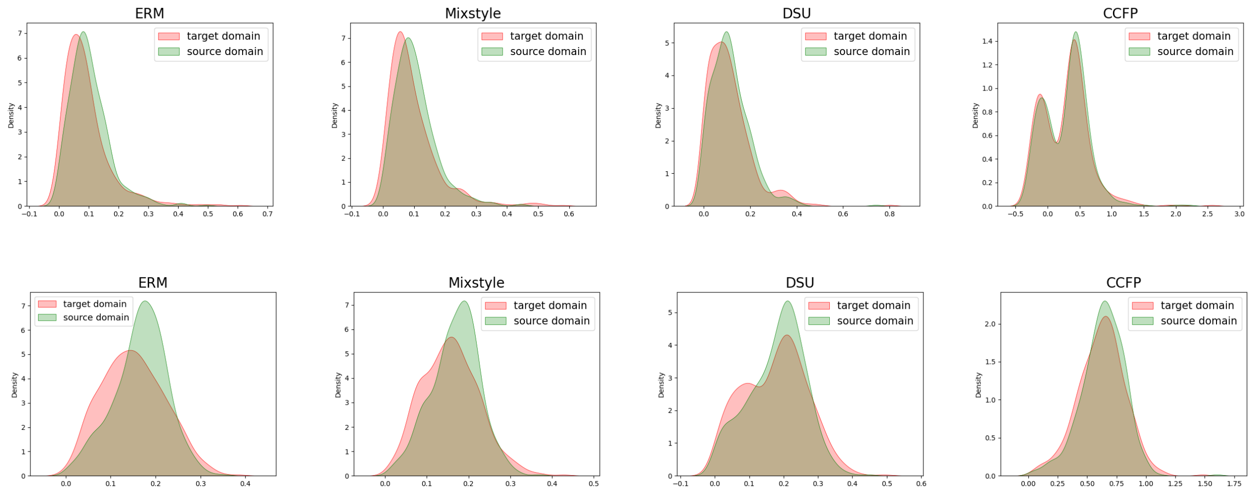


Figure 2. The visualization of feature statistics at the position 3. The top raw is the mean statistics and the bottom raw is the std statistics. We conduct the experiments on the PACS dataset with ERM, Mixstyle, DSU and our CCFP.

remove the L_{dis} in Eq.10 in this experiment. Table 7 shows that our dual stream architecture and the explicit semantic consistency regularization can achieve a significant performance gain (0.3% for Mixstyle and 1.4% for DSU).

Algorithm	**A**	**C**	**P**	**S**	**Avg.**
Mixstyle[70]	84.0	79.9	94.3	81.6	84.9
Mixstyle (dual)	84.6	80.3	**96.5**	79.5	85.2
DSU[32]	81.9	79.6	95.0	79.6	84.1
DSU (dual)	86.3	79.4	94.6	**81.7**	85.5
CCFP (ours)	**87.5**	**81.3**	96.4	81.4	**86.6**

Table 7. Validation of the additional semantic consistency for previous feature perturbation methods on PACS benchmark.

Effects of LDP inserted positions. In CCFP, we use a set of Gram matrices of intermediate features from a set of layers $\{f^1, \cdots, f^K\}$ to describe the domain-specific characteristics[69, 68]. To verify the effects of LDP inserted positions, we name the position of ResNet after the first Conv, Max Pooling, and 1,2,3-th ConvBlock as 1,2,3,4,5 respectively, and the effects of LDP on different inserted positions are evaluated accordingly. We conduct the experiments on dataset PACS and OfficeHome with the default hyperparameters given by DomainBed. Table 8 shows that more inserted LDP modules can produce relatively higher classification accuracy. Hence we plug the LDP modules into all five positions for our main experiments.

Visualization analysis on CCFP. To confirm that our CCFP can alleviate the domain shift phenomena, we conduct experiments on the PACS dataset where we choose art painting as the target domain and the rest as the source domain. We capture the intermediate features at position 4 to study the feature statistic shifts. Figure 2 shows the feature statistics distribution from source domains and target domains based on ERM, Mixstyle, DSU, and our CCFP. For a fair comparison, we reproduce the results of ERM, Mixstyle, DSU, and CCFP with the same fixed steps (5,000 steps, which is the same as the default value given by DomainBed) and only consider the final checkpoint. It is shown that our CCFP can obviously mitigate the domain shift between the source and target domain features compared with ERM and surpass Mixstyle and DSU. The result shows that our method can help against the domain shift.

Positions	**1-3**	**2-4**	**3-5**	**1-5**	**ERM**
PACS	85.3	84.8	85.4	**86.6**	83.6
OfficeHome	68.4	68.5	68.3	**68.9**	64.5

Table 8. Effects of different inserted positions on PACS and OfficeHome benchmark.

6. Conclusions

In this paper, we propose a simple yet efficient cross contrasting feature perturbation framework. Unlike previous works, our method does not use generative-based models or domain labels. Our approach can adaptively generate perturbed features with large domain transportation from the original features while preserving semantic consistency, and encourage the model to predict consistent semantic representation against the domain shift. The experiments show that our method performs better than the previous state-of-the-art on the DomainBed benchmark.

7. Acknowledgments

This work is supported by National Key Research and Development Program of China (2021YFF1200800)

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *stat*, 1050:27, 2020.

[2] Haoyue Bai, Rui Sun, Lanqing Hong, Fengwei Zhou, Nanyang Ye, Han-Jia Ye, S-H Gary Chan, and Zhenguo Li. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6705–6713, 2021.

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

[4] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[5] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[6] Gilles Blanchard, Aniket Anand Deshmukh, Ürun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1):46–100, 2021.

[7] 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.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] 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.

[10] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[11] Cian Eastwood, Alexander Robey, Shashank Singh, Julius von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. Probable domain generalization via quantile risk minimization. *arXiv preprint arXiv:2207.09944*, 2022.

[12] 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*, pages 1657–1664, 2013.

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[14] 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.

[15] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

[16] 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*, pages 2551–2559, 2015.

[17] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *stat*, 1050:10, 2014.

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

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348, 2021.

[21] Wenjian Huang, Hao Wang, Jiahao Xia, Chengyan Wang, and Jianguo Zhang. Density-driven regularization for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 35:887–900, 2022.

[22] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

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

[24] 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*, pages 9619–9628, 2021.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014.

[26] 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*, pages 5815–5826. PMLR, 2021.

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[28] Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. *Advances in Neural Information Processing Systems*, 31, 2018.

[29] 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, 2018.

[30] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generaliza-

tion. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[31] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021.

[32] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and LINGYU DUAN. Uncertainty modeling for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

[33] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[34] Qiujing Lu, Yipeng Zhang, Mingjian Lu, and Vwani Roychowdhury. Action-conditioned on-demand motion generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2249–2257, 2022.

[35] Wang Lu, Jindong Wang, Yiqiang Chen, Sinno Jialin Pan, Chunyu Hu, and Xin Qin. Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–19, 2022.

[36] 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*, pages 5715–5725, 2017.

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

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

[39] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[40] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021.

[41] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: Reducing the bias towards global statistics in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9482–9491, 2021.

[42] 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*, pages 1406–1415, 2019.

[43] 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*, pages 12556–12565, 2020.

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

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

[46] 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.

[47] Swami Sankaranarayanan and Yogesh Balaji. Meta learning for domain generalization. In *Meta-Learning with Medical Imaging and Health Informatics Applications*, pages 75–86. Elsevier, 2023.

[48] Yuge Shi, Jeffrey Seely, Philip Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2021.

[49] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

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

[51] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[52] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[53] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[54] 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*, pages 5018–5027, 2017.

[55] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

[56] 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.

[57] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2018.

[58] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the*

26th ACM international conference on Multimedia, pages 402–410, 2018.

[59] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[60] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[61] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

[62] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *stat*, 1050:3, 2020.

[63] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7097–7107, 2022.

[64] Daoan Zhang, Mingkai Chen, Chenming Li, Lingyun Huang, and Jianguo Zhang. Aggregation of disentanglement: Reconsidering domain variations in domain generalization. *arXiv preprint arXiv:2302.02350*, 2023.

[65] Daoan Zhang, Chenming Li, Haoquan Li, Wenjian Huang, Lingyun Huang, and Jianguo Zhang. Rethinking alignment and uniformity in unsupervised image semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11192–11200, 2023.

[66] Linfeng Zhang, Xin Chen, Junbo Zhang, Runpei Dong, and Kaisheng Ma. Contrastive deep supervision. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 1–19. Springer, 2022.

[67] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group distribution shift. *arXiv preprint arXiv:2007.02931*, 2020.

[68] Yipeng Zhang, Hoyoung Chung, Jacquline P Ngo, Tonmoy Monsoor, Shaun A Hussain, Joyce H Matsumoto, Patricia D Walshaw, Aria Fallah, Myung Shin Sim, Eishi Asano, et al. Characterizing physiological high-frequency oscillations using deep learning. *Journal of neural engineering*, 19(6):066027, 2022.

[69] Yipeng Zhang, Qiujing Lu, Tonmoy Monsoor, Shaun A Hussain, Joe X Qiao, Noriko Salamon, Aria Fallah, Myung Shin Sim, Eishi Asano, Raman Sankar, et al. Refining epileptogenic high-frequency oscillations using deep learning: a reverse engineering approach. *Brain communications*, 4(1):fcab267, 2022.

[70] Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa. Domain generalization with optimal transport and metric learning. *arXiv preprint arXiv:2007.10573*, 2020.

[71] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European conference on computer vision*, pages 561–578. Springer, 2020.