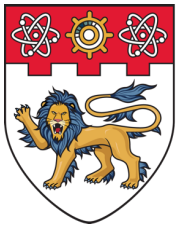
Explore the Influential Samples in Domain Generalization



NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Zike Wu

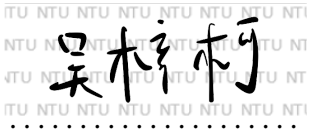
School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Engineering

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.



11/08/2023

Date

Zike Wu

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

11/08/2023

.....

Date

.....

Associate Prof. Hanwang Zhang

Authorship Attribution Statement

This thesis contains materials from 1 paper that is currently submitted to a conference in which I am listed as the first author.

It is submitted to AAAI 2023: **Zike Wu**[*], Jiaxin Qi[*], Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. DOMAIN+: Splitting a New Influential Domain for Domain Generalization.[1] *Under Review*.

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang designs the research topic and joins the paper writing.
- Prof. Qianru Sun polishes the paper.
- Prof. Xian-Sheng Hua polishes the paper.
- Co-first author Jiaxin Qi participates in the idea discussion and algorithm design, takes charge of part of the code implementation, and polishes the paper.
- I participate in the idea discussion and algorithm design, take charge of the code implementation and write the draft paper.

11/08/2023

.....

Date

.....

Zike Wu

[1]The superscript $*$ indicates the equal contributions

Acknowledgements

I would like to begin by extending my heartfelt appreciation to my supervisor, Prof. Hanwang Zhang. His support and mentorship have been instrumental in guiding me through my journey. He not only gave me the opportunity to further my education, but also provided invaluable encouragement and assistance in moments of doubt and adversity. His insights and professionalism have always been a beacon, guiding my work. Prof. Zhang's faith in our decisions and respect for our autonomy speaks volumes about his charisma. I carry with me the wisdom of his words, and they will continue to inspire me, regardless of my future endeavors.

My gratitude extends to my collaborators — Dr. Jiaxin Qi, Dr. Panzhou, Prof. Kenji Kawaguchi, Prof. Qianru Sun, and Prof. Xian-Sheng Hua. Their valuable insights and support have been indispensable. I should also recognize the incredible community at MReal Lab, especially friends like Yucheng Han, Beier Zhu, Xuanyu Yi, Yuxuan Wang, Xiaoyuan Liu, Tan Wang, Zhongqi Yue, and others. Their camaraderie made this journey even more memorable. A special mention to my girlfriend, Xuanye Chen, whose unwavering support has been a bedrock for me over the past many years.

Finally, I would like to express my deepest gratitude to my parents, Sijiu Wu and Min Zhang. Their constant encouragement and faith in me, allowing me the freedom to chart my own course, have been the pillars of my strength throughout this journey.

Zike Wu, August 2023

Abstract

Domain Generalization (DG) aims to learn a model that generalizes in testing domains unseen from training. All DG methods assume that the domain-invariant features can be learned by discarding the domain-specific ones. However, in practice, the learned invariant features usually contain "spurious invariance" that is only invariant across training domains but still variant to testing ones. We point out that this is because the contribution of the minority training samples without such spurious invariance is outgunned. Therefore, we are motivated to split these samples out of the original domains to form a new one, to which the spurious invariance is no longer invariant and thus removed. We present a cross-domain influence-based method, DOMAIN+, to obtain the new domain. Specifically, for each sample per training domain, we estimate its influence by up-weighting it and then calculating how much the invariance loss of the other training domains changes—the more it changes, the higher the influence, and the more likely the sample belongs to the new domain. Then, with the split domains, we can deploy any off-the-shelf DG methods to achieve better generalization. We benchmark DOMAIN+ on DOMAINBED and show that it helps existing SOTA methods achieve new SOTAs.

Contents

List of Figures

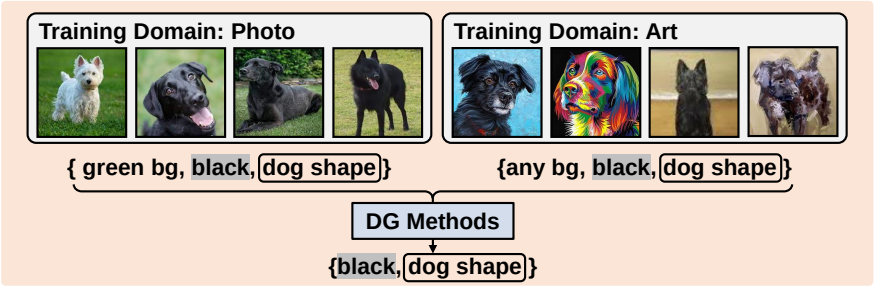
List of Tables

Chapter 1

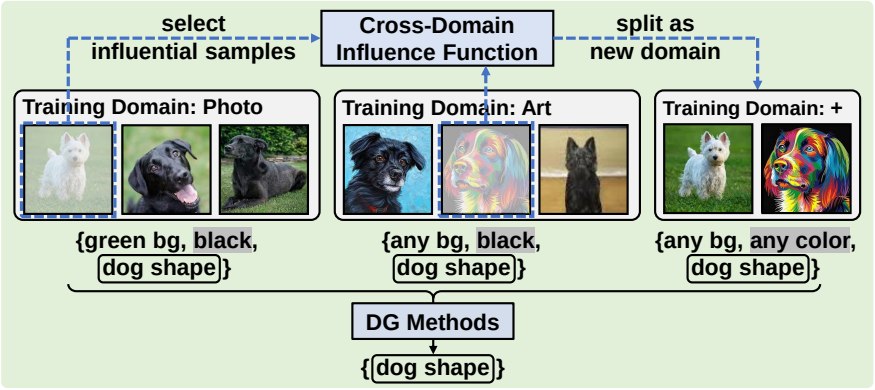
Introduction

In an era where machine learning models are increasingly deployed in diverse settings, their ability to generalize to unseen data domains is crucial [2, 3]. This challenge is central to Domain Generalization (DG), a field dedicated to developing models robust across varied and novel environments [4, 5]. Recent work shows that deep models are good at fitting training data but bad at generalizing to unseen domains [2, 3, 6, 7]. For example, when a model is trained in `Photo` domain, where most dogs are black, it will recklessly learn the color features to identify dogs, and thus it is less discriminative when the color is no longer needed, *e.g.*, tested in `Sketch` domain. In practice, models are always tested in various domains, and thus we are interested in the DG task: training a model in multiple domains to achieve *invariance*, which is generalizable in unseen domains [4, 5]. The effects of domain generalization have far-reaching implications that go beyond just theoretical considerations, which are vital to ensure the reliability of autonomous driving [8–10], medical imaging diagnostics [11, 12], *etc*.

To achieve invariance, all DG methods aim to keep the domain-invariant features (or causal features [13, 14]) by discarding the domain-specific ones [15–17]. As shown in Figure 1.1(a), {`black, dog shape`} are the invariant features obtained by DG methods as they are indeed discriminative for most training samples in both `Photo` and `Art`. Embarrassingly, the community recently finds out that the most naive Empirical Risk Minimization (ERM) objective, which simply merges the samples of all the training domains without any domain-invariant pursuit, shows competitive or even better performance compared to DG [3]. The reason is because although ERM is widely known to be easily biased by spurious training correlation [6, 15, 18–20], *e.g.*, most dogs are black, as



(a) DG methods learn spurious invariance



(b) DG methods with our DOMAIN+ learn invariance

FIGURE 1.1: Illustration of the spurious invariance (grey shade) learned by conventional DG and removed by our DOMAIN+. "bg": background, bordered word: domain-invariant feature.

long as the training samples across domains are diverse, *e.g.*, dogs with different backgrounds in `Photo` and `Art`, ERM can still remove sufficient domain-specific features, *e.g.*, it also learns {`black, dog shape`} as well as DG.

This implies that existing DG methods still lack a self-diagnostic mechanism to remove the ***Spurious Invariance*** shared by all the training domains but variant in the testing domain. We call the invariance "spurious" because it cannot be overturned by using the cross-domain validation only on the training domains [15, 16, 21], just like the underlying true invariance that is also shared by all training domains, removing which will definitely increase the training loss. In Figure 1.1(a), {`black`} is a spurious invariance because it is no longer discriminative in unseen domains without color such as `Sketch`. The reason is that the {`black`} samples prevail in all training domains over {`other color`} samples, *e.g.*, the white in `Photo` and the colorful in `Art`. Thus, these rare samples make minor contributions to counter that {`black`} is not the true invariance, because removing which will diminish the model fitting for the majority "black dog" training samples.

FIGURE 1.2: Visualizations of the samples ranked by cross-domain influence (top) and IRM loss (bottom) from low to high. Red borders denote the selected rare samples by DOMAIN+.

You may propose a straightforward remedy to up-weight those rare samples. It sounds appealing but it is hardly applicable in practice due to the following two challenges. First, it is hard to identify the sample-wise "rarity" as the notion of "invariance loss" is defined on the dataset-level to evaluate how consistently the model (or feature) behaves across domains [15, 16, 22]. So, popular hard sample mining methods only identify those with high training loss [23–25], which only quantifies how well the model fits a specific sample but not low invariance [26, 27]. Second, even if we can accurately zoom into those samples, the re-sampling of them will introduce not only the desired features (*e.g.*, {`white`} in `Photo`), but also other associated ones (*e.g.*, {`green bg`}), which may mislead the entire training [28, 29].

To address the **first challenge**, a principled way to judge if a sample has no spurious invariance is to ask the self-diagnostic counterfactual question: *If we have removed the sample, how would it affect the invariance re-trained on the new training data excluding it?* In particular, we define the answer as a sample-wise real value called ***Cross-domain Influence***: as the samples without spurious invariance are rare, if we remove one of them in one domain, the domain's spurious invariance will become more dominant, *e.g.*, the percentage of black dogs in `Photo` is higher, then the spurious invariance will be more easily achieved in the domain after training. So, such "purer" spuriousness helps other domains achieve the spurious invariance faster too—decreasing their invariance losses; in contrast, if we remove one of the majority samples with spurious invariance, it won't significantly decrease the invariance loss as the spurious invariance is still dominant. According to the above discussion, we give a formal definition of rare samples in Chapter 4.2.

However, the above "leaving one sample out and re-training" makes the cross-domain influence estimation prohibitively expensive. Thanks to the recent advances in approximating the sample influence without re-training [30, 31], we can implement our cross-domain influence by "differentiating" a sample in one domain, *i.e.*, up-weighting the sample by an infinitesimal amount, and then estimating the mean of the invariance loss changes in each other domains by a closed-form expression (Chapter 4). As shown in Figure 1.2, the influence ranking of the samples indeed tells us more about the spurious invariance than the conventional sample "hardness". For example, our high influence identifies rare dogs that are {`non-standing, abnormal action, colorful`}, which do not suggest high training loss necessarily.

Finally, to address the **second challenge**, after identifying the most influential samples, instead of re-sampling, we respect them as a new domain by splitting them from the original ones, and then use any off-the-shelf DG methods on the old domains plus the new one, and hence we dub our method DOMAIN+. As illustrated in Figure 1.1(b), DOMAIN+ can help any DG method to achieve the true invariance {`dog shape`}, which is the only invariance across the newly split domains. In Chapter 5, we use 3 classic open-sourced SOTAs: IRM [15], CORAL [17], and Fish [16], as our baselines on 4 popular datasets: PACS, VLCS, OfficeHome, and TerraIncognita. Specifically, we follow DOMAINBED [3]—a stringent and reproducible DG benchmark—to conduct all the experiments. The results show that we can consistently improve all the baselines, demonstrating that our DOMAIN+ helps DG achieve a better invariance.

Chapter 2

Literature Review

2.1 Domain Generalization

Domain Generalization (DG) trains a model on multiple training domains and tests its generalization ability in unseen domains. DG methods can be categorized into two camps: 1) Without using domain labels. They use a domain-agnostic augmentation/regularization to help the model learn more generalizable features [32–35]. However, DomainBed [3] shows that a strong ERM beats most of them, implying that these methods for improving ERM are mostly due to an unfair hyper-parameter tuning, and thus they fail to learn domain-invariant features. Therefore, we focus on the other camp in this paper. 2) Using domain labels. They use domain-wise regularization to encourage the model to learn domain-invariant features by penalizing the domain-specific features. They include: *Invariant/causal learning* [15, 21, 36, 37], which uses invariance loss to penalize the learning of different features across training domains; *Feature/gradient alignment* [16, 17, 22, 38, 39], which minimizes the distance between the features/gradients of the same class from different domains. *Adversarial learning* [40, 41], which regularizes that the learned features should not predict the domain labels, *i.e.*, the domain-specific features are removed. However, all of them suffer from the spurious invariance, which will be addressed by our Domain+.

2.2 Influence Function

Influence functions are a concept from robust statistics, used to estimate the impact of removing or altering a single data point in a statistical model [42]. This method approximates the effect on certain objectives, like testing loss, when a data point is excluded from training. Recent work has expanded the use of influence functions. These techniques have been used to measure the impact of individual samples or features on the model performance, with a special emphasis on testing loss, including re-weighting [30, 31, 43], altering features or labels [44, 45]. These methods are essentially considered an oracle tuning trick, providing insights into the influence of individual data components on the overall model.

Despite their utility, the direct application of traditional influence functions in Domain Generalization (DG) is faced with significant challenges. One major limitation is the impracticality of evaluating the model based on the oracle setting, as it violates the generalization scenario [3]. To address these limitations in the context of DG, our research introduces the concept of cross-domain influence, which relies solely on training data. This novel approach circumvents the need for extensive retraining by utilizing an efficient estimation method inspired by Koh and Liang [30]. Our cross-domain influence method is designed to assess the impact of data across different domains, offering a more practical and scalable solution for evaluating influence in complex models and large datasets.

Chapter 3

Preliminaries

Given training data $\mathcal{D}$ consisting of K domains $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$, where $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, x_i^k is a sample in domain k, y_i^k is its one-hot label, and n_k is the number of samples in $\mathcal{D}_k$. Domain Generalization (DG) aims to train a model f on $\mathcal{D}$ to predict the labels of testing samples in any unseen domains $\mathcal{D}_u$. The crux of learning f is to capture the domain-invariant (causal) features, which are invariantly discriminative in any domain, by discarding all the domain-specific features that are only discriminative in training but not testing.

3.1 Empirical Risk Minimization (ERM)

It simply merges the samples of all the training domains as a whole without domain index, *i.e.*, $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{n}$, where $n = \sum_{k=1}^{K} n_k$. ERM learns f on $\mathcal{D}$ by minimizing the softmax cross-entropy (CE) loss:

$$\mathcal{L}_{\text{ERM}}(f, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \text{CE}(y_i, f(x_i)), \tag{3.1}$$

where $f(x_i)$ is the softmax prediction of x_i. ERM can remove some domain-specific features and shows competitive performance as we mentioned in Chapter 1. The reason is that some domain-specific features are no longer dominant in the combined set. For example, in Figure 1.1(a), although `green bg` is a domain-specific feature in `Photo`,

it is less dominant in `Photo` and `Art` combined. However, ERM cannot remove dataset-specific features when most domains contain similar features, which still causes bias in unseen testing domains.

3.2 Invariant Risk Minimization (IRM)

Domain Generalization (DG) methods aim to discard the domain-specific features by additionally minimizing a penalty overall training domains, *i.e.*, the ***invariance loss*** $\mathcal{L}(f, \mathcal{D}_k)$:

$$\mathcal{L}_{\mathrm{DG}}(f) = \frac{1}{K}\sum_{k=1}^{K}[\mathcal{L}_{\mathrm{ERM}}(f, \mathcal{D}_k) + \lambda \cdot \mathcal{L}(f, \mathcal{D}_k)], \tag{3.2}$$

where $\lambda > 0$ is a trade-off hyper-parameter. For example, Invariant Risk Minimization (IRM) [15], a classic DG method, implements the invariance loss as:

$$\mathcal{L}(f, \mathcal{D}_k) = \sum_{i=1}^{n_k} \|\nabla_{\theta|\theta=1}\, \mathrm{CE}(y_i^k, f(x_i^k) \cdot \theta)\|^2, \tag{3.3}$$

where θ is a "dummy" classifier, whose gradient is not used to update itself but to calculate the penalty. Invariance loss encourages the model to be equally optimal in different training domains, by penalizing the learning of different domain-specific features. However, DG methods in the form of Eq. (3.2) cannot remove the domain-specific features shared by all training domains, leaving the ***spurious invariance***, which is invariant across training domains but variant to the testing domains. The reason is that $\mathcal{L}(f, \mathcal{D}_k)$ is essentially a pooling of domain samples, and in this way, the contribution of some rare samples without the spurious invariance is thus suppressed.

Chapter 4

Methodology

To help DG methods overcome the spurious invariance, we propose DOMAIN+: 1) find the rare samples without spurious invariance by the proposed cross-domain influence, 2) split them from their original domains as a new domain, and then train DG methods on the original domains plus the new one. DOMAIN+ algorithm is summarized in Algorithm 1.

4.1 Algorithm

4.1.1 Cross-Domain Influence

As we discussed in Chapter 1, the sample "rarity" cannot be identified by the dataset-level loss such as Eq. (3.3). To this end, we introduce a sample-level index called ***cross-domain influence*** for sample x from domain k_x:

$$I(x) = \frac{1}{K-1} \sum_{\substack{k=1, \\ k \neq k_x}}^{K} \mathcal{L}_k(f^*) - \frac{1}{K-1} \sum_{\substack{k=1, \\ k \neq k_x}}^{K} \mathcal{L}_k(f^*_{\bar{x}}), \tag{4.1}$$

where $\mathcal{L}_k(f) := \mathcal{L}(f, D_k)$, f^* and $f^*_{\bar{x}}$ denote the optimal model trained on the entire dataset $\mathcal{D}$ and $\mathcal{D} \setminus \{x\}$, respectively. The term "cross-domain" means that the sample removal happens in its own domain but its counterfactual influence is calculated by the

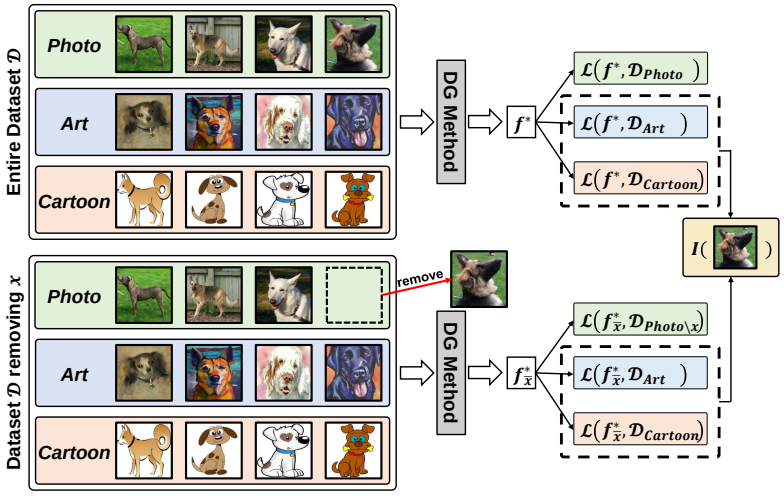


FIGURE 4.1: Illustration of the cross-domain influence in Eq. (4.1).

mean of the invariance changes across other domains. The calculation details are illustrated in Figure 4.1. However, Eq. (4.1) needs to re-train the model on the new dataset $\mathcal{D}\backslash\{x\}$ that is prohibitively expensive.

Thanks to the recent advances in approximating the sample influence without re-training [30, 42], we can implement $I(x)$ by "differentiating" a sample x from domain k_x to derive the gradients of the invariance loss of other domains, *i.e.*, by only training once, we can effectively estimate the influence for each sample:

$$\begin{aligned} I(x) &= \frac{1}{K-1}\sum_{k=1,k\neq k_x}^{K} \frac{\mathrm{d}\mathcal{L}_k(f^*_\epsilon)}{\mathrm{d}\epsilon}\bigg|_{\epsilon=0} \\ &= -\frac{1}{K-1}\sum_{k=1,k\neq k_x}^{K} \nabla\mathcal{L}_k(f^*)H^{-1}_{f^*}\nabla\mathcal{L}(f^*,x), \end{aligned} \tag{4.2}$$

where ϵ denotes an infinitesimal perturbation, $f^*_\epsilon := \arg\min_f \mathcal{L}_{\mathrm{DG}}(f) + \epsilon\mathcal{L}(f,x)$ is the optimal model after perturbing x, where $\mathcal{L}(f,x) = \|\nabla_{\theta|\theta=1}\,\mathrm{CE}(y, f(x)\cdot\theta)\|^2$ if we implement IRM as the invariance loss, the perturbation only happened on $\mathcal{L}(f,x)$ due to we are interested in the changes of the invariance loss but not the whole loss, and $H_{f^*} := \nabla^2\mathcal{L}_{\mathrm{DG}}(f^*)$ denotes the Hessian matrix, which derives from the influence function [30], and more details can be found in Chapter 4.1.2.

Note that the sample-wise gradient of invariance loss reflects the domain changes and it is meaningful for domain-level invariance, which is different from the sample-wise loss

we have discussed. Our cross-domain influence function is calculated on other training domains, which is more reasonable than the original one based on the testing set.

4.1.2 Deriving the Influence Function

Recall DG loss

$$\mathcal{L}_{\mathrm{DG}}(f) = \frac{1}{K}\sum_{k=1}^{K}[\mathcal{L}_{\mathrm{ERM}}(f, \mathcal{D}_k) + \lambda \cdot \mathcal{L}(f, \mathcal{D}_k)],$$

and invariance loss

$$\mathcal{L}(f, \mathcal{D}_k) = \sum_{i=1}^{n_k} \|\nabla_{\theta|\theta=1}\, \mathrm{CE}(y_i^k, f(x_i^k)\cdot\theta)\|^2.$$

Let $f^* := \arg\min_f \mathcal{L}_{\mathrm{DG}}(f)$ and $f_\epsilon^* := \arg\min_f \mathcal{L}_{\mathrm{DG}}(f) + \epsilon\mathcal{L}(f, x)$ denote the optimal model, and we define the parameter change $\Delta_\epsilon := f_\epsilon^* - f^*$. Note that f^* doesn't depend on ϵ, we have

$$\frac{\mathrm{d}\Delta_\epsilon}{\mathrm{d}\epsilon} = \frac{\mathrm{d}f_\epsilon^*}{\mathrm{d}\epsilon}. \tag{4.3}$$

Note f_ϵ^* is the optimal model after perturbing x, we have:

$$0 = \nabla\mathcal{L}_{\mathrm{DG}}(f_\epsilon^*) + \epsilon\nabla\mathcal{L}(f_\epsilon^*, x), \tag{4.4}$$

as the first-order derivative the optimal model should be zero. Then, we perform a Taylor Expansion on the right-hand side:

$$\begin{aligned} 0 \approx &\big[\nabla\mathcal{L}_{\mathrm{DG}}(f^*) + \epsilon\nabla\mathcal{L}(f^*, x)\big] + \\ &\big[\nabla^2\mathcal{L}_{\mathrm{DG}}(f^*) + \epsilon\nabla^2\mathcal{L}(f^*, x)\big]\Delta_\epsilon, \end{aligned} \tag{4.5}$$

where $o(\|\Delta_\epsilon\|)$ term is ignorable. Solving for Δ_ϵ, we have:

$$\begin{aligned} \Delta_\epsilon \approx &-\big[\nabla^2\mathcal{L}_{\mathrm{DG}}(f^*) + \epsilon\nabla^2\mathcal{L}(f^*, x)\big]^{-1} \\ &\big[\nabla\mathcal{L}_{\mathrm{DG}}(f^*) + \epsilon\nabla\mathcal{L}(f^*, x)\big]. \end{aligned} \tag{4.6}$$

Note f^* is the optimal model trained by $\mathcal{L}_{\text{DG}}$, we have $\nabla\mathcal{L}_{\text{DG}}(f^*) = 0$. Keeping only $o(\epsilon)$ terms and let $\epsilon \to 0$, with Eq. (4.3) we have:

$$\left.\frac{\mathrm{d}f_\epsilon^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -H_{f^*}^{-1}\nabla\mathcal{L}(f^*, x), \tag{4.7}$$

where $H_{f^*} := \nabla^2\mathcal{L}_{\text{DG}}(f^*)$ denotes the Hessian matrix.

Finally, we derive $I(x)$ by chain rule

$$\begin{aligned} I(x) &= \frac{1}{K-1}\sum_{k=1,k\neq k_x}^{K}\left.\frac{\mathrm{d}\mathcal{L}_k(f_\epsilon^*)}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \\ &= \frac{1}{K-1}\sum_{k=1,k\neq k_x}^{K}\nabla\mathcal{L}_k(f_\epsilon^*)\left.\frac{\mathrm{d}f_\epsilon^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \\ &= -\frac{1}{K-1}\sum_{k=1,k\neq k_x}^{K}\nabla\mathcal{L}_k(f^*)H_{f^*}^{-1}\nabla\mathcal{L}(f^*, x). \end{aligned}$$

4.1.3 Second-order Stochastic Estimation

Considering the computational challenges in Eq. (4.2), we used the Second-order Stochastic estimation technique for the linear-time approximation based on implicit Hessian-vector products (HVPs) [30, 46]. The idea is to avoid explicitly computing $H_{f^*}^{-1}$ by HVP, as $[\nabla^2\mathcal{L}_{\text{DG}}(f^*)]\,v$ can be computed for arbitrary v in the same time that $\nabla\mathcal{L}_{\text{DG}}(f^*)$ would take, *i.e.*, $O(p)$, where p denote the number of parameters [47].

Specifically, for each domain k, we first compute $s_k := H_{f^*}^{-1}\nabla\mathcal{L}_k(f^*)$ and then update $I(x)$ by $-s_k \cdot \nabla\mathcal{L}(f^*, x)$. Formally, we compute $I(x)$ by following equation:

$$I(x) = -\frac{1}{K-1}\sum_{k=1,k\neq k_x}^{K} s_k \cdot \nabla\mathcal{L}(f^*, x). \tag{4.8}$$

In the following, we introduce the stochastic estimation of s_k. Consider the first j terms in Taylor Expansion of H^{-1}, we have:

$$H_j^{-1} = \sum_{i=0}^{j}(I - H_{f^*})^{-1}. \tag{4.9}$$

Eq. (4.9) can be written in a recursive form:

$$H_j^{-1} = I + (I - H_{f^*})H_{j-1}^{-1}. \tag{4.10}$$

By Taylor Expansion, we have $H_j^{-1} \to H_{f^*}^{-1}$ as $j \to \infty$. From [30], we can use the unbiased estimator of H_{f^*} to form $\tilde{H}_j^{-1}$, so that $\mathbb{E}[\tilde{H}_j^{-1}] = H_j^{-1}$, *i.e.*, $\mathbb{E}[\tilde{H}_j^{-1}] \to H_{f^*}$. Particularly, we can uniformly sample x_k from domain k and use $\nabla^2\mathcal{L}_{\text{DG}}(f^*, x_k)$ as an unbiased estimator of H_{f^*}.

Formally, consider uniformally sample t points $x_{k_1}, x_{k_2}, \ldots, x_{k_t}$ from domain k, and let $\tilde{H}_0^{-1}\nabla\mathcal{L}_k(f^*) := \nabla\mathcal{L}_k(f^*)$, we can compute $\tilde{H}_j^{-1}$ recursively:

$$\begin{aligned}\tilde{H}_j^{-1}\nabla\mathcal{L}_k(f^*) =& \nabla\mathcal{L}_k(f^*) \\ &+(I - \nabla^2\mathcal{L}_{\text{DG}}(f^*, x_k))\tilde{H}_{j-1}^{-1}\nabla\mathcal{L}_k(f^*).\end{aligned} \tag{4.11}$$

Thus, we obtain our final estimate of $H_{f^*}^{-1}\nabla\mathcal{L}_k(f^*)$ as $\tilde{H}_t^{-1}\nabla\mathcal{L}_k(f^*)$ by Eq. (4.11). We pick t to be large enough such that $\tilde{H}_t^{-1}$ converges and we repeat the procedure r times and average the results. In this work, we empirically choose $r = 3$ and $t = 1000$ on all datasets, hence the overall time complexity is $O(rt)$, which is the same order as $O(n)$.

4.1.4 Rare samples split into a new domain

After estimating the cross-domain influence of each sample x, we split the rare samples by $I(x) > \alpha$ from their original domain, and construct a new domain $\mathcal{D}^+$, where α is a threshold. Then, we train DG methods on $\{\mathcal{D}_k \backslash \mathcal{D}^+\}_{k=1}^K \cup \{\mathcal{D}^+\}$ to achieve the invariance. We'd like to highlight again that the influence is fundamentally different from the sample hardness in hard example mining [23, 48]. Besides the qualitative samples in Figure 1.2, we also show the feature distributions of all the samples of different classes in Figure 5.1. Interestingly, we can see the difference of $\mathcal{D}^+$ selected by influence and training loss: as the rare samples with large influence are usually confounded by the majority, they are more evenly distributed than the "hard" samples, which are merely considered as the eccentric points far from the mainstream.

Algorithm 1: DOMAIN+

Input : Dataset $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^K$, Threshold α
Output: New Dataset $\{\mathcal{D}_k \setminus \mathcal{D}^+\}_{k=1}^K \cup \{\mathcal{D}^+\}$
Train f on $\mathcal{D}$ by Eq. (3.2) and derive the optimal f^*;
Initialize: $\mathcal{D}^+ \leftarrow \varnothing$;
foreach $\mathcal{D}_k \in \mathcal{D}$ **do** // Enumerate domains
  Initialize: $\mathcal{D}_r \leftarrow \varnothing$; // Rare sample set
  foreach $x \in \mathcal{D}_k$ **do**
    Initialize: $I(x) \leftarrow 0$;
    foreach $\mathcal{D}_j \in \mathcal{D} \setminus \{\mathcal{D}_k\}$ **do**
      $I(x) \leftarrow I(x) - \nabla\mathcal{L}_j(f^*) H_{f^*}^{-1} \nabla\mathcal{L}(f^*, x)$;
      // Eq. (4.2)
    $I(x) \leftarrow I(x)/(K-1)$
    if $I(x) > \alpha$ **then**
      $\mathcal{D}_r \leftarrow \mathcal{D}_r \cup \{x\}$; // x is rare
  $\mathcal{D}^+ \leftarrow \mathcal{D}^+ \cup \mathcal{D}_r$; // Update DOMAIN+
Apply any DG methods on $\{\mathcal{D}_k \setminus \mathcal{D}^+\}_{k=1}^K \cup \{\mathcal{D}^+\}$.

4.2 Justification

We provide a formal definition of rare samples according to the recent advance in influence function:

Definition 1. *Given a pooled multi-domain dataset $\mathcal{D}$. A set $\mathcal{D}^+$ is said to contain all rare samples in $\mathcal{D}$ if and only if:*

$$\mathcal{D}^+ := \{x \in \mathcal{D} | I(x) > 0\},$$

where $I(\cdot)$ indicates the cross-domain influence function.

In practice, we follow recent influence-based methods [30, 45] and select $\mathcal{D}^+ = \{x | I(x) > \alpha\}$, where α is a threshold slightly greater than 0 to tolerate the estimation error. We find that computing the analytical solution for α is not necessary. As shown in Figure 4.2, influence experiences a sharp transition at some threshold, allowing us to simply tune the value of α.

Therefore, we can split rare samples into a new domain by samples' influence according to Definition 1, and with the new domain, we have the following proposition.

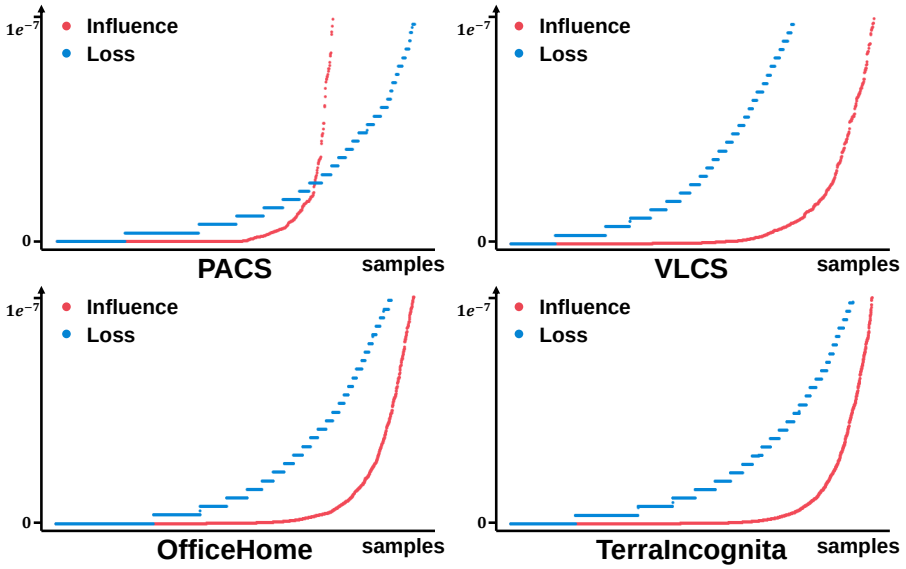


FIGURE 4.2: Visualization of sorted cross-domain influence (red) and training loss (blue) of training samples using IRM. We train the model on the default three training domains for each dataset. Each dot denotes a sample and its influence/loss value.

Proposition 1. $\mathcal{D}^+$ as a new training domain can reduce the degree of freedom in the invariant solution space w.r.t. the learned invariant features.

Here, the degree of freedom (DOF) indicates the dimension of image space. The reduction of DOF is equivalent to project spurious invariant features to kernel space, which means the model has a better generalization ability [15]. Hence, Proposition 1 says that DOMAIN+ will help DG methods achieve better invariance. We provide a theoretical proof in Chapter 4.3.

To demonstrate why traditional hard-sample mining methods *fail to select rare samples* with spurious features, we have the following proposition.

Proposition 2. Let a set of hard samples $\mathcal{D}_\ell$ satisfies:

$$\mathcal{D}_\ell := \{x | \ell(x, y) > \alpha\},$$

then $\mathcal{D}_\ell$ and $\mathcal{D}^+$ are comprised of different samples.

The proof is in Chapter 4.3. This proposition can be evidenced by our CMNIST experiment, as shown in Figure 5.2, where the selection by loss involves many dominant samples (*i.e.*, red images). As such, we resort to our proposed cross-domain influence

function for rare sample selection, which correctly selects the rare samples (*i.e.*, green images).

4.3 Theoretical Proof

Proposition 1. $\mathcal{D}^+$ as a new training domain can reduce the degree of freedom in the invariant solution space w.r.t. the learned invariant features.

Proof. Here we consider a linear projection $f : \mathcal{X} \to \mathcal{Y}$ as an invariant predictor, where $\mathcal{X}$ denotes the feature space and $\mathcal{Y}$ denotes the one-hot label space. Since we only require the classifier to be invariant to the domain, we slightly abuse the concept of f for brevity and ignore the nonlinear feature extractor. From *Rank–nullity Theorem*, we have

$$dim(\ker(f)) = dim(\mathcal{X}) - dim(\mathrm{im}(f)), \tag{4.12}$$

where $\mathrm{im}(f)$ and $\ker(f)$ denote the image space and kernel space of f, respectively. Let x_c be a specific feature, *s.t.*, $f(x_c) = y$. Thus, for all $x_0 \in \ker(f)$, we have

$$f(x_c + x_0) = f(x_c) + f(x_0) = y + \mathbf{0} = y. \tag{4.13}$$

From Eq. (4.13), we know that x_c is the discriminative feature and x_0 is the non-discriminative one. Therefore, we intuitively expect every spurious feature x_s would belong to an invariant solution space $\ker(f)$, *i.e.*, achieve true invariance.

However, in practice, some spurious invariance features x_s are dominant in training domains, and result in learning an optimal model f^*, *s.t.*, $f^*(x_s + x_c) = y$, but $f^*(x_c) \neq y$. Thanks to IRM Theorem 9 [15], which proves that adding a *new domain* will remove one degree of freedom in invariant solution space, *i.e.*, reducing the $dim(\mathrm{im}(f))$. As the dimension of feature space $\mathcal{X}$ is fixed, $dim(\ker(f))$ increases, which means more spurious invariant feature x_s is more likely to be constrained in $\ker(f)$.

In the following, we prove that our split new domain $\mathcal{D}^+$ is a *new domain* by proving it lies in *linear general position* of original domains according to Eq. (4.14).

$$\dim\left(\mathrm{span}\left(\left\{\mathbb{E}_{X_i}\left[X_i X_i^\top\right] x - \mathbb{E}_{X_i,\epsilon_c}\left[X_i \epsilon_i\right]\right\}_{c\in\{d,r\}}\right)\right) = 2, \tag{4.14}$$

where X_d, X_r denote the feature distribution of dominant features and rare features, respectively, and Eq. (4.14) always hold because of Definition 1. Moreover, as all original domains lie in *linear general position*, Eq. (4.14) can further prove that domains containing only dominant features (*i.e.*, original domains minus $\mathcal{D}^+$) lie in *linear general position* with the new domain with only rare features (*i.e.*, $\mathcal{D}^+$), which means $\mathcal{D}^+$ is a valid *new domain*. With Eq. (4.14) and IRM Theorem 9 [15], Proposition 1 yields the proof. □

Proposition 2. Let a set of hard samples $\mathcal{D}_\ell$ satisfies:

$$\mathcal{D}_\ell := \{x | \ell(x, y) > \alpha\},$$

then $\mathcal{D}_\ell$ and $\mathcal{D}^+$ are comprised of different samples.

Proof. We prove the above proposition by a case study. Consider the following binary classification problem with spurious feature:

$$\begin{aligned}
&\mathbf{y} \in \{-1, 1\} \\
&\mathbf{z} \sim \begin{cases} P(\mathbf{z} = \mathbf{y} | \mathbf{y}) = \rho \\ P(\mathbf{z} = -\mathbf{y} | \mathbf{y}) = 1 - \rho \end{cases} \\
&\epsilon \sim \mathcal{N}(0, \sigma^2) \\
&\mathbf{x} = [\mathbf{y} + \epsilon, \mathbf{z}],
\end{aligned}$$

where $\mathbf{y}$ is the ground-truth label, $\mathbf{z}$ is a spurious feature that correlated with label with probability ρ, and $\mathbf{x}$ is the sample feature. The target is to train a predictor which can predict the label $\mathbf{y}$ from the given sample $\mathbf{x}$. This case can be considered as a simple version of Colored MNIST, where $\mathbf{y}$ denotes the number label, and $\mathbf{z}$ denotes the color label. Consider a simple regression function

$$f(\mathbf{x}) = \mathbf{x} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \tag{4.15}$$

where β_1 and β_2 are trainable parameters with a log-loss objective

$$\ell(f(\mathbf{x}), \mathbf{y}) = \log(1 + \exp(-\mathbf{y} f(\mathbf{x}))). \tag{4.16}$$

Particularly, when $\beta_1 = 1$ and $\beta_2 = 0$, it is an invariant predictor. However, as shown in IRM [15], such simple log-loss methods cannot eliminate spurious correlations, which result in non-zero β_2.

In the following, we showcase that samples with larger loss has no implication that it contains no spurious features. Let $\mathbf{y} = 1$, and consider two samples:

$$\mathbf{x}_1 = [1 + \epsilon_1, 1]$$
$$\mathbf{x}_2 = [1 + \epsilon_2, -1],$$

where $\mathbf{x}_1$ contains spurious feature while $\mathbf{x}_2$ not.

We will show that $\mathbf{x}_2$ is not always a 'harder' sample compared to $\mathbf{x}_1$ w.r.t. objective function in Eq. (4.16). Let $\ell(f(\mathbf{x}_1), \mathbf{y}) > \ell(f(\mathbf{x}_2), \mathbf{y})$, we derive:

$$\epsilon_1 < \epsilon_2 - \frac{2\beta_2}{\beta_1}. \tag{4.17}$$

This is easy to hold in practice, since both ϵ_1 and ϵ_2 are random variables with large variance σ^2. Therefore, if this condition holds, $\mathbf{x}_1$ will be treated as a "hard sample" instead of $\mathbf{x}_2$. However, $\mathbf{x}_2$ is the rare sample without spurious feature. So Proposition 2 yields proof. □

Chapter 5

Experiments

5.1 Settings

5.1.1 Dataset

Following DomainBed, we demonstrated our Domain+ on 4 popular multi-domain image classification datasets. The examples of the datasets are shown in Table 5.1 and Table 5.2.

1) **PACS** [49] contains $9,991$ images of 7 classes from four domains, including `art`, `cartoons`, `photos`, and `sketches`.

2) **VLCS** [50] contains $10,729$ photographs of 5 classes from four domains, including `Caltech101`, `LabelMe`, `SUN09`, and `VOC2007`.

3) **Office-Home** [51] contains $15,588$ images of 65 classes from four domains, including `art`, `clipart`, `product` and `real`.

4) **Terra Incognita** [52] contains $24,788$ photos of wild animals from 10 classes, taken at four different locations, including `L100`, `L38`, `L43` and `L46`.

5.1.2 DomainBed Benchmark

It is a stringent and reproducible testbed for domain generalization that provides consistent implementations across SOTA methods for fair comparisons [3]. We followed

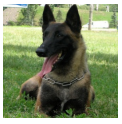
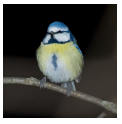
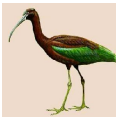
PACS				VLCS			
A	C	P	S	C	L	S	V
							
							

TABLE 5.1: Examples from PACS and VLCS.

the training-domain validation set in DOMAINBED for model selection by default, and we additionally applied the training-domain validation (oracle) setting. Specifically, we split each training domain into training and validation subsets, which account for 80% and 20%, respectively. We choose the model maximizing the average accuracy on the validation sets. For a fair comparison, we follow the settings in Fish [16] that report the average over 5 random trials. To evaluate the effectiveness of DOMAIN+, we follow DOMAINBED and use accuracy as our main metric. Accuracy measures the proportion of correct predictions out of all predictions made, offering a clear indication of our model's performance in accurately generalizing across various domains.

5.1.3 Baselines

We chose 3 popular DG SOTAs: **IRM** [15], **CORAL** [17], and **Fish** [16], and applied our DOMAIN+ to them, where we later named **IRM+**, **CORAL+**, and **Fish+**, respectively. We compared their performances with other SOTAs based on the implementation of DOMAINBED, including ERM [53], DRO [36], Mixup [54], MLDG [33], MMD [22], RSC [32], ANDMask [38] and SagNet [55].

OfficeHome				TerraIncognita			
A	**C**	**P**	**R**	**L100**	**L38**	**L43**	**L46**
							
							

TABLE 5.2: Examples from OfficeHome and TerraIncognita.

Algorithm	PACS					VLCS				
	A	**C**	**P**	**S**	**Avg**	**C**	**L**	**S**	**V**	**Avg**
ERM [53]	**86.7**±1.3	79.6±2.7	95.8±0.6	79.2±2.6	85.3±1.3	97.1±1.0	**65.7**±1.5	69.7±2.9	74.3±3.6	76.7±1.2
DRO [36]	83.6±2.2	79.7±2.3	96.5±0.4	78.9±2.3	84.7±1.3	96.9±1.2	63.3±1.1	70.0±2.5	72.9±2.7	75.8±1.4
Mixup [54]	85.3±1.1	80.5±1.2	96.9±0.3	75.9±2.9	84.6±1.1	97.6±0.7	63.2±1.5	70.6±1.6	74.9±1.4	76.6±0.9
MLDG [33]	83.0±4.9	76.2±1.8	95.8±1.1	74.5±2.0	82.4±1.4	97.2±0.9	63.2±2.2	70.1±2.1	72.5±1.6	75.7±1.1
MMD [22]	83.4±2.1	79.4±3.7	95.7±0.7	74.0±7.0	83.1±2.3	97.4±0.9	62.9±2.0	69.9±1.8	74.8±3.1	76.2±1.5
RSC [32]	80.6±2.9	77.5±3.4	95.1±0.6	76.9±2.7	82.5±1.4	93.7±1.8	64.2±1.8	67.8±1.4	71.1±3.5	74.2±1.0
ANDMask [38]	84.3±3.1	77.6±1.9	96.3±0.7	72.7±4.4	82.7±2.3	96.7±1.4	63.9±2.1	67.1±3.3	70.4±3.1	74.5±1.7
SagNet [55]	83.2±0.6	81.1±1.2	95.5±1.2	77.9±2.2	84.4±0.8	96.1±1.3	63.3±2.3	72.3±3.4	73.7±2.7	76.3±0.9
IRM [15]	85.7±2.1	79.8±1.6	95.8±0.4	78.0±1.7	84.8±0.4	94.7±3.1	64.7±1.5	70.2±1.3	73.8±3.7	75.9±0.8
IRM+	85.9±2.8	81.1±0.3	96.7±0.7	78.7±1.4	85.6±0.7	97.2±0.8	65.5±1.7	71.3±2.0	75.9±1.2	77.5±0.8
CORAL [17]	84.2±2.4	78.8±3.1	96.6±0.6	77.5±1.3	84.3±0.8	97.1±0.5	65.5±1.2	70.3±2.5	**76.8**±2.2	77.4±0.8
CORAL+	86.5±2.0	**81.3**±2.3	**97.0**±0.6	**80.8**±0.8	**86.4**±0.7	**98.3**±0.6	65.3±1.6	71.6±1.9	76.5±2.2	77.9±0.7
Fish [16]	85.3±1.8	79.0±1.3	95.9±1.0	78.3±2.8	84.6±1.2	97.5±1.1	64.2±1.6	71.2±0.7	75.4±1.4	77.1±0.6
Fish+	86.1±1.8	81.1±2.0	96.9±0.8	78.7±3.4	85.7±1.0	98.0±0.9	65.3±1.2	**73.0**±1.3	76.4±1.2	**78.2**±0.6

TABLE 5.3: Test accuracy (%) of PACS and VLCS based on training-domain validation hyper-parameter tuning. "+" mark indicates that it trained by our DOMAIN+.

Algorithm	PACS					VLCS				
	A	**C**	**P**	**S**	**Avg**	**C**	**L**	**S**	**V**	**Avg**
ERM	87.2±1.6	83.1±2.0	97.6±0.5	81.7±0.8	87.4±0.3	98.1±0.4	66.6±1.0	73.2±1.9	77.8±1.2	78.9±0.9
DRO	86.7±1.3	81.5±1.4	97.1±0.4	83.0±1.2	87.1±0.4	98.0±0.6	67.0±1.1	72.7±1.4	76.6±2.8	78.6±0.8
Mixup	85.6±0.7	80.7±1.6	97.3±0.2	80.6±1.3	86.1±0.4	98.5±0.3	66.8±1.3	73.7±1.6	78.0±1.5	79.3±0.7
MLDG	86.0±1.0	82.9±1.6	96.7±0.4	80.4±1.4	86.5±0.6	98.3±0.9	66.6±1.0	72.1±2.5	77.1±2.2	78.5±0.9
MMD	86.3±0.8	82.8±1.0	97.4±0.5	80.8±1.7	86.8±0.6	98.6±0.2	67.7±0.6	72.6±0.7	77.2±1.5	79.0±0.5
ANDMask	86.8±1.2	79.1±1.5	97.2±0.4	78.0±1.7	85.3±0.5	97.7±0.9	64.2±4.1	69.2±3.1	74.9±1.4	76.5±1.2
SagNet	85.5±1.4	82.6±1.5	96.4±0.4	80.3±1.2	86.2±0.4	96.8±0.9	67.5±0.7	74.2±2.1	78.9±1.4	79.3±0.7
IRM	87.0±2.1	82.0±1.2	97.1±0.5	81.1±2.3	86.8±1.1	98.3±0.2	66.5±1.6	72.8±2.5	78.6±1.6	79.1±0.9
IRM+	87.9±1.2	82.8±1.3	97.7±0.4	81.9±1.1	87.6±0.4	98.5±0.7	67.7±0.4	74.2±1.2	79.1±0.2	79.9±0.5
CORAL	87.6±1.0	82.2±0.7	97.2±0.6	81.1±0.8	87.0±0.4	97.7±0.4	66.1±1.1	73.2±1.4	78.8±1.3	79.0±0.2
CORAL+	89.0±1.1	83.5±1.6	97.6±0.1	82.4±1.5	88.1±0.9	98.4±0.6	67.7±1.1	73.9±1.6	78.2±1.9	79.6±0.7
Fish	88.2±1.3	82.4±2.0	97.0±0.4	79.8±1.6	86.8±0.9	98.0±0.6	66.4±1.4	72.7±1.8	77.8±1.9	78.7±0.8
Fish+	87.9±0.3	83.4±1.5	98.0±0.2	82.4±0.9	87.9±0.4	98.5±0.2	66.5±0.6	74.8±2.5	78.8±2.1	79.6±0.7

TABLE 5.4: Test accuracy (%) of PACS and VLCS based on testing-domain validation hyper-parameter tuning. "+" mark indicates that it trained by our DOMAIN+.

Algorithm	OfficeHome					TerraIncognita				
	A	C	P	R	Avg	L100	L38	L43	L46	Avg
ERM [53]	59.5±1.8	52.3±1.3	73.9±1.3	75.6±1.0	65.3±0.4	49.7±2.9	43.3±1.6	56.1±2.1	35.3±2.8	46.1±1.8
DRO [36]	58.8±1.4	52.9±1.5	74.5±1.1	75.5±0.7	65.4±0.7	48.9±3.4	41.1±3.9	**57.5**±0.5	37.3±1.9	46.2±1.0
Mixup [54]	61.4±1.4	53.9±1.7	76.1±0.7	77.1±0.6	67.1±0.5	54.7±4.5	44.1±3.8	55.1±3.2	31.1±3.7	46.2±2.7
MLDG [33]	57.2±0.6	51.4±1.9	73.1±1.0	74.8±0.9	64.1±0.4	50.4±3.8	37.5±4.0	52.6±3.9	34.1±3.6	43.6±1.9
MMD [22]	58.4±1.3	53.4±0.7	73.9±0.8	75.2±0.6	65.2±0.6	48.8±3.2	40.9±3.1	54.2±1.8	36.6±4.1	45.1±2.1
ANDMask [38]	55.8±1.3	50.5±1.6	73.2±0.7	75.0±0.7	63.6±0.4	44.6±3.8	40.5±1.3	53.6±2.1	37.0±3.0	43.9±1.5
SagNet [55]	59.1±1.6	52.4±2.5	74.6±0.9	75.3±0.8	65.3±0.4	50.6±3.6	44.0±2.3	54.8±1.3	31.4±3.9	45.2±2.7
IRM [15]	58.5±1.0	52.0±1.3	73.5±1.5	74.8±0.9	64.7±0.7	53.5±4.4	41.8±3.6	55.6±1.7	37.7±4.2	47.2±1.4
IRM+	60.5±1.2	52.9±1.4	75.2±1.2	76.2±0.4	66.2±0.4	54.8±4.2	**45.4**±3.3	56.3±2.0	38.1±0.9	**48.6**±1.3
CORAL [17]	63.0±1.2	55.3±0.9	76.0±0.5	76.8±0.9	67.8±0.4	51.9±2.0	41.1±2.8	52.4±3.4	37.3±2.8	45.7±2.0
CORAL+	**63.3**±0.9	**56.4**±0.9	**76.6**±1.1	**78.4**±0.2	**68.7**±0.4	**55.5**±2.4	44.1±3.3	56.0±3.1	37.8±3.0	48.4±2.3
Fish [16]	59.0±1.4	52.4±1.8	73.7±0.7	74.5±0.6	64.9±0.8	49.9±2.0	41.7±1.7	54.5±1.6	37.9±3.5	46.0±0.9
Fish+	59.9±1.5	53.0±1.4	75.0±0.7	75.9±1.1	65.9±0.9	53.6±2.2	44.7±2.2	56.2±1.2	**38.2**±3.5	48.2±1.9

TABLE 5.5: Test accuracy (%) of OfficeHome and TerraIncognita based on training-domain validation hyper-parameter tuning. "+" mark indicates that it trained by our DOMAIN+.

Algorithm	OfficeHome					TerraIncognita				
	A	C	P	R	Avg	L100	L38	L43	L46	Avg
ERM	60.0±1.1	52.6±1.6	74.4±1.3	75.6±0.9	65.6±0.3	62.6±3.7	47.8±1.9	57.7±1.1	42.6±2.5	52.7±1.2
DRO	59.3±1.5	52.7±1.5	75.0±0.8	75.8±0.6	65.7±0.8	59.5±3.1	46.8±3.3	60.2±1.1	43.1±1.5	52.4±1.6
Mixup	61.3±1.6	54.9±0.7	75.7±0.6	77.4±0.3	67.3±0.3	67.7±1.0	49.1±1.6	58.8±1.0	39.8±1.3	53.9±0.4
MLDG	57.8±0.8	52.7±0.9	73.5±0.6	75.5±0.9	64.9±0.5	58.0±3.0	52.7±3.4	58.8±2.1	41.0±0.7	52.6±1.1
MMD	59.0±1.1	52.8±0.9	73.4±1.0	75.7±1.0	65.2±0.6	60.2±2.6	45.7±1.2	57.8±1.3	43.8±0.6	51.9±1.3
ANDMask	57.5±1.4	52.6±0.6	73.1±0.9	75.4±1.0	64.6±0.4	58.6±6.9	46.9±2.9	55.2±2.0	44.0±1.0	51.2±2.1
SagNet	60.0±1.2	53.1±1.7	74.7±1.0	75.4±1.4	65.8±0.5	61.8±1.8	49.1±2.7	57.2±0.8	41.0±1.5	52.3±1.3
IRM	59.3±1.6	53.2±1.1	74.4±1.3	75.4±0.8	65.6±0.5	62.4±2.9	46.8±2.4	57.9±1.1	43.8±0.7	52.7±1.4
IRM+	60.6±0.7	53.8±1.4	75.5±1.0	76.0±0.6	66.5±0.5	62.9±2.9	49.0±2.6	59.6±1.8	43.6±1.8	53.8±1.9
CORAL	62.5±1.1	55.5±1.5	76.4±1.0	78.1±0.5	68.1±0.5	58.5±2.0	49.6±3.0	57.4±1.7	44.5±1.2	52.5±1.1
CORAL+	64.3±1.0	56.5±1.2	76.9±0.6	78.6±0.7	69.1±0.4	63.6±0.8	49.6±1.5	59.0±0.6	44.1±1.8	54.1±0.6
Fish	60.2±1.3	53.1±1.3	74.5±0.5	75.1±0.5	65.7±0.5	59.5±2.9	48.1±2.0	57.7±1.7	42.3±3.0	51.9±0.9
Fish+	60.2±1.4	54.2±1.7	75.2±1.0	76.1±1.2	66.4±0.7	61.8±1.7	48.7±0.6	59.0±2.1	43.7±2.5	53.3±0.8

TABLE 5.6: Test accuracy (%) of OfficeHome and TerraIncognita based on testing-domain validation hyper-parameter tuning. "+" mark indicates that it trained by our DOMAIN+.

5.2 Implementation Details

5.2.1 Efficient Influence Calculation

Considering the computational challenges in Eq. (4.2), we used the Second-order Stochastic estimation technique for the liner-time approximation based on implicit Hessian-vector products (HVPs) [30, 46], and the procedure is detailed in the Chapter 4.1.3. In particular, we run it three times, $1,000$ steps each, and average the results as the influence in Eq. (4.2).

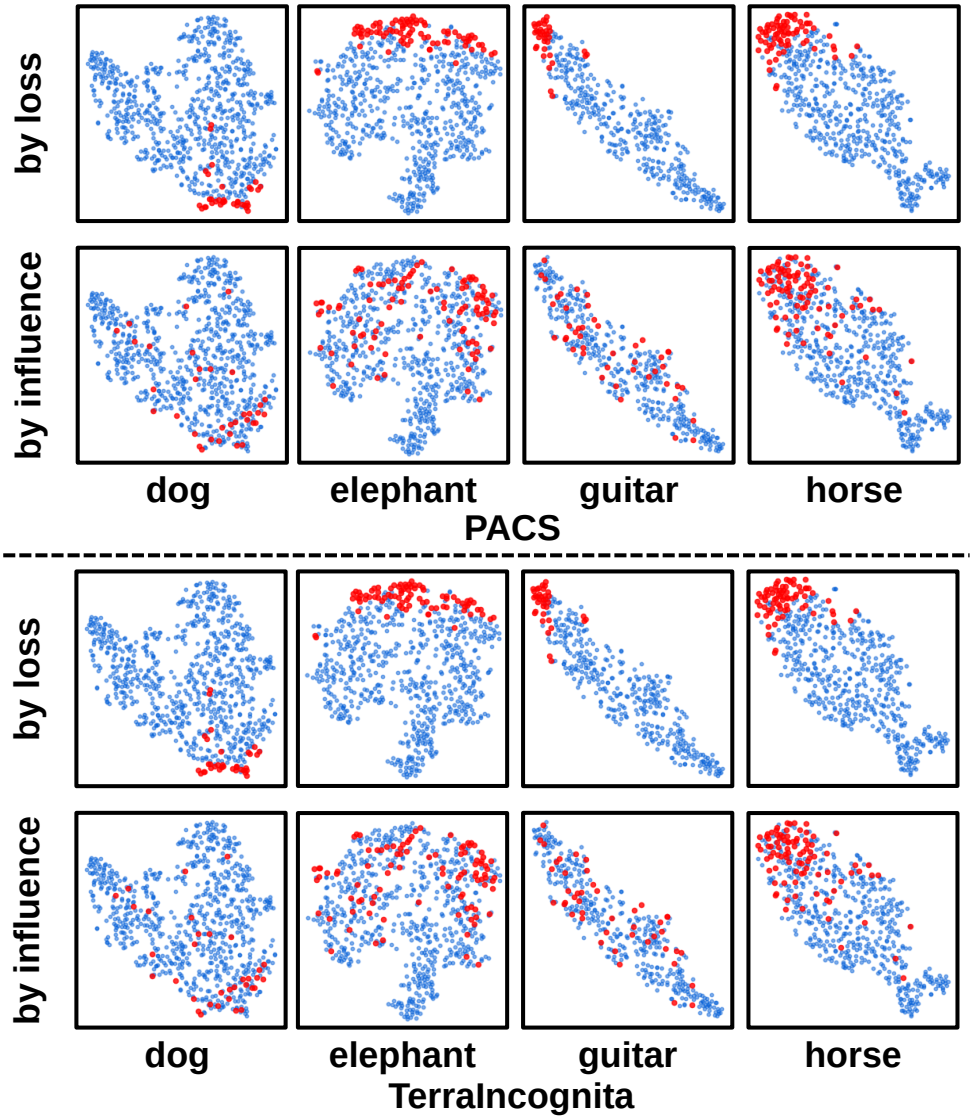


FIGURE 5.1: t-SNE [1] visualization of the training sample features extracted by IRM model. We trained the model on the default three domains on each dataset. Red dots are the selected rare samples by training loss (Top) and cross-domain influence (Bottom).

5.2.2 Parameter Settings

Following the settings in DOMAINBED, we used pre-trained ResNet-50 [56] as the backbone for all methods on all datasets and optimize all models using Adam [57]. We applied the default settings and hyperparameters in DOMAINBED, specifically, we applied Adam [57] optimizer with learning rate as 5×10^{-5}, we set batch size as 32 and training step as 5000 for all experiments. For our hyperparameter α, we search the best value from $\{10^{-12}$,10^{-10},10^{-8},10^{-6},10^{-4},10^{-2},$10^{-1}\}$, specifically, we set it as $\{10^{-12}$, 10^{-8},

10^{-10}, 10^{-6}} for A, C, P, S in PACS, respectively; {10^{-8}, 10^{-4}, 10^{-8}, 10^{-10}} for C, L, S, V in VLCS, respectively.

5.3 Results and Analysis

We show the effectiveness of our DOMAIN+ by the following Q&A.

Q1. *How does* DOMAIN+ *improve DG methods?*

A1. Our main results are shown in Table 5.3, 5.4, 5.5 and 5.6. Compared to the original DG methods, our DOMAIN+ consistently improves most of the settings by a large margin. Specifically, we improve IRM, CORAL and Fish by 1.3%, 1.6%, and 1.3% averaged over the four datasets, respectively. In particular, we have made the largest improvement on the TerraIncognita dataset, where the baselines perform worse than other datasets. One possible reason is that this dataset may contain more spurious invariance (as its only domain variance is the camera's latitude, there could be more shared features across domains than PACS), which downplays the baselines, and thus our DOMAIN+ for removing the spurious invariance plays a more essential role in improvement.

Q2. *How does* DOMAIN+ *perform compared to SOTAs?*

A2. Compared to the original SOTAs, especially the ERM implemented by DOMAINBED [3], the DG methods equipped with our DOMAIN+ achieve new SOTAs in each setting of each dataset. Specifically, we improve the SOTAs by 1.1%, 0.8%, 0.9%, and 1.4% averaged performance for the four datasets, respectively. Noteworthy, some original DG methods cannot even beat ERM. However, after applying our DOMAIN+, all of them outperform ERM in most cases. This demonstrates that by removing the spurious invariance, we effectively promote the potential invariance of these DG methods.

To further show the effect of DOMAIN+ in feature learning, in Figure 5.3, we visualized the extracted features of IRM (Top) and IRM+ (Bottom). At the top, we find that some samples belonging to the same class (dots with the same color) are not perfectly clustered together, which caused incorrect predictions. However, at the bottom, the confusion of feature clusters is much less, indicating our DOMAIN+ helps IRM learn better domain-invariant features.

Q3. *Why only influence can select rare samples?*

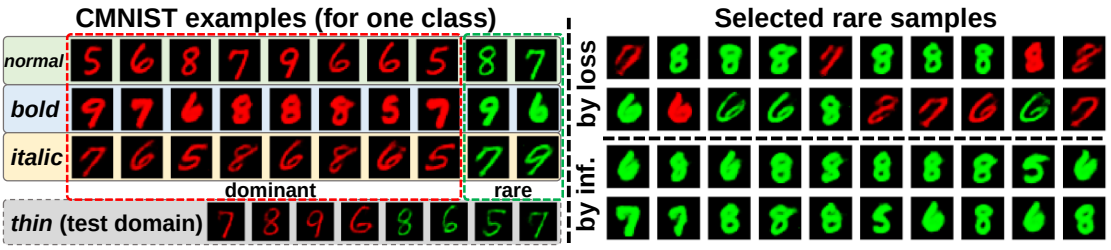


FIGURE 5.2: Visualizations of our synthesized CMNIST dataset (Left) and selected samples with the highest loss/influence (Right).

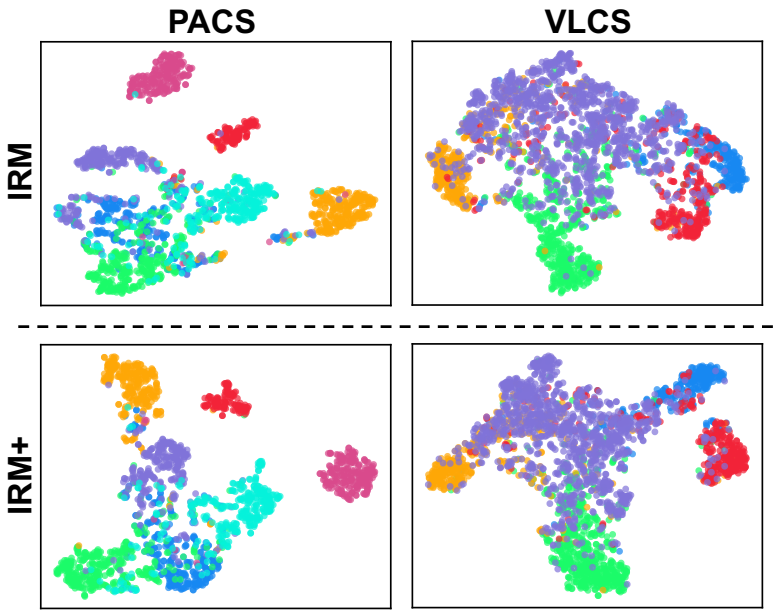


FIGURE 5.3: t-SNE [1] visualization of the features of test samples extracted by IRM and IRM+ (IRM with our DOMAIN+). We trained the model on the default three domains on each dataset. Different colors denote different classes.

A3. Table 5.7 shows that the new domain selected by influence helps DG methods achieve better performance, *i.e.*, better invariance, compared to the one selected by the training loss. The reasons are two-fold. First, influence is a better measure of "rarity". As shown in Figure 4.2, most of the samples have influence values similar to 0, indicating that they are the dominant samples, while the dominant counterpart by training loss is hard to be identified as its value curve is not as sharp as the influence. Second, the training loss only focuses on hard samples that do not fit the model well, but do not necessarily have spurious invariance. As shown in Figure 5.1, the selected samples by training loss are far from the sample center, which means that its selection only focuses on eccentric training samples, *e.g.*, noisy samples. However, the selection of influence is

Algorithm	A	C	P	S	Avg
IRM	85.7±2.1	79.8±1.6	95.8±0.4	78.0±1.7	84.8±0.4
IRM+ (Random)	84.3±1.4	79.8±1.5	96.2±0.9	74.4±6.0	83.6±1.8
IRM+ (Loss)	85.1±3.0	78.6±1.6	96.3±0.6	75.5±2.7	83.9±1.5
IRM+ (Cluster)	84.6±0.3	80.1±1.6	95.5±1.6	77.9±1.9	84.5±1.4
IRM+ (Ours)	85.9±2.8	81.1±0.3	96.7±0.7	78.7±1.4	85.6±0.7

TABLE 5.7: Ablations on influence. "Random", "Loss", "Cluster" and "Ours" denote different sample selection methods.

Algorithm	A	C	P	S	Avg
IRM	85.7±2.1	79.8±1.6	95.8±0.4	78.0±1.7	84.8±0.4
IRM+	85.9±2.8	81.1±0.3	96.7±0.7	78.7±1.4	85.6±0.7
IRM w/o $\mathcal{D}^+$	79.3±3.1	70.8±2.0	92.3±1.4	74.4±1.7	79.2±0.8
IRM++	84.7±1.9	78.4±2.3	96.7±0.5	74.4±1.8	83.6±0.4
IRM-zero	84.9±2.9	80.9±1.3	95.5±0.3	78.4±2.5	85.0±1.3

TABLE 5.8: Experimental results on further exploration of DOMAIN+, where IRM++ denotes re-training IRM with DOMAIN++, IRM w/o $\mathcal{D}^+$ denotes re-training IRM without $\mathcal{D}^+$, and IRM-zero denotes no domain label is provided.

more scattered, which means that the rare samples are indeed confounded by the majority distribution—spurious correlation (invariance) is identified. It is worth noting that, as shown in Table 5.7, our method also outperforms the naive data splitting by feature clusters [58], which demonstrates clustering method also fails to split the rare samples.

Q4. *What is the difference of selected samples between loss and influence?*

A4. As there is no dataset with the *spurious invariance* ground-truth, we generate a specific Colored MNIST dataset (CMNIST), and the details are illustrated in Fig. 5.2. We first set 2 classes where (digit ≥ 5) is one class and (digit < 5) is another class. Then, we define 3 training domains (*normal*, *bold*, *italic*) and 1 test domain (*thin*). We use two colors (red and green) to set the same dominants (*e.g.*, most images are red for digits ≥ 5) for each training domain, where the dominant ratio is 70%, 80%, and 90% for *normal*, *bold*, and *italic*, respectively, and color distribution is balanced in the test domain. In our CMNIST, the *spurious invariance* is color, *e.g.*, red dominates digits

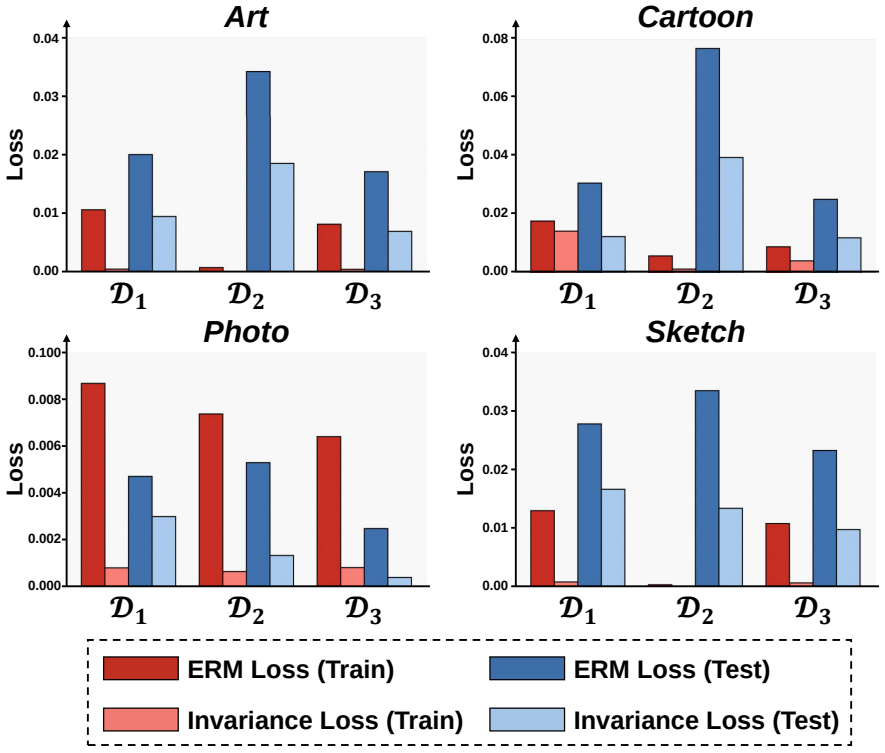


FIGURE 5.4: The training/testing ERM/Invariance loss for IRM on PACS with different setups of training domains, where $\mathcal{D}_1$ denotes the original training dataset $\mathcal{D}$, $\mathcal{D}_2$ denotes $\{\mathcal{D}_k \setminus \mathcal{D}^+\}_{k=1}^K$, and $\mathcal{D}_3$ denotes $\{\mathcal{D}_k \setminus \mathcal{D}^+\}_{k=1}^K \cup \{\mathcal{D}^+\}$, *i.e.* our DOMAIN+.

≥ 5 for all training domains, so green samples are rare. Our cross-domain influence correctly selects the rare samples (*i.e.*, green images), while the selection by loss involves many dominant samples (*i.e.*, red images). This enables our method to better remove the spurious invariance. Thus quantitatively, our IRM+ (73.4%) is better than IRM (71.0%) and IRM+ (Loss) (70.6%).

Q5. *How about just training without the rare samples?*

A5. Unlike the conventional influence-based methods [31, 45], which treat rare samples as "harmful" and simply discard them from training, removing the rare samples will cause the DG methods to learn more spurious invariance and thus degrade the performance, as the results shown in Table 5.8. We conduct more experiments to demonstrate the need for constructing the new domain. As illustrated in Figure 5.4, we observe a decrease in invariance loss after removing rare samples, but an increase in testing loss, indicating that the model learns more spurious invariance without rare samples. In contrast, with the training on DOMAIN+, the simultaneous drop of both testing loss and invariance loss demonstrates that constructing a new domain by rare samples is necessary.

Q6. *More splits? No domain labels?*

A6. To explore the potential of DOMAIN+, we propose DOMAIN++, which applies DOMAIN+ on top of an existing split domains by DOMAIN+. Our experimental results are listed in Table 5.8. We find that compared to DOMAIN+, DOMAIN++ cannot further improve the performance, which is even worse than IRM. This suggests that our influence selects sufficient rare samples without spurious invariance, and further selection may introduce unexpected approximation error as the influence estimation is essentially an approximation.

We also implement DOMAIN-ZERO, when there is no domain labels. We first randomly split the training data into two domains and apply our DOMAIN+ to create a new domain. Then we can implement DG methods to learn the invariance. In Table 5.8, when we apply DOMAIN-ZERO on PACS, we still follow the conventional setting but do not use the domain labels. The improvements of IRM-zero (IRM with DOMAIN-ZERO) compared to the original IRM shows the potential future of our influenced-based domain splitting method on more tasks.

Chapter 6

Conclusion

Our research offers significant contributions to the field of Domain Generalization (DG) by introducing DOMAIN+, a novel tool designed to advance the capabilities of current DG methodologies. The crux of most DG methods is domain-invariant feature learning, which is traditionally achieved through imposing a domain-invariant loss. Despite these well-intended mechanisms, our thorough investigation uncovers a pervasive issue hindering true invariance: spurious invariance.

In response to this limitation, we proposed DOMAIN+, specifically designed to mitigate the impact of spurious invariance. DOMAIN+ uniquely addresses this issue by identifying and isolating rare samples into a separate domain. By design, these samples are devoid of spuriously invariant features, thus enhancing the performance of DG methods. This innovative approach differs significantly from existing practices, marking a fundamental shift in the handling of domain generalization.

The success of selecting such rare samples is attributed to our proposed cross-domain influence function, which can be efficiently estimated without re-training. Extensive experimental results on DOMAINBED benchmark show that DOMAIN+ helps existing SOTA methods achieve new SOTAs and outperform the strong ERM baseline.

Furthermore, Our DOMAIN+ has shown substantial promise in advancing DG even in scenarios without any domain annotation. It represents a significant leap forward in DG research. Its capability to streamline DG methods and its potential to function without domain annotations make it a promising tool for future research.

Limitations and future work. While DOMAIN+ marks a significant stride forward, we acknowledge certain limitations and areas for future exploration. One potential limitation lies in the identification and classification of 'rare samples', which may vary significantly across different datasets and domains. The criteria and algorithms for this identification process warrant further refinement and testing. Additionally, the integration of DOMAIN+ into a comprehensive, automated system for DG poses another area for future research. Developing an integrated system would involve streamlining the interaction between DOMAIN+ and various DG methodologies, ensuring compatibility and efficiency across diverse data environments. Exploring the scalability of DOMAIN+ in larger, more complex datasets, and its adaptability to real-world applications are crucial steps forward.

List of Author's Publications

Conference Proceedings

- **Zike Wu***, Jiaxin Qi*, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. DOMAIN+: Splitting a New Influential Domain for Domain Generalization.[1] *Under Review*.
- **Zike Wu**, Pan Zhou, Kenji Kawaguchi, and Hanwang Zhang. Fast Diffusion Model. *Under Review*.

[1] The superscript $*$ indicates the equal contributions

Bibliography

[1] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. xv, 23, 25

[2] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 1

[3] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. 1, 4, 5, 6, 19, 24

[4] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2022.3195549. 1

[5] 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*, pages 1–1, 2022. ISSN 1558-2191. doi: 10.1109/TKDE.2022.3178128. 1

[6] Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Transporting causal mechanisms for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8599–8608, 2021. 1

[7] Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3091–3100, 2021. 1

[8] Jules Sanchez, Jean-Emmanuel Deschaud, and François Goulette. Domain generalization of 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18077–18087, 2023. 1

[9] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021.

[10] Shanliang Yao, Runwei Guan, Zitian Peng, Chenhang Xu, Yilu Shi, Yong Yue, Eng Gee Lim, Hyungjoon Seo, Ka Lok Man, Xiaohui Zhu, and Yutao Yue. Radar perception in autonomous driving: Exploring different data representations, December 2023. 1

[11] Shekoofeh Azizi, Laura Culp, Jan Freyberg, Basil Mustafa, Sebastien Baur, Simon Kornblith, Ting Chen, Nenad Tomasev, Jovana Mitrović, Patricia Strachan, et al. Robust and data-efficient generalization of self-supervised machine learning for diagnostic imaging. *Nature Biomedical Engineering*, pages 1–24, 2023. 1

[12] Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P Lungren, Serena Yeung, and Akshay S Chaudhari. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *NPJ Digital Medicine*, 6(1):74, 2023. 1

[13] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021. 1

[14] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. 1

[15] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 2, 3, 4, 5, 8, 15, 16, 17, 18, 20, 21, 22

[16] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *ICLR*, 2022. 2, 3, 4, 5, 20, 21, 22

[17] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 1, 4, 5, 20, 21, 22

[18] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3716–3725, 2020. 1

[19] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 33:655–666, 2020.

[20] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710, 2021. 1

[21] 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. 2, 5

[22] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018. 3, 5, 20, 21, 22

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3, 13

[24] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[25] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR, 2020. 3

[26] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020. 3

[27] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021. 3

[28] Jiaxin Qi, Kaihua Tang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Class is invariant to context and vice versa: on learning invariance for out-of-distribution generalization. In *European Conference on Computer Vision*, pages 92–109. Springer, 2022. 3

[29] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021. 3

[30] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017. 4, 6, 10, 12, 13, 14, 22

[31] Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *Proceedings of the 37th International Conference on Machine Learning*, pages 715–724. PMLR, November 2020. 4, 6, 27

[32] 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. 5, 20, 21

[33] 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*, 2018. 20, 21, 22

[34] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

[35] 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. 5

[36] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*, 2019. 5, 20, 21, 22

[37] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450, 2021. 5

[38] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. In *ICLR*, 2021. 5, 20, 21, 22

[39] 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. 5

[40] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. In *NIPS*, 2014. 5

[41] Ryo Okumura, Masashi Okada, and Tadahiro Taniguchi. Domain-adversarial and-conditional state space model for imitation learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020. 5

[42] R. Dennis Cook and Sanford Weisberg. *Residuals and Influence in Regression*. New York: Chapman and Hall, 1982. 6, 10

[43] Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6340–6347, 2020. 6

[44] Jakub Sliwinski, Martin Strobel, and Yair Zick. Axiomatic characterization of data-driven influence measures for classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 718–725, 2019. 6

[45] Shuming Kong, Yanyan Shen, and Linpeng Huang. Resolving training biases via influence-based data relabeling. In *International Conference on Learning Representations*, 2022. 6, 14, 27

[46] Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187, 2017. 12, 22

[47] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994. doi: 10.1162/neco.1994.6.1.147. 12

[48] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 13

[49] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 19

[50] 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. 19

[51] 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. 19

[52] 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. 19

[53] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, November 1999. ISBN 978-0-387-98780-4. 20, 21, 22

[54] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020. 20, 21, 22

[55] 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. 20, 21, 22

[56] 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. 23

[57] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017. doi: 10.48550/arXiv.1412.6980. 23

[58] Lian Duan, Lida Xu, Ying Liu, and Jun Lee. Cluster-based outlier detection. *Annals of Operations Research*, 168(1):151–168, April 2009. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-008-0371-9. 26