ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240659

Rethinking Domain Generalization from Perspective of Gradient Granularity

Yujie Zhou[a], Haigen Hu[a,*], Qianwei Zhou[a], Qiu Guan[a] and Mingfeng Jiang[b]

[a]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China.
[b]School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China.

Abstract. Domain generalization (DG) aims to enhance the ability of model learning from source domains to generalize to other unseen domains. Existing gradient-based methods focus on learning better domain-invariant features using gradients from multiple source domains, but do not consider the impact of gradient granularity on model training. In this paper, we rethink how to mitigate the gradient conflicting problem from an optimization perspective. The limitations of existing gradient-based methods are theoretically analyzed in terms of modification ratio and modification frequency, showing that gradient granularity is a key factor in ensuring correct modification of the gradient. To address this issue, a gradient modification method, called CorGrad, is proposed by layering and slicing refinement operations to increase the modification frequency and the modification ratio. It can better reduce domain-specific information so that the model can learn better domain-invariant features. Finally, extensive experiments are conducted to verify the effectiveness of the proposed CorGrad, and the results show that the proposed CorGrad can obtain competitive performance in five DG benchmarks, and an average performance of 60.4% can be obtained on the DomainBed when using ResNet18 as the backbone. The code is publicly available at https://github.com/libzwo/CorGrad.

1 Introduction

Domain generalization (DG) aims to incorporate knowledge from multiple source domains and generalize it to the unseen target domain by overcoming out-of-distribution (OOD) in testing data. The assumption of independent and identically distributed (IID) is not always valid in practice due to the domain shift between the training and testing data. To minimize the influence of domain shift, a series of gradient-based methods have been proposed. The gradient is widely perceived as a rich representation of the task itself [1, 42], and the gradient conflict caused by the different domains existing in training sets is considered one of the biggest challenges for the DG task [25].

Recent work [20, 17, 42, 25, 35, 31] has made great progress in the DG task, showing that addressing conflicting gradients during multi-source domain training can mitigate inter-domain interference and typically enhance generalization capabilities for unseen domains. In our opinion, the conflict direction component of the gradient mainly contains domain-specific information. Therefore, elimination of the conflicting direction component is essential to reduce the model bias towards domain-specific features. The elimination process can help facilitate the learning of domain-independent features and ultimately enhances the ability to generalize effectively to unseen domains.

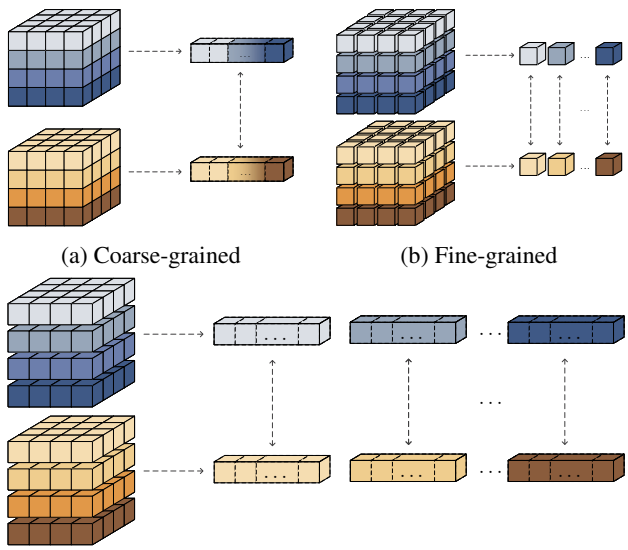


Figure 1: Schematic diagram of different gradient granularities. Here different colored blocks represent parameters from different layers, and block-to-block intervals represent model parameter divisions. Bidirectional arrows represent conflict computation between two gradients of the same magnitude corresponding to each other.

Existing methods fall mainly into two categories from the perspective of gradient granularity: coarse-grained gradient [35, 31] and fine-grained gradient [25]. The former involves selecting the gradients of all model parameters as conflicting gradients, essentially treating the gradients of all model parameters as a high-dimensional vector (*e.g.* , ResNet18 with 11,176,512 learnable parameters). Figure 1a illustrates the corresponding concept of the coarse-grained method. The large cube, representing the parameters of a model, is presented as a continuous entity without gaps, and it is considered as a high-dimensional vector that encompasses all the model parameters. In fact, we have shown mathematically that two high-dimensional vectors from random distributions tend to be orthogonal as dimensions go to infinity (see ection 3.2 for details). Following this principle, suppose that there is a smaller modification ratio (see Figure 2 for the definition) to the model in the coarse-grained method, the reason is that the component in the direction of conflict becomes smaller as vectors get closer to the orthogonality. It is obvious from Figure 2 that

* Corresponding Author. Email: hghu@zjut.edu.cn

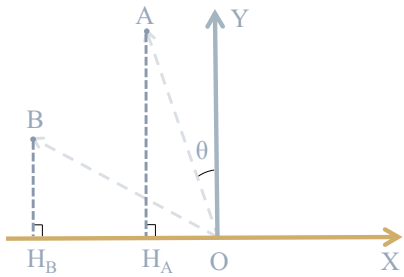


Figure 2: An illustration of gradient modification ratios with different angles, where the modification ratio of $\overrightarrow{OA}$ is defined as $|\overrightarrow{OH_A}|/|OA|$.
$\overrightarrow{OA}$ is closer to the orthogonal with respect to $\overrightarrow{OX}$ than $\overrightarrow{OB}$, leading to a smaller conflict direction component $\overrightarrow{OH_A}$ compared to $\overrightarrow{OH_B}$. Our conjecture is also verified by subsequent experimental results (see Figure 6 for details). On the contrary, the fine-grained method uses the gradient of a single learnable parameter of the model as the conflicting gradient, which is one-dimensional (as shown in Figure 1b). However, it is easy to lose contextual information with other dimensions using only a single component of a vector. Similarly to the thermal motion of frequent collisions between microscopic particles, we believe that fine-grained methods will result in frequent collisions, thereby leading to a lower modifications frequency (see Equation (3) for the definition). In fact, this hypothesis is further confirmed by the frequency of gradient modification updates in subsequent experiment results (see Figure 7 for details).

Based on the above analysis, we can summarize the following two points for the existing methods. (i) For coarse-grained methods, the higher the vector dimension, the easier it is to be orthogonal, thereby leading to smaller orthogonal conflict direction components and a lower modification ratio. (ii) For fine-grained methods, with lower vector dimensions, conflicts between vectors become more frequent, resulting in a lower modification frequency. Therefore, selecting the right gradient granularity is the key to optimizing gradient conflict strategies, while simultaneously increasing the modification ratio and the modification frequency remains a major challenge in the DG task. In addition, the dominant gradient contains a large amount of domain-specific information, and its presence may cause the model to be biased toward specific source domains. Existing methods usually ignore the dominant gradient, which will undoubtedly have a negative impact on the DG task. Smoothing the dominant gradients is the key to ensuring that no bias occurs, and it is another challenge for DG task.

To address the aforementioned challenges, we propose a gradient correction scheme from the perspective of gradient granularity to solve the problem of gradient conflict during the training process of multi-source domains in this work. The proposed scheme aims to increase both the gradient modification ratio and the gradient modification frequency, thus it is called **CorGrad**. Figure 1c illustrates our train of thought for the scheme. Specifically, we first analyze the limitations of existing gradient-based methods from the perspective of gradient granularity, showing that appropriate gradient granularity is crucial to optimize gradient conflict strategies. Then, a gradient granularity subdivision scheme is proposed to simultaneously increase the modification frequency and modification ratio in gradient-based methods. Finally, we propose an adaptive smoothing strategy to mitigate the adverse impacts of the dominant gradient on model training during the training process. Our contributions can be summarized as follows.

- We conducted an analysis of existing gradient-based methods from the perspective of gradient granularity.
- A gradient granularity subdivision scheme is proposed to increase both the modification frequency and the modification ratio in gradient-based methods.
- An adaptive smoothing strategy is proposed to address the adverse impacts of the dominant gradient during training.
- Extensive experiments indicate that the proposed gradient optimization strategy is competitive and promising during multi-source domain learning tasks.

2 Related work

2.1 Domain Generalization

Domain Generalization (DG) aims to improve the generalization ability of machine learning algorithms from observed source domains to unseen target domain. Currently, most of the existing DG methods can be primarily categorized into three types: Data-based, Model-based, and Optimization-based.

Data-based DG methods. From the *data* perspective, data augmentation methods, such as GANs [10, 22], Variational Autoencoder (VAE) [16, 8], and other image editing methods [40, 38], can both serve to expand the diversity of training samples. These methods aim to enhance the robustness and generalization capabilities of models. For example, Mixup [44] extends the training distribution by linearly interpolating random pairs of examples and labels. Cutmix [43] cuts out a patch from one image and replaces it with the corresponding patch from another image. Remix [14] solves the problem of category imbalance by assigning larger weights to small samples of category labels.

Model-based DG methods. From the *model* perspective, a common approach involves training and integrating models that are specific to multiple domains. For example, Xu *et al.* [39] leveraged the low-rank structure extracted from multiple latent domains to address the challenges of DG. Mancini *et al.* [24] utilized information from robust classification models to construct specific classifiers for different source domains and then optimally combined them to build a classification model for the target domain. However, many of these methods introduce complex network parameter learning, which, in turn, complicates network optimization and convergence across multiple source domain settings.

Optimization-based DG methods. From the *optimization* perspective, many methods were proposed to learn generalized features by designing different training strategies. For example, some work focuses on learning domain-invariant feature representations through explicit feature alignment[26, 9, 22, 12, 3], adversarial learning[7, 22, 23, 41], meta-learning-based methods[4, 6, 20, 21], or gradient-based methods[20, 17, 35], etc. Li *et al.* [22] designed a conditional invariant adversarial network to learn domain-invariant representations and ensure the invariance of the joint distribution across domains. Zhang *et al.* [45] dealt with domain generalization from the training scheme perspective and proposed a target-specific normalization method to further boost the generalization ability in the unseen target domain.

The aforementioned data-based methods aim to learn domain-invariant features by expanding the data distribution of source domain, whereas model-based methods seek to achieve the same goal by aggregating data distributions from various source domains. However, these methods suffer from two main drawbacks as follows. (i) The generated source domain may have significant deviations, which is not conducive to learning domain invariant features; (ii) They tend

to involve extensive learning of network parameters, complicating network optimization and convergence. To address these issues, we propose a new gradient-based method from the optimization perspective in this work.

2.2 Gradient-based Methods

Gradients are typically the driving force behind the training and optimization process in deep learning algorithms. However, when attempting to train a single model using data from multiple distributions, gradient conflict usually occurs, thereby leading to some suboptimal solutions. In DG, conflicting gradients usually contain domain-specific information, which can be detrimental to learning domain-invariant features. Existing methods to resolve gradient conflicts can be categorized into gradient alignment and gradient modification.

Gradient alignment. Gradient alignment is integrated into the learning objective. For example, Shi et al. [35] explicitly optimizes the dot product between domain gradients using an efficient first-order algorithm. In a further study, Rame et al. [31] enforces the alignment of gradient variance across domains. In contrast, Li et al. [20] adopts a meta-learning approach, where the meta-objective aims to align gradients between pseudo-source and pseudo-target domain.

Gradient modification. Gradient modification means performing some gradient surgery at each gradient step. For example, PCGrad [42] addresses conflicts for multi-task learning by projecting a gradient of the task onto the orthogonal plane defined by the gradients of other conflicting tasks. In DG, the masking of gradient components was proposed to exhibit conflicting signs across domains [25, 28]. Shahtalebi *et al.* [34] further extended the above approach by introducing a smoothed-out masking technique to keep agreement among gradient magnitudes. These existing methods ignore the granularity of the gradient, resulting in low frequency or low rate of the gradient modification during training. To address this issue, a gradient correction scheme is proposed from the perspective of gradient granularity in this work.

3 Preliminaries and Analysis

3.1 Basic Concepts and Definition

In DG, we have access to a training set composed of N source domains $\mathcal{D}_s = \{D_1, D_2, \cdots, D_N\}$, where the i^{th} domain is characterized by a dataset $D_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{M_i}$ containing M_i labeled data points, and all domains have the same number of classes. The aim is to learn a classification $f(x_j^{(i)}; \theta)$ that predicts the class label $\hat{y}_j^{(i)}$ corresponding to the input $x_j^{(i)}$ by only using the source domains, so that it is able to generalize well on unseen target domains $\mathcal{D}_t$. For multiple source domains, we define the training cost function as the average loss over all source domains.

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(\theta) \tag{1}$$

where $\mathcal{L}_i(\theta)$ represents the loss associated with the i^{th} domain, defined as follows:

$$\mathcal{L}_i(\theta) = \frac{1}{M_i}\sum_{i=1}^{M_i}\ell\left(f(x_j^{(i)};\theta), y_j^{(i)}\right) \tag{2}$$

where $\ell(\cdot,\cdot)$ is a classification loss function, *e.g.* cross-entropy, which is used to measure the error between the predicted label $\hat{y}$ and the true label y.

For ease of description, the *modification frequency* f_m is defined as follows:

$$f_m = \frac{n_f}{n_t} \tag{3}$$

where n_f and n_t represent the number of occurrences of the modification gradient and the total number of occurrences, respectively.

3.2 Orthogonal Analysis

In this section, we propose a series of mathematical proofs aimed at obtaining the conclusion that two random high-dimensional vectors are nearly orthogonal. This conclusion reflects the limitations of existing gradient-based methods. Our proposed method aims to specifically address these limitations.

To demonstrate that two random high-dimensional vectors are nearly orthogonal, we denote the angle between the two vectors by θ. Due to the isotropy, we fix a vector as follows.

$$\boldsymbol{y} = (1, 0, \ldots, 0) \tag{4}$$

Without loss of generality, we consider the random vector as:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \tag{5}$$

Then x is transformed to hypersphere coordinates:

$$\begin{cases} x_1 = \cos(\phi_1) \\ x_2 = \sin(\phi_1)\cos(\phi_2) \\ x_3 = \sin(\phi_1)\sin(\phi_2)\cos(\phi_3) \\ \quad\vdots \\ x_{n-1} = \sin(\phi_1)\cdots\sin(\phi_{n-2})\cos(\phi_{n-1}) \\ x_n = \sin(\phi_1)\cdots\sin(\phi_{n-2})\sin(\phi_{n-1}) \end{cases} \tag{6}$$

where $\phi_{n-1} \in [0, 2\pi)$ and the rest $\phi_{1,2,\ldots,n-2} \in [0, \pi]$. Meanwhile, the angle between x and y is derived as:

$$\arccos\langle x, y\rangle = \arccos\cos(\phi_1) = \phi_1 \tag{7}$$

It means that the angle between the two is ϕ_1. Then the probability that the angle between x and y does not exceed θ is derived as:

$$\begin{aligned} P_n(\phi_1 \le \theta) &= \frac{\int_0^{2\pi}\cdots\int_0^{\pi}\int_0^{\theta}\Delta d\phi_1 d\phi_2\cdots d\phi_{n-1}}{\int_0^{2\pi}\cdots\int_0^{\pi}\int_0^{\pi}\Delta d\phi_1 d\phi_2\cdots d\phi_{n-1}} \\ &= \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\sqrt{\pi}}\int_0^{\theta}\sin^{n-2}\phi_1 d\phi_1 \end{aligned} \tag{8}$$

where Δ represents the integral on the n-dimensional hypersphere $\sin^{n-2}(\phi_1)\sin^{n-3}(\phi_2)\cdots\sin(\phi_{n-2})$ and the probability density function of θ is given as:

$$p_n(\theta) = \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\sqrt{\pi}}\sin^{n-2}\theta \tag{9}$$

From Equation (9), it can be found that the maximum probability is $\theta = \frac{\pi}{2}$, and $\sin^{n-2}\theta$ is about $\theta = \frac{\pi}{2}$ symmetric, thus its mean is also $\frac{\pi}{2}$.

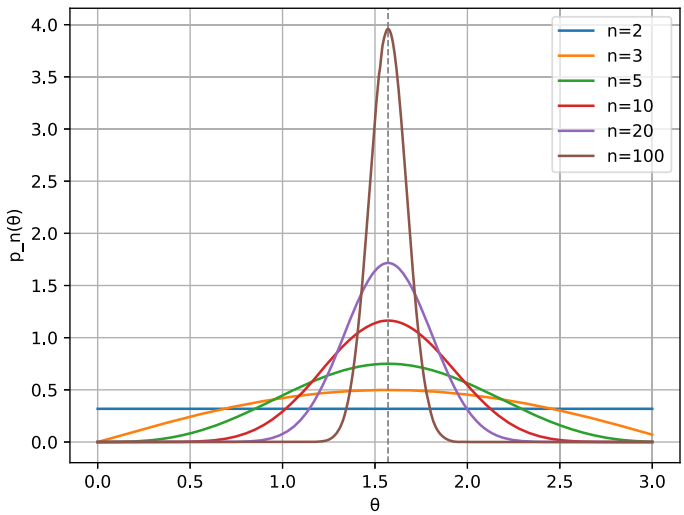


Figure 3: Angle distribution of vectors with increasing vector dimension.

However, this does not adequately describe the distribution. We still need to consider the variance:

$$Var_n(\theta) = \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})\sqrt{\pi}} \int_0^{\pi} (\theta - \frac{\pi}{2})^2 \sin^{n-2}\theta d\theta \qquad (10)$$

The analytical solution for this integral is quite complex, so we provide a partial numerical solution in the following Figure 3.

Based on the above analysis, we can infer that as the dimension increases, the angle between any two high-dimensional vectors tends to approach orthogonality.

3.3 Gradient Modification Criterion

Inspired by [29], suppose that three gradients $\{g_1, g_2, g_3\}$ corresponding to three instances $\{z_1, z_2, z_3\}$ belong to three different domains (shown in Figure 4). Since $\theta_2 < 90°$, taking a step along g_2 or g_3 can improve the classifier performance on both z_2 and z_3. That means that z_2 and z_3 contain some shared information (*i.e.* , domain-invariant features) recognized by the classifier. In contrast, considering $\theta_3 > 90°$, updating one gradient along z_1 or z_3 may degrade the classifier performance due to the low-level information sharing between z_1 and z_3.

Following the g_2 direction in this example seems to be the best choice because it can classify z_3 well without affecting the performance of z_1 too much, which means that its gradient update direction is the most favorable for learning domain-invariant features.

Through the above analysis, we can summarize the criterion of gradient modification as follows. (i) when the inner product between gradients from different source domains is less than 0, these gradients may contain domain-invariant information and do not require modification. (ii) Conversely, when the inner product is greater than 0, the components in conflicting directions may contain domain-specific information, which is detrimental to model learning and should be eliminated through gradient modification.

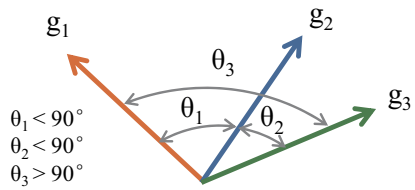


Figure 4: Example of three gradient instances.

3.4 Layering and Slicing

In this section, we firstly propose a layering and slicing scheme from the perspective of gradient granularity. Then, an adaptive smoothing strategy is proposed to smooth dominant gradients, and the Gram-Schmidt orthogonalization strategy is introduced to eliminate the conflict component. Finally, the pseudo-code of the overall procedure is given in Algorithm 1.

Based on the analysis in Section 3.2, we have shown that high-dimensional vectors tend to be more orthogonal, thereby leading to a low modification ratio. To address this issue, we propose two modules to reduce dimensions from the perspective of gradient granularity: layering and slicing.

(i) **Layering module:** Due to the significant differences between different layers in gradient magnitudes, the gradients should be processed from different layers separately. Significantly, the concept of layering here is different from the existing layer-to-layer gradient back-propagation methods, and the proposed layering module is an improvement of the existing coarse-grained methods. The reason is that the proposed module no longer calculates the gradients of the conflicting model parameters as a whole, but divides the corresponding gradients according to different layers.

(ii) **Slicing module:** Considering that high-dimensional vectors are prone to orthogonality and the gradients within the same layer should be further sliced, we propose a conceptually clear and operationally simple implementation that requires only one line of code, specifically, for the original gradient of high dimensions in a layer, we split it equally into n vectors, each of which is of the same length but of lower dimensions.

The proposed layering and slicing scheme can effectively reduce the vector dimensions, thus ensuring a wider distribution of vector angles θ. As shown in Figure 5, higher dimensional vectors tend to exhibit lower modification ratio due to higher orthogonality. After being processed by the proposed layering and slicing scheme, the angle θ between vectors no longer tends to be orthogonal. The obtained angles are more widely distributed, and the modification ratio increases as the vector dimensions decrease.

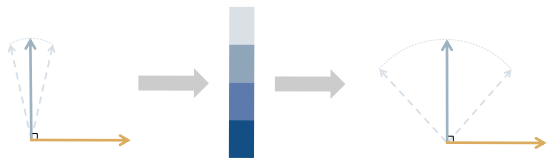


Figure 5: Comparison of vector angle distribution between before and after slicing. The dashed lines illustrate potential angular distributions, while different colored blocks represent the gradients divided by layer. In this schematic, the gradients of all parameters of a model are categorized into four blocks based on layers.

3.5 Adaptive Smoothing

During the training process of multi-source domains, the existence of dominant gradients (*i.e.* , the gradient from one source domain significantly surpasses the gradients from other source domains) can lead to the model being biased toward specific source domain, thereby introducing an excessive amount of domain-specific information, which does not help to learn domain-invariant features. To address this issue, an adaptive smoothing strategy is proposed to mitigate

the influence of dominant gradients. For ease of description, the gradient of the i^{th} source domain is denoted as $g_i = \nabla_\theta \mathcal{L}_i(\theta)$. The L_2 norm of gradients is used to compare the magnitude of gradients $\|g\|_2 = \sqrt{(g_1^2 + g_2^2 + \cdots + g_n^2)}$. Considering that the magnitude of different gradients as different observations in the same distribution, the outlier gradient (*i.e.* , dominant gradient) is defined as the gradient that differs from the mean of the distribution by one standard deviation. In addition, the Equation (11) is used to smooth the dominate gradient.

$$g_i' = \alpha g_i + (1-\alpha) g_{mean} \tag{11}$$

where α is a smoothing hyper-parameter and g_{mean} is the mean of the gradient across source domains.

3.6 Conflict Component Elimination

Since the conflicting direction component of the gradient contains domain-specific information that is not conductive to model learning, it should be eliminated as much as possible. The Gram-Schmidt process, known for efficiently orthogonalizing sets of vectors in inner product spaces, is widely used to construct orthogonal bases [19, 42]. Inspired by this, we employ Gram-Schmidt orthogonalization in this work to eliminate conflicting components, defined as:

$$g_i' = g_i - g_{(i \to j)} \tag{12}$$

where g_i represents a conflicting gradient. $g_{(i \to j)}$ represents the conflicting direction component of g^i with respect to g^j, defined as:

$$g_{(i \to j)} = \frac{\langle g_i, g_j \rangle}{\|g_j\|_2} g_j \tag{13}$$

Through the Gram-Schmidt orthonormalization, we project the gradient g_i onto the normal plane of the gradient g_j. This amounts to removing the conflicting component $g_{(i \to j)}$, thereby leading to the model learning domain-invariant features instead of being biased towards a specific domain.

3.7 Overall Procedure

Suppose that the gradient of the k^{th} layer for the domain D_i is g_i^k, and the gradient for the domain D_j is g_j^k. The processing step of the proposed CorGrad is listed as follows.

(1) A gradient g_i^k is divided into $\{g_i^{k_1}, g_i^{k_2}, \cdots g_i^{k_n}\}$ by using the proposed slicing module.
(2) If $g_i^{k_n} \cdot g_j^{k_n} < 0$, we replace $g_i^{k_n}$ with its projection onto the normal plane of $g_j^{k_n}$: $g_i^{k_n} = g_i^{k_n} - \frac{g_i^{k_n} \cdot g_i^{k_n}}{\|g_i^{k_n}\|^2} \cdot g_j^{k_n}$ else the original gradient $g_i^{k_n}$ remains unchanged.
(3) The CorGrad is repeated this process across all of the other domains sampled in random order from the current batch $D_j \forall j \neq i$, resulting in the gradient g_i^{PC} that is applied for domain D_i.
(4) For $\{g_i^{PC}, g_j^{PC}, \cdots, g_m^{PC}\}$, we calculate g_{mean}^{PC} and g_{std}^{PC} by $\{\|g_i^{PC}\|_2, \|g_j^{PC}\|_2, \cdots, \|g_m^{PC}\|_2\}$.
(5) It determines whether g^{PC} is outlier gradient by computing $g^{PC} - g_{mean}^{PC} - g_{std}^{PC}$, where positive values indicate outlier gradient.
(6) If g^{PC} is an outlier gradient, we replace g^{PC} with a smoother value $g^{PC} = \alpha g^{PC} + (1-\alpha) g_{mean}^{PC}$. We perform the same procedure for all domains in the batch to obtain their respective gradients.

The same procedure is conducted for all domains in the batch to obtain their respective gradients, and the overall update procedure is illustrated in Algorithm 1.

Algorithm 1: CorGrad Update Rule

Input: Model parameters θ, N source domain $\mathcal{D} = \{D_1, D_2, \cdots, D_N\}$, Model layers $L = \{l_1, l_2 \cdots l_n\}$

$g_d \leftarrow \nabla_\theta \mathcal{L}_d(\theta) \quad \forall d$;
$\{g_d^{l_1}, g_d^{l_2} \cdots g_d^{l_n}\} \leftarrow g_d \quad \forall d$;
for $\mathcal{D}_i \in \mathcal{D}$ **do**
  for $\mathcal{D}_j \overset{uniformly}{\sim} \mathcal{D} - \mathcal{D}_i$ *in random order* **do**
    for $l_k \in L$ **do**
      if $g_i^{l_k} \cdot g_j^{l_k} < 0$ **then**
        Set $g_i^{l_k} = g_i^{l_k} - \frac{g_i^{l_k} \cdot g_j^{l_k}}{\|g_i^{l_k}\|^2}$
  $g_{std}, g_{mean} \leftarrow \{g_1, g_2, \cdots, g_N\}$;
  for $g_i \in \{g_1, g_2, \cdots, g_N\}$ **do**
    if $g_i - g_{mean} > g_{std}$ **then**
      $g_i = \alpha g_i + (1-\alpha) g_{mean}$
return update $\Delta\theta = \sum_i g_i$

4 Experiments

4.1 Experiments on DomainBed

Datasets. Five distinct datasets from the DomainBed [11] benchmark are used to conduct a comprehensive evaluation of the proposed method. These datasets include PACS, VLCS, OfficeHome, TerraInc, and DomainNet. **(1) PACS** consists of 9,991 images categorized into 7 classes, and is widely utilized in domain generalization literature due to its substantial distributional shift across four domains, which includes art painting, cartoon, photo, and sketch. **(2) VLCS** comprises 10,729 images collected from five different classes. These images are originally sourced from four separate datasets, namely PASCAL VOC 2007, LabelMe, Caltech, and Sun. In the domain generalization context, each of these datasets is considered a distinct domain.**(3) OfficeHome** is an object recognition dataset with 15,588 images that span 65 classes, which can be further divided into four domains: artistic, clipart, product, and real-world. **(4) TerraInc** encompasses 24,788 animal images captured in various wilderness locations. There are a total of 10 classes, with the specific location serving as the varying domain, denoted as L100, L38, L43, and L46. **(5) DomainNet** boasts 586,575 images distributed across 345 classes, with domains categorized into six types: clipart, infograph, painting, quickdraw, real, and sketch.

Implementation details. Following the prevalent design, we use the ImageNet pre-trained ResNet18 model as the backbone for all datasets. The number of slices n is set dynamically $n \in [5, 100]$ in the proposed CorGrad and the weight parameter for the smooth dominant gradient is initially dynamically set to $\alpha \in [0.5, 1]$ and gradually decays to 0 with the iteration rounds. For all datasets, we assess these methods employing a leave-one-out strategy. This strategy designates one domain as the target domain, which is held out for evaluation, while considering the other domains as the source domains for training and testing. Specifically, following the relevant settings in the DomainBed benchmark, the training and test samples are randomly split in the ratio of 8:2, the batch size is 32 by default, except for the DomainNet dataset, which is trained with 15,000 iterations, and the remaining four datasets are trained with 5,000 iterations. The remaining hyper-parameters such as learning rate, weight decay, *et al.* are dynamically adjusted according to [11].

Table 1: Comparisons with state-of-the-art domain generalization methods. Table shows the out-of-domain accuracy of the five domain generalizations using ResNet18 as a backbone. The **best result** is highlighted in bold. Top5 accumulates the number of datasets where a method achieves the top 5 performances. The results marked by † are copied from previous work. The average accuracy and standard deviation are calculated from five trails.

Algorithm	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Avg.	Top5
MMD† [22]	81.3 ± 0.8	74.9 ± 0.5	59.9 ± 0.4	42.0 ± 1.0	7.9 ± 6.2	53.2	0
RSC† [13]	80.5 ± 0.2	75.4 ± 0.3	58.4 ± 0.6	39.4 ± 1.3	27.9 ± 2.0	56.3	0
IRM† [2]	80.9 ± 0.5	75.1 ± 0.1	58.0 ± 0.1	38.4 ± 0.9	30.4 ± 1.0	56.6	0
ARM† [46]	80.6 ± 0.5	75.9 ± 0.3	59.6 ± 0.3	37.4 ± 1.9	29.9 ± 0.1	56.7	0
DANN† [7]	79.2 ± 0.3	76.3 ± 0.2	59.5 ± 0.5	37.9 ± 0.9	31.5 ± 0.1	56.9	1
GroupGRO† [33]	80.7 ± 0.4	75.4 ± 1.0	60.6 ± 0.3	41.5 ± 2.0	27.5 ± 0.1	57.1	0
CDANN† [23]	80.3 ± 0.5	76.0 ± 0.5	59.3 ± 0.4	38.6 ± 2.3	31.8 ± 0.2	57.2	0
VREx† [18]	80.2 ± 0.5	75.3 ± 0.6	59.5 ± 0.1	43.2 ± 0.3	28.1 ± 1.0	57.3	1
CAD† [32]	81.9 ± 0.3	75.2 ± 0.6	60.5 ± 0.3	40.5 ± 0.4	31.0 ± 0.8	57.8	1
CondCAD† [32]	80.8 ± 0.5	76.1 ± 0.3	61.0 ± 0.4	39.7 ± 0.4	31.9 ± 0.7	57.9	0
MTL†[5]	80.1 ± 0.8	75.2 ± 0.3	59.9 ± 0.5	40.4 ± 1.0	35.0 ± 0.0	58.1	0
ERM†[37]	79.8 ± 0.4	75.8 ± 0.2	60.6 ± 0.2	38.8 ± 1.0	35.3 ± 0.1	58.1	0
MixStyle†[47]	82.6 ± 0.4	75.2 ± 0.7	59.6 ± 0.8	40.9 ± 1.1	33.9 ± 0.1	58.4	1
PCGrad[25]	81.3 ± 0.2	75.0 ± 0.3	58.3 ± 0.5	41.3 ± 0.1	36.4 ± 0.4	58.5	1
Mixup†[40]	79.2 ± 0.9	76.2 ± 0.3	61.7 ± 0.5	42.1 ± 0.7	34.0 ± 0.0	58.6	1
MLDG[20]	81.2 ± 0.4	75.6 ± 0.3	61.2 ± 0.2	40.1 ± 0.4	35.9 ± 0.8	58.8	1
Fishr[31]	81.4 ± 0.5	76.0 ± 0.2	61.2 ± 0.3	42.6 ± 1.0	34.3 ± 0.3	59.1	1
SagNet† [27]	81.7 ± 0.6	75.4 ± 0.8	62.5 ± 0.3	40.6 ± 1.5	35.3 ± 0.1	59.1	1
SelfReg† [15]	81.8 ± 0.3	76.4 ± 0.7	62.4 ± 0.1	41.3 ± 0.3	34.7 ± 0.2	59.3	2
Fish[35]	81.7 ± 0.5	**76.9** ± 0.4	62.2 ± 0.4	40.3 ± 0.4	35.5 ± 0.3	59.3	2
CORAL† [36]	81.7 ± 0.0	75.5 ± 0.4	62.4 ± 0.4	41.4 ± 1.8	36.1 ± 0.2	59.4	2
SD† [30]	81.9 ± 0.3	75.5 ± 0.4	**62.9** ± 0.2	42.0 ± 1.0	36.3 ± 0.2	59.7	3
CorGrad(Ours)	**82.1** ± 0.4	76.6 ± 0.2	61.2 ± 0.3	**44.9** ± 0.5	**37.3** ± 0.6	**60.4**	**4**

Experimental results. The average out-of-domain performance and the Top5 scores of the latest DG methods across the five benchmarks are presented in Table 1. We note that the ERM method achieves good performance compared to the existing arts. In fact, as a strong baseline, the ERM outperforms half of the methods in terms of average accuracy. It is obvious that the proposed CorGrad consistently outperforms the ERM [37] on all benchmarks, showing an average improvement of +2.3%. Furthermore, the proposed CorGrad can obtain varying degrees of improvement over existing gradient-based methods: +1.9% compared to PCGrad [25], +1.6% over MLDG [20], +1. 3% over Fishr [31] and +1. 1% over Fish [35]. Notably, CorGrad achieves the best performance in 3 out of the 5 benchmarks, surpassing other state-of-the-art DG methods, and ranks within the top 5 in 4 benchmarks.

4.2 Modification Ratio Analysis

To better illustrate the limitations of the coarse-grained method in relation to modification ratios, we present the results of our method compared to the coarse-grained method in five different data sets in Figure 6. The modification ratio is defined as the ratio of the magnitude of the conflicting direction component of the gradient to the magnitude of the gradient itself. It is worth noting that the overall trend of the model's correction rate on all five different datasets is gradually decreasing, due to the fact that during the model training process, the disturbing information from a particular source domain is continuously reduced, gradually approaching the ideal domain-invariant features. It is obvious that our method consistently outperforms the coarse-grained method in terms of modification ratio on all datasets, reaffirming the conclusions detailed in Section 3.2. High-dimensional vectors are prone to orthogonality, resulting in fewer conflict direction components and smaller modification ratios, which are not conducive to model training.

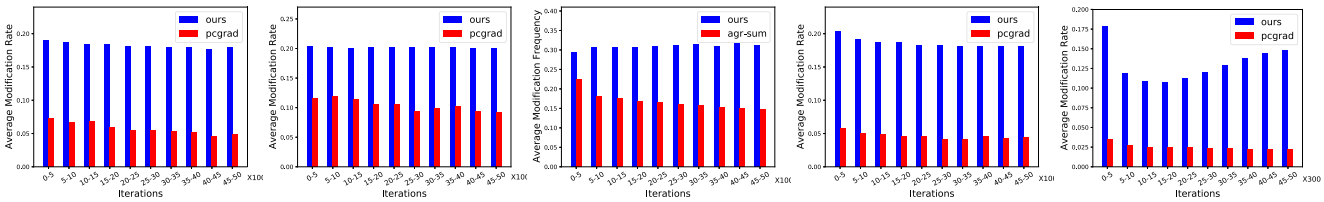
4.3 Modification Frequency Analysis

To elucidate the limitations of the fine-grained method in terms of modification frequency, we present a comparative analysis between our method and the fine-grained method in five distinct datasets in Figure 7. The modification frequency is defined as the ratio of the number of times a modification gradient event occurs to the total number of occurrences. It is important to note that for the first four datasets, the initial modification frequency of the fine-grained method is approximately $\frac{1}{4}$. The reason is that three source domains were utilized when employing the leave-one-out method, which theoretically results in an initial modification frequency of $\frac{1}{2^{N-1}}$, where N represents the number of source domains. However, in the case of the DomainNet dataset with five initial source domains, the initial modification frequency should be close to $\frac{1}{16}$. Notably, due to the very low initial modification frequency, subsequent model modifications become very small, leading to the stagnation of subsequent gradient updates. It is obvious that our method, which modifies gradient granularity, can substantially boost the modification frequency compared to the fine-grained method. This also proves our earlier conjecture that the fine-grained method will result in frequent collisions, thereby leading to a lower modification frequency.

4.4 Ablation Study

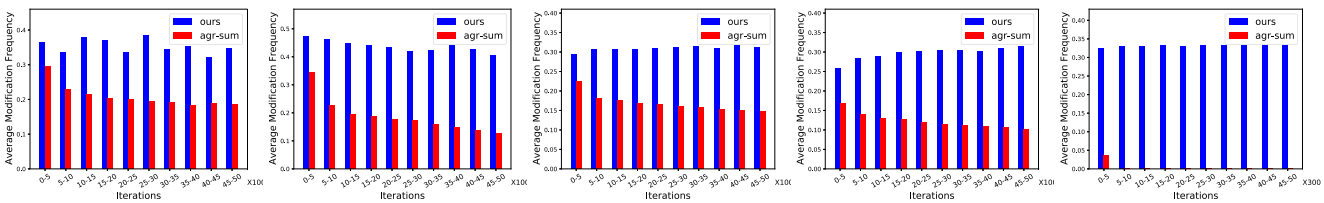
In the subsequent sections, we perform an ablation study to assess the efficacy of the individual modules within CorGrad. All experiments are conducted in this segment are based on the extensively employed DomainBed benchmark. We present the average accuracy results for various methods on the VLCS, TerraInc, and DomainNet datasets, employing ResNet18 as the backbone. For specific experimental details, please refer to Section 4.1. The leave-one-domain strategy is employed, and the accuracy obtained represents the averages obtained from five independent trials.

The following observations can be derived from Table 2. Firstly, PCGrad [25] and our method can both obtain significant outperformance compared to ERM, verifying the effectiveness of the gradient



(a) PACS (b) VLCS (c) OfficeHome (d) TerraInc (e) DomainNet

Figure 6: Visualized results of the modification ratio by applying our method and the coarse-grained method on five datasets. The horizontal coordinate is averaged into ten groups based on the number of iterations, and the vertical coordinate is the average modification ratio, which is represented by using the ratio of the gradient's conflicting direction component to the modulus of the gradient itself.



(a) PACS (b) VLCS (c) OfficeHome (d) TerraInc (e) DomainNet

Figure 7: Visualized results of the modification frequency by applying our method and the fine-grained method on five datasets. The horizontal coordinate is evenly divided into ten groups according to the number of iterations, and the vertical coordinate is the average modification frequency.

Table 2: Ablation study by using the proposed CorGrad on three benchmarks: VLCS, TerraInc, and DomainNet. Here the "L" and "S" denote the hierarchical and slicing operations performed on the gradient granularity, respectively. "D" denotes the adaptive smoothing of the dominant gradient. The obtained precision (%) and standard deviation are calculated using ResNet18 as backbone based on 5 trials for each target domain.

Models	granularity		dominate	Test datasets			Avg.
	L	S	D	VLCS	TerraInc	DomainNet	
ERM	-	-	-	75.8 ± 0.2	38.8 ± 1.0	35.3 ± 0.1	50.0
PCGrad	-	-	-	75.0 ± 0.3	41.3 ± 0.1	36.4 ± 0.4	50.9
Ours w/o S,D	✓	-	-	75.4 ± 0.5	42.9 ± 0.6	36.6 ± 0.4	51.6
Ours w/o D	✓	✓	-	75.9 ± 0.4	44.6 ± 0.6	36.9 ± 0.2	52.4
CorGrad(Ours)	✓	✓	✓	76.6 ± 0.2	44.9 ± 0.5	37.3 ± 0.6	53.0

conflict strategy. Secondly, our method can solve the problem of conflicting gradients using Gram-Schmidt orthogonalization, and it can remove conflicting directional components, thereby learning better domain invariant features. In particular, the incremental inclusion of operations demonstrates tangible improvements: Hierarchical consideration of the overall gradient alone raises the average accuracy by +0.7% ($50.9\% \rightarrow 51.6\%$). Further enhancements through the slicing operation boost accuracy by an additional +0.8% ($51.6\% \rightarrow 52.4\%$). The proposed adaptive smoothing strategy can prevent biased learning towards specific source domains, increasing the average accuracy by another +0.6% ($52.4\% \rightarrow 53.0\%$). In general, the proposed CorGrad can obtain an average performance +2.1% better than the PCGrad. This suggests that modifying the granularity of the gradient not only increases the modification frequency and ratio, but also improves final accuracy. Thirdly, across the board, the proposed CorGrad consistently surpasses ERM and PCGrad significantly. The results verify the effectiveness of the proposed method in addressing the challenges of domain generalization.

5 Conclusion

In this paper, we first analyze the limitations of existing gradient modification methods from the perspective of gradient granularity in the area of DG, showing that coarse and fine grained methods can lead to a low ratio and a low frequency of gradient modification, respectively. To address this issue, a gradient modification method, called CorGrad, is proposed by selecting the appropriate granularity to increase the modification frequency and the modification ratio. In addition, an adaptive smoothing strategy is proposed to smooth out the influences of the dominant gradient. Finally, extensive experiments are conducted to verify the effectiveness of the proposed CorGrad, and the results show that the proposed CorGrad method can obtain competitive performance, and outperform most state-of-the-art algorithms in five DG benchmarks. Especially, using ResNet18 as the backbone, it can achieve an average performance of 60.4% on DomainBed without using any additional information, demonstrating its strong DG capabilities.

Acknowledgments

The authors would like to express their appreciation to the referees for their helpful comments and suggestions. This work was supported in part by the National Natural Science Foundation of China (Grant nos. 62373324 and 62271448), and in part by the Zhejiang Provincial Natural Science Foundation of China (Grant no. LGF22F030016).

References

[1] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. Fowlkes, S. Soatto, and P. Perona. Task2vec: Task embedding for meta-learning.

In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6429–6438, 2019.

[2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] Y. Atzmon, F. Kreuk, U. Shalit, and G. Chechik. A causal view of compositional zero-shot recognition. *Advances in Neural Information Processing Systems*, 33:1462–1473, 2020.

[4] Y. Balaji, S. Sankaranarayanan, and R. Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

[5] G. Blanchard, A. A. Deshmukh, Ü. Dogan, G. Lee, and C. Scott. Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1):46–100, 2021.

[6] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in neural information processing systems*, 32, 2019.

[7] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1): 2096–2030, 2016.

[8] Z. Ge, Z. Song, X. Li, and L. Zhang. Meta conditional variational auto-encoder for domain generalization. *Computer Vision and Image Understanding*, 222:103503, 2022.

[9] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[11] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[12] S. Hu, K. Zhang, Z. Chen, and L. Chan. Domain generalization via multidomain discriminant analysis. In *Uncertainty in Artificial Intelligence*, pages 292–302. PMLR, 2020.

[13] Z. Huang, H. Wang, E. P. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020.

[14] S. Jiang, Z. Lin, Y. Li, Y. Shu, and Y. Liu. Flexible high-resolution object detection on edge devices with tunable latency. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 559–572, 2021.

[15] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] M. Koyama and S. Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. 2020.

[18] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[19] S. J. Leon, Å. Björck, and W. Gander. Gram-schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications*, 20 (3):492–532, 2013.

[20] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[21] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455, 2019.

[22] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[23] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 624–639, 2018.

[24] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE international conference on image processing (ICIP)*, pages 1353–1357. IEEE, 2018.

[25] L. Mansilla, R. Echeveste, D. H. Milone, and E. Ferrante. Domain generalization via gradient surgery. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6610–6618, 2021.

[26] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR, 2013.

[27] H. Nam, H. Lee, J. Park, W. Yoon, and D. 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.

[28] G. Parascandolo, A. Neitz, A. Orvieto, L. Gresele, and B. Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

[29] D. Peng and S. J. Pan. Learning gradient-based mixup towards flatter minima for domain generalization. *arXiv preprint arXiv:2209.14742*, 2022.

[30] M. Pezeshki, O. Kaba, Y. Bengio, A. C. Courville, D. Precup, and G. Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

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

[32] Y. Ruan, Y. Dubois, and C. J. Maddison. Optimal representations for covariate shift. *arXiv preprint arXiv:2201.00057*, 2021.

[33] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[34] S. Shahtalebi, J.-C. Gagnon-Audet, T. Laleh, M. Faramarzi, K. Ahuja, and I. Rish. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. *arXiv preprint arXiv:2106.02266*, 2021.

[35] Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[36] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016.

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

[38] Q. Xu, R. Zhang, Y. Zhang, Y. Wang, and Q. Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021.

[39] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*, pages 628–643. Springer, 2014.

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

[41] F.-E. Yang, Y.-C. Cheng, Z.-Y. Shiau, and Y.-C. F. Wang. Adversarial teacher-student representation learning for domain generalization. *Advances in Neural Information Processing Systems*, 34:19448–19460, 2021.

[42] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[43] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[44] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[45] J. Zhang, L. Qi, Y. Shi, and Y. Gao. Generalizable semantic segmentation via model-agnostic learning and target-specific normalization. *arXiv preprint arXiv:2003.12296*, 2(3):6, 2020.

[46] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678, 2021.

[47] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.