Master's Thesis of Data Science

Distillation-based Single Source Domain Generalization via Mutual Information Regularization

상호 정보 정규화를 통한
지식 증류 기반 단일 소스 도메인 일반화

August 2023

Graduate School of Data Science
Seoul National University
Data Science Major

Dongkyu Cho

Distillation-based Single Source Domain Generalization via Mutual Information Regularization

Adviser Sanghack Lee

Submitting a master's thesis of
Data Science

July 2023

Graduate School of Data Science
Seoul National University
Data Science Major

Dongkyu Cho

Confirming the master's thesis written by
Dongkyu Cho
July 2023

Chair	Jaejin Lee	(Seal)
Vice Chair	Sanghack Lee	(Seal)
Examiner	Taesup Kim	(Seal)

Abstract

Machine learning frequently suffers from the discrepancy in data distribution, commonly known as domain shift. Single-source Domain Generalization (sDG) is a task designed to simulate domain shift artificially, in order to train a model that can generalize well to multiple unseen target domains from a single source domain. A popular approach is to learn robustness via the alignment of samples generated by data augmentation. However, prior works frequently overlooked what can be learned through such alignment. In this paper, we study the effectiveness of augmentation-based sDG methods by connecting recent identifiability results by Von Kügelgen et al. [89]. We highlight the overlooked issues in using augmentation for OOD generalization and search ways to alleviate them. We introduce a novel sDG method that leverages pretrained models to guide the learning process via a feature-level regularization of mutual information, which we name PROF (Progressive mutual information Regularization for Online distillation of Frozen oracles). PROF can be added to conventional augmentation-based methods to dampen the fluctuation of the OOD performance. We further introduce a data-effective alignment objective as well as a novel augmentation method for fine-grained simulation of domain shift.

Keyword: Domain Generalization, Causal Representation Learning

Student Number: 2021-20711

Table of Contents

Chapter 1

Introduction

Distribution shift is prevalent in many machine learning settings. The term is often referred to as *domain shift*, where a domain is understood as the joint probability distribution from which samples are drawn. An important aspect of domain shift is that it severely hinders the generalizability of trained models [45]. The issue is easily observable when a model trained in a source domain suffers in a target domain that is inconsistent with the source. Domain Generalization (DG) is a task designed to simulate domain shift, where the model is given multiple labeled datasets at training time, where the main objective is to accomplish *out-of-distribution generalization* across unseen domains [3, 5, 19, 76].

Single-source Domain Generalization (sDG) is a variant of DG, where only a single source domain is provided at train time. The shortage of data and the absence of additional source domains make sDG challenging, mainly because conventional DG methods that leverage multiple domains cannot be easily adopted. To overcome such barriers, prior works on sDG often utilize data augmentation to generate unseen domains [7, 48, 68, 88, 90, 93] and learn domain-invariant features through an alignment of the generated domains using contrastive loss [37, 61].

However, there is a relative void in the discussion on what is learned through the alignment of augmented samples. In this paper, we analyze the effectiveness of

augmentation-based sDG approaches from a novel perspective of style-content disentanglement. Style-Content (S-C) disentanglement aims to identify a partitioned latent space, namely style and content [30, 31, 36, 71, 72]. While the definitions of style and content vary across settings, here we define content as latent features that are invariant across augmentations (i.e. augment-invariant), while style is the latent feature subpart that changes with the augmentation. Recently, Von Kügelgen et al. [89] studied an interesting connection between S-C disentanglement and data augmentation, demonstrating that contrastive learning provably learns to retrieve the augment-invariant features under some assumptions. We connect the discovery to the sDG literature to analyze the effectiveness of retrieving domain-invariant information from augmented data. We examine the problem from a causal standpoint by illustrating it via a causal graph [65].

We state our contributions as the following. (1) We analyze the single source domain generalization task through the lens of S-C disentanglement and highlight the difficulties of learning domain-invariant information from augmentation-based sDG methods. (2) To mitigate the issues brought by the aforementioned obstacles, we introduce a novel method PROF that functions as a regularizer for sDG. (3) We further devise a novel alignment objective MDAR (Multi-Domain Alignment with Redundancy reduction) that serves as a strong baseline for sDG.

Chapter 2

Preliminaries

Learning domain agnostic models from limited source domains is a longstanding area of investigation. In this section, we revisit related works on S-C disentanglement and domain generalization.

Style-Content Disentanglement Style-Content disentanglement seeks to separate the aggregated latent variable into two parts, denoted as style and content [71, 72]. While the term style and content originated from the style transfer literature [14, 55, 82], recent works try to push the idea further using concepts of causal inference [64–66] and Independent Component Analysis (ICA) [6, 18, 20, 53, 70, 89]. Inspired by previous discoveries on nonlinear ICA [32], newly proposed techniques utilize auxiliary variables to assure conditional independence between the entangled features (e.g., time stamp or environment index, domain index [20, 30, 31, 36, 41]). The concept has been put into use in a number of fields, namely adversarial learning [55], transfer learning [73], and domain generalization [3, 35, 54, 92]. Notably, disentanglement is used to elucidate the underlying mechanism of self-supervised learning [56, 77, 79, 85, 104] and data augmentation [26, 33, 89].

Domain Generalization In the multi-source domain generalization field, disentanglement of domain-invariant features have shown great success in training robust domain-agnostic models [3, 9, 50, 74, 100] by leveraging shared information across domains. To learn domain-invariant information, researchers commonly analyze the data generating process (DGP) using structural causal models to design effective algorithms [35, 54, 92]. On the contrary, disentanglement is rarely discussed in the sDG literature. This is due to innate conditions of sDG, where only one domain is available for training. This setting makes it hard to apply conventional disentanglement approaches developed in the multi-DG literature. To tackle this, a line of work focuses on how to augment *unseen* domains effectively with generative models [13, 48, 68, 88, 90, 93]. However, there is a lack of discussion on whether augmented samples can simulate unseen domains, or whether it can be used to learn domain-invariance. Apart from augmentation-based sDG methods, others approached the task from a meta-learning viewpoint [17], or adopt concepts inspired by causal inference [51]. The task is often modified to solve real-world problems (e.g., medical image processing [52, 62, 81]). A recent movement in the multi-DG literature highlights the use of pretrained models for OOD generalization, leveraging the knowledge of the pretrained models [5, 44, 49, 96]. Such works closely resemble the methods introduced in the Knowledge Distillation (KD) literature [1, 2, 24, 78, 84]. However, unlike most KD works that focus on transferring i.i.d knowledge, our study focuses on using KD for OOD robustness [60, 91].

Chapter 3

Limitations of Augmentation for sDG

In this section, we present an overlooked problem of augmentation-based sDG methods. Specifically, we revisit recent works on S-C disentanglement to analyze the effectiveness of utilizing augmentation for out-of-domain generalization.

A general view towards augmentation-based sDG methods We present a general expression for augmentation-based sDG methods and discuss their effectiveness. Generally, augmentation-based methods can be expressed as *augment and align*, minimizing the following objective (omitting some arguments for simplicity) denoting x and $\bar{x}$ as an original sample and its augmented view:

$$L := L_{ce} + L_{\text{MaxEnt}}(x, \bar{x}; \Phi). \tag{3.1}$$

where L_{ce} the cross-entropy loss $L_{ce}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i y_i \log(\hat{y}_i)$ with $\mathbf{y}$ the ground truth label vector, $\hat{\mathbf{y}}$ the softmax prediction of the model, y_i and $\hat{y}_i$ the *i*-th dimension of $\mathbf{y}$ and $\hat{\mathbf{y}}$, respectively, and L_{MaxEnt} is an objective that simultaneously aligns the mapped representations $\Phi(x)$ and $\Phi(\bar{x})$ under entropy regularisation, where Φ is a feature extractor. Commonly, a self-supervised contrastive loss (hereinafter contrastive loss) is used as L_{MaxEnt} [61, 99]. Recently, Von Kügelgen et al. [89] showed that the optimization of a contrastive loss provably minimizes L_{MaxEnt}, learning Φ to extract

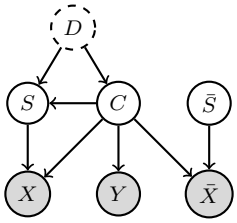


Figure 3.1: A causal diagram depicting DGP under data augmentation. The shaded nodes are observable.

features that are augment-invariant, under a certain condition. In this perspective, conventional augmentation-based sDG methods could be understood as retrieving augment-invariant features.

A causal interpretation of data augmentation In this section, following [89], we illustrate the underlying data generating process (i.e., DGP) using a causal graph and incorporate data augmentation into the causal graph under sDG setting.

An instance of a given labeled dataset is typically composed of an observation X (i.e., image) and its label Y. Although supervised learning predicts Y directly from X, this does not reflect the underlying causality. We can think of the existence of hidden features (e.g., real-world attributes regarding the subject of the image and the background), which we will refer W, that affect both the image and label. At this moment, the causal graph for DGP can be simply represented as $X \leftarrow W \rightarrow Y$ where W is unobserved. Now, we incorporate data augmentation into the picture. Given *label-preserving* augmentation methods, we attain $\bar{X}$ the augmented view of X. Such an augmentation can be considered as manipulating only the style S (augment-variant) to yield $\bar{S}$ while retaining its content (augment-invariant) C where C and S partitions W, that is, $W = (C, S)$ (see [89] for a detailed discussion). Yet, this does not imply that C and S are independent. C causally affects S (also corroborated by experimental results [39, 89]). A way to understand this separation is by viewing such an augmentation as a soft intervention [11] on S, resulting in a modified style $\bar{S}$. By definition, $(C, \bar{S})$ becomes the hidden features of $\bar{X}$. Furthermore, C consistently affects Y regardless of

the *label-preserving* augmentation. This understanding results in the graph in Fig. 3.1 (W is implicit) except D .

Von Kügelgen et al. [89] showed that under certain conditions, the above DGP is sound, and augmentation separates C and S. However, this picture misses an important variable: the domain D. By definition, observations are drawn from the distribution of the domain, thus latent variables W are affected by the domain the data is generated from. Therefore it is unavoidable to incorporate a variable indicating domain D in the figure. In sDG, D is fixed in the sense that we are given just one domain. Due to the single source setting, we cannot distinguish what information is shared across different domains, leaving both C and S potentially affected by D. For that reason, unless the domain shift between the source and target is moderate, optimizing solely the augment-and-align objective Eq. (3.1) would be insufficient to address the issue caused by a large domain gap.

Learning to ignore To address a large domain shift, we first begin with some observations. Conventional *augment and align* methods are vulnerable to domain shift in the sense that its effectiveness is strongly affected by the augmentation's proximity to the domain shift. While advanced augmentation methods may simulate small shifts in distribution (e.g., MNIST $\rightarrow$ USPS in Digits Sec. 5.1), it is very hard to approximate large domain shifts (e.g., Photo $\rightarrow$ Sketch in PACS Sec. 5.1). If the gap between the source and target domain is large, failure in simulating domain shift would make its augment-invariant features less relevant to domain-invariant features, leading to overfitting to the source domain.

To avoid learning irrelevant features, we can think of a hypothetical regularizer that encourages the model to learn information relevant to domain-invariance, while discouraging domain-specific features. Certainly, this would require a condition that the regularizer be an oracle that can distinguish domain-invariant information. Using this oracle regularizer, we hope to solve the phenomena commonly associated with the large

domain gap. Especially, the mid-training fluctuation of OOD performance [48, 68, 90, 93]. We view that the fluctuation is strongly correlated with the challenge in acquiring domain-invariant features under a large domain gap. We empirically observe that the level of domain gap between the *source* and *target* closely matches the magnitude of the mid-train fluctuation, where the increase in domain gap is simultaneously observed with the increase in fluctuation. Detailed information regarding the measure of domain gap is included in Sec. 5.3.1. We view that mid-training fluctuation is a serious issue since it manifests that the simulated domains do not properly reflect unseen domains and, further, it harms the credibility of learned models due to uncertainty in its real-world performance. In the following section, we search ways to implement the *hypothetical* oracle regularizer, inspired by works in knowledge distillation.

Chapter 4

Leveraging Pretrained Models for Domain Invariance

We present a novel single source domain generalization method for image data, where the aim is to alleviate the issue of mid-train fluctuation. The overview of our architecture is depicted in Fig. 4.1. At large, the architecture for our method involves three neural networks, a domain generator G, task model classifier F, and an oracle O. We sequentially learn multiple domain generators $\{G_k\}_{k=1}^{K}$ and use the samples created by the generators (i.e., augmented samples) to train the task model F. More specifically, the generators provide the task model with *challenging* augmented samples, while the task model guides the generator to create *valid* augmentations. We train the above process using a combination of two losses: $L = L_f + w_g \cdot L_g$ where L_f (4.9) and L_g (4.7) are the loss used to train the task model and the generator, respectively. $w_g \in \{0, 1\}$ is a switch used to control the training of G. The exact forms for L_f and L_g will become clear at the end of this section.

We build our method upon the idea that learning domain-invariance solely from augmented domains is vulnerable to overfitting to the source, especially when the domain gap is too large to simulate via data augmentation. To alleviate this issue, we propose an oracle regularizer: under the hypothesis that the oracle is capable of generalizing well to unseen domains, we use the oracle to guide the task model to become less domain dependent. Specifically, our oracle regularization objective

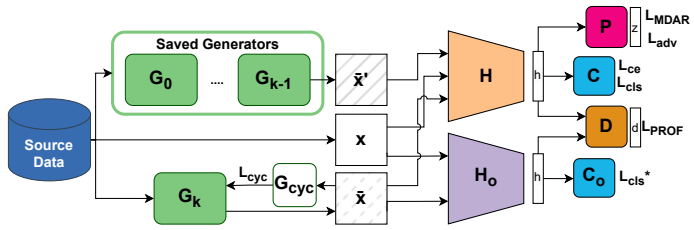


Figure 4.1: The illustration of our method. We sequentially train multiple generators $G_{0\ldots K}$. The Oracle H_o regulates the task model H's learning process. During the training, multiple modules (e.g., P, D, C, C_o) are used for optimization.

regulates the sDG process via an alignment between hidden feature representation of the task model and the oracle, which we name PROF. In the following section, we elaborate our ideas in depth.

Notation We begin by introducing related notations regarding our method. To begin with calligraphic letters are used to denote state space of a variable. For example, $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{H}$ respectively represent the space of the input image, intermediate feature representation, and labels.

- **Task model:** The task model $F = C \circ H$ consists of a feature-extractor $H : \mathcal{X} \to \mathcal{H}$ and a classification head $C : \mathcal{H} \to \mathcal{Y}$.
- **Generator:** A trainable generator $G : \mathcal{X} \to \mathcal{X}$ consists of an encoder-decoder architecture with a style-transfer module placed between the encoder and decoder.
- **Oracle:** The oracle model $O = C_o \circ H_o$ consists of a frozen feature-extractor $H_o : \mathcal{X} \to \mathcal{H}$ and a trainable classification head $C_o : \mathcal{H} \to \mathcal{Y}$. Both F and O use separate feature-extractors (H and H_o) to map the input data as intermediate representation and pass the representation to the classification head (C and C_o) for the downstream classification task. Yet, we match the dimension of representation for the oracle and task model.

- **Distillation Head:** The distillation head $D : \mathcal{H} \to \mathcal{D}$ is used to impose regularization for the task model via oracle's representation. Instead of directly comparing the intermediate representation in $\mathcal{H}$, representations from H_o and H are mapped through the shared distillation head.

- **Projection Head:** Similar to the distillation head, the task model uses a projection head $P : \mathcal{H} \to \mathcal{Z}$ to project the intermediate representations into a different dimension. The projection head is reserved for alignment of augmented views with MDAR, and its associated adversarial loss L_{adv}, thus not for PROF.

4.1 Oracle Regularizer

We devise a novel learning method PROF (Progressive mutual information Regularization for Online distillation of Frozen oracles) to guide the learning process. PROF reformulates the sDG problem under the assumption that if there exists an oracle model O that can generalize well to unseen domains, we can leverage the oracle to learn sDG. The objective for PROF can be formulated as:

$$L_{\text{PROF}}(x, \bar{x}) = \text{BT}(D(H(x)), D(H_o(x)), \lambda_{\text{PROF}}) + \text{BT}(D(H(\bar{x})), D(H_o(\bar{x})), \lambda_{\text{PROF}}), \tag{4.1}$$

where x denotes the original sample and $\bar{x}$ the augmented view created by G, λ_{PROF} is a user-set parameter, and Barlow Twins (BT) is defined as [99]:

$$\text{BT}(z, z^+, \lambda) = \textstyle\sum_i (1 - M_{ii})^2 \; + \lambda \sum_i \sum_{j \neq i} M_{ij}^2, \tag{4.2}$$

where M refers to the cross-correlation matrix of the two positive-pair feature representations z, z^+, and λ a user-set parameter.[1] BT (4.2) is a feature-decorrelation loss originally introduced in [99] as a contrastive learning objective BT is a combination of two terms balanced via a hyperparameter λ, where the first term $\sum_i (1 - M_{ii})^2$ aligns

[1]The actual computation involves a batch of data to obtain an empirical cross-correlation matrix.

two representations by spurring the diagonal values in the cross-correlation matrix M of (z, z^+) to be 1 while the second term $\sum_i \sum_{j \neq i} M_{ij}^2$ minimizes redundancy in the representation by encouraging the off-diagonal values to be closer to 0.

Discussion The idea of PROF is that we can distill the oracle's knowledge into the task model by maximizing the shared information between the two models. PROF aims to maximize the MI between the intermediate output features of the two feature-extractors $H_o(x)$ and $H(x)$. PROF functions as a regularization term that guides the task model from deviating too far from the oracle, encouraging the student task model to learn the oracle's behavior on data. From this perspective, an intended objective for PROF could be formulated as

$$\max_{H} \; I(H(x); H_o(x))$$

where H, H_o is respectively the feature-extractor of the oracle O and the task model F and $I(X;Y) = \mathbb{E}_{p(x,y)}[\log p(x \mid y)/p(x)]$ indicates the mutual information (MI). However, directly estimating and optimizing MI are challenging, as exact estimation of MI is intractable [63]. There exists InfoNCE loss [61] which adopts a lower bound of MI [67] as a surrogate objective for MI optimization:

$$I_{\text{NCE}}(X;Y) \triangleq \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K} \log \frac{\exp(f(x_i, y_i))}{\frac{1}{K}\sum_{j=1}^{K} \exp(f(x_i, y_i))}\right] \leq I(X;Y).$$

However, a problem of InfoNCE as a variational bound of MI is that InfoNCE requires a large batch size for convergence [25, 78], making it doubtful for use in small datasets (e.g., PACS [47]). Consequently, we indirectly approximate InfoNCE with a feature decorrelation loss [99], based on empirical and theoretical results that show its functional proximity [27, 83, 89]. Contrary to InfoNCE, the feature decorrelation converges effectively with small batch sizes and large vector dimensions.

Now we discuss the availability of an oracle. In reality, oracles are not readily available. However, previous studies [5, 44, 49] report that models that are pretrained from a large domain (e.g., ImageNet [75]) or with deeper models (e.g. RegNet [69, 80],

ViT [10]) tend to generalize better at unseen domains. Hence we utilize a pretrained model trained on a larger domain as an oracle. To preserve the knowledge of the oracle, we freeze the feature-extractor H_o of the oracle.

4.2 Learnable Domain Shift Simulators

We sequentially train multiple ($k = 1 \dots K$) generators to obtain varying simulated domains, adopting methods of Li et al. [48], Wang et al. [93]. The objective of the generator is to generate a sufficient set of label-preserving augmentations that intervene on the domain-specific features. To simulate domain shift, we must assure that the augmented domain is label-preserved, while different from the source domain. The objective for generator L_g is a weighted sum of following losses:

$$L_{cls}(\bar{x}, y) = L_{ce}(C(H(\bar{x})), y). \tag{4.3}$$

$$L_{cyc}(x, \bar{x}) = \|x - G_{cyc}(\bar{x})\|_2. \tag{4.4}$$

$$L_{div}(\bar{x}_1, \bar{x}_2) = -\|\bar{x}_1 - \bar{x}_2\|_2. \tag{4.5}$$

$$L_{adv}(x, \bar{x}) = -\mathrm{BT}(P(H(x)), P(H(\bar{x})), \lambda_{adv}), \tag{4.6}$$

L_{cls} is a cross-entropy loss that assures the validity of the generated samples $\bar{x}$.[2] L_{cyc} ensures that the output of G, when passed through model G_{cyc} in the opposite direction should be as similar (i.e., recoverable) as possible to the original input image [103]. The two losses originate from existing works on sDG [48, 93] to assure the consistency of the generated samples. L_{div} is a negated L2-norm between two augmented views $\bar{x}_1$ and $\bar{x}_2$ of the same x created with the generator. Intuitively, optimizing with L_{div} encourages the generator to augment diverse samples. L_{adv} is an adversarial loss function designed to reverse the feature-decorrelation process in Eq. (4.8) by negating the loss used in Eq. (4.2). We train the generator with the weighted sum L_g of the above four objectives

[2]When PROF is enabled, the classification loss of the oracle $L_{ce}(C_o(H_o(\bar{x})), y)$ is added to L_{cls}.

(w_{adv} is set to 0 when PROF is used without MDAR):

$$L_g = L_{cls} + w_{cyc} \cdot L_{cyc} + w_{div} \cdot L_{div} + w_{adv} \cdot L_{adv}. \tag{4.7}$$

Note that L_g is added to the total loss L with a weight $w_g \in \{1, 0\}$ that functions as a gate in training the k^{th} generator, set as $w_g = 1$ during the first half of the training epochs for G_k, then as $w_g = 0$ to stop learning.

Inspired by fine-grained studies on domain shift [35, 94], we upgrade the domain generator to intervene on a wider variety of attributes. Whilst previous augmentation methods [48, 93] were limited to manipulating certain attributes (e.g., color, stroke), our method further allows spatial manipulations (e.g., shape, location) using Spatial Transformation Networks (STN) [34]. No additional requirements are needed for training STN. We elaborate further on the generator architecture in Sec. 5.2.1.

4.3 Multi-Domain Alignment with Redundancy Reduction

We now introduce a novel alignment objective MDAR (Multi-Domain Alignment with Redundancy reduction). MDAR aims to disentangle latent features that are invariant across multiple augmented views. We design MDAR as a fair baseline of conventional *augment and align* method.

In learning the k^{th} generator G_k, we create an augmented view $\bar{x}$ for a batch of original samples x using the k^{th} generator G_k. We then randomly load two previously learned generators to construct another augmented views $\bar{x}', \bar{x}''$. With $\{x, \bar{x}, \bar{x}', \bar{x}''\}$, we use the projection head $P \circ H$ to get their corresponding representations $\{z, \bar{z}, \bar{z}', \bar{z}''\}$. Using BT (4.2), we encourage a pair of non-identical representations (e.g., $z_i, z_j,\ i \neq j$) vary in a similar way, that is, their cross-correlation matrix M to be closer to an identity matrix. We compute the BT (4.2) across all possible ordered pairs within the set of $\{z, \bar{z}, \bar{z}', \bar{z}''\}$ (i.e., $\zeta := \{(z, \bar{z}), (z, \bar{z}'), (z, \bar{z}'') \dots\}$) and use the average of all the

pairwise losses as our alignment loss L_{MDAR} (4.8) written as:

$$L_{\text{MDAR}}(x, \bar{x}, \bar{x}', \bar{x}'', \lambda_{\text{MDAR}}) = \frac{1}{n_\zeta} \sum_{\{a,b\}\in\zeta} L_{\text{BT}}(a, b, \lambda_{\text{MDAR}}), \tag{4.8}$$

where n_ζ denotes the length of ζ and λ_{MDAR} a user-set parameter.[3] Intuitively, via optimizing L_{MDAR}, we can train the generator in a way that multiple views $\{z, \bar{z}, \bar{z}', \bar{z}''\}$ are aligned. In terms of S-C disentanglement, MDAR retrieves the augment-invariant features. Different from the commonly used InfoNCE loss [61], our self-supervised objective (4.8) does not require any negative pairs, thus works well on small batch sizes [27, 86, 99], suitable for benchmarks like PACS.

We train the task model F using a weighted combination of multiple losses, the baseline cross-entropy loss L_{ce} of both x and $\bar{x}$, with L_{PROF} written as:

$$L_f = L_{ce}(C(H(x)), y) + L_{cls} + w_{\text{PROF}} \cdot L_{\text{PROF}} \tag{4.9}$$

where w_{PROF} is a user-set parameter. In Chapter 5, we further use a variant $L_f = L_{ce}(C(H(x)), y) + L_{cls} + w_{\text{MDAR}} \cdot L_{\text{MDAR}}$ for baseline experiments.

[3] n_ζ can be switched with ${n_\alpha}^2$ in Eq. (4.8), where n_α is the length of $\{z, \bar{z}, \bar{z}', \bar{z}''\}$.

Chapter 5

Experiment

We present our experimental settings including datasets and model architectures. Then, we report our experimental results using the accuracy for each target domain, as well as the mean accuracy over all target domains. For reproducibility, all experiments were conducted using a fixed random seed.

5.1 Experimental Settings

Datasets Following the experimental settings in prior sDG works [48, 68, 93], we adopted three broadly used benchmarks for our sDG problem. **(PACS)** PACS [47] is widely used to test the generalizability of trained models against domain shift. It consists of 4 domains of differing style (Photo, Art, Cartoon, and Sketch) with 7 classes. We follow the setting of Wang et al. [93], train our model with the Photo domain, and evaluate on the remaining target domains (Art, Cartoon, and Sketch). Among the three benchmarks, PACS is the main target of PROF due to its large gap between domains. **(Corrupted CIFAR-10)** Corrupted CIFAR-10 (i.e. CIFAR-10-C) is a benchmark to test the image classifier robustness under distortion [22]. We train our model with the train split of the CIFAR-10 [42] dataset and test the model accuracy in CIFAR-10-C. We evaluate the robustness of the model with 19 types and 5 levels of corruption.

This benchmark shows small distributional divergence between datasets since each target domain is created via augmentation of the source domain (CIFAR-10). Hence we expect that the task is sufficient with conventional *augment and align* methods. **(Digits)** The Digits dataset is a popular benchmark for sDG, comprised of 5 different Digit classification datasets (e.g., MNIST [8], SVHN [59], MNIST-M [16], SYNDIGIT [15], USPS [46]). In our Digits experiment, we train our model with the first 10,000 samples of the MNIST dataset and assess its generalization accuracy across the remaining four domains.

Implementation Summary In all experiments, we utilized, for the task model, the identical architectures used in previous sDG works, where we use additional heads D and P [48, 68, 90, 93]. For PACS, we adopted AlexNet [43] pretrained on Imagenet [75], finetuned on the photo domain. We followed the original split of the PACS dataset [47] for fair comparison. For corrupted CIFAR-10, we used a Wide Residual Network [98] of depth 16, and width 4, pretrained on the CIFAR-10 dataset. For Digits, we used the identical network architecture (i.e. `conv-pool-conv-pool-fc-fc-softmax`) used in previous works, pretrained on MNIST. For the oracle, we selected pretrained models appropriate for each experiment. For PACS, we chose a RegNetY-16GF [69] pretrained on the Instagram dataset with SWAG (Supervised Weakly through hashtAGs) [80] following experimental reports of Cha et al. [5], Li et al. [49]. For Corrupted CIFAR-10, we select a ResNet50 [21] pretrained on the ImageNet-1K dataset [75]. The oracle is finetuned on the source domain (e.g. Photo, CIFAR-10) and frozen. We test the sensitivity of the hyperparameters using the validation split of the source dataset. Information of each hyperparameter is included in Sec. 5.2.4.

Before training, we pretrained the models with the train split of the source domains. Details on the pretraining are provided in Sec. 5.2.3. For PACS, we sequentially train 20 generators and 30 epochs for each generator with a batch size of 64. For Corrupted CIFAR-10, we sequentially train 20 generators and 30 epochs for each generator

with a batch size of 256. For Digits, we trained 100 generators with 10 epochs for each generator with a batch size of 128. For all experiments, we employed Adam [38] optimizer with a learning rate of 1e-4. Detailed information about the training hyperparameters is included in Sec. 5.2.2. All experiments were executed using random seed control for reproducibility.

5.2 Implementation Detail

In this section, we report the implementation details of our method.

5.2.1 Model Architecture

We report the details of model architectures used in our experiments. All models were built to match the architecture used in previous studies.

Task Model The task model architecture varies in each experiment. For each experiment, we report the feature extractor H, including an additional layer (i.e. buffer) used to match the feature extractor's output dimension to the oracle's.

The model used in the PACS experiment is AlexNet [43]. The model consists of 5 convolutional layers with channels of {96, 256, 384, 384, 256}, followed by two fully-connected layers of size 4096 units. The buffer is a 2-layered MLP that maps the output dimension 4096 to that of the oracle (RegNetY-16GF), which is 3024. Hence, the final output dimension of the feature extractor is 3024.

The model used in the Corrupted CIFAR-10 experiment is a Wide Residual Network (i.e. WRN) of width $w = 4$ and depth 16 [98]. WRN is a model that boosts its performance by widening the network by a certain factor w. The model consists of 4 network blocks with channels incrementally increasing as $\{16, 16w, 32w, 64w\}$. Specifically, the 4 blocks refer to an initial convolutional layer, followed by three additional network blocks. We further follow the original WRN design and set the dropout rate as 0.3. The buffer is a 2-layered MLP that maps the output dimension 256

to that of the oracle (ResNet50), which is 2048. Hence, the final output dimension of the feature extractor is 2048.

For the model used in the Digits experiment, please refer to Sec. 5.1. The architecture consists of two 5×5 convolutional layers, with 64 and 128 channels respectively. Each convolutional layer is followed by a MaxPooling layer (2×2). The network also includes two fully-connected layers with sizes of 1024, 1024 being the final output dimension of the feature extractor. Since we do not employ oracle for the Digits experiment, a buffer was not added.

Generator In this section, we describe the generator in detail. While the design of the generator slightly varies in each experiment, the basic architecture is the same. The generator consists of an encoder and a decoder, with a spatial transformer network (STN) and a style-transfer module in between the encoder and the decoder. The four components are placed in the order of `Encoder - STN - Style-Transfer - Decoder`.

We begin by illustrating the overall process of how an image is augmented by the generator. First, the input image is passed through the encoder to get a feature representation vector. The feature vector is then passed through the STN and the style-transfer module for modification. The modified vector is then reconstructed via a decoder, returning an augmented image. The mentioned process is illustrated in Fig. 5.1. In the figure, we depict how each module modifies the input image.

STN is a module that learns to perform spatial transformations on the input [34]. During the process, the STN module learns transformation parameters, where the parameters each define the magnitude of spatial transformations (e.g., rotation, scaling, translation). The STN module can be inserted at any point in the generator, allowing the generator to selectively transform the data up to a degree that is label-preserving. We place the STN right after the Encoder, following the experimental results of the original paper [34]. In Fig. 5.1, we can see that the STN performs spatial transformations,

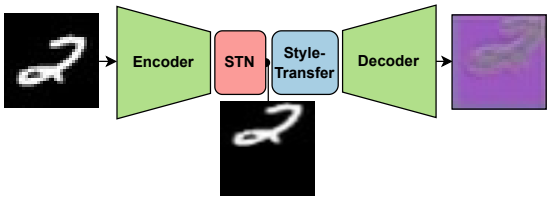


Figure 5.1: The illustration of the Generator.

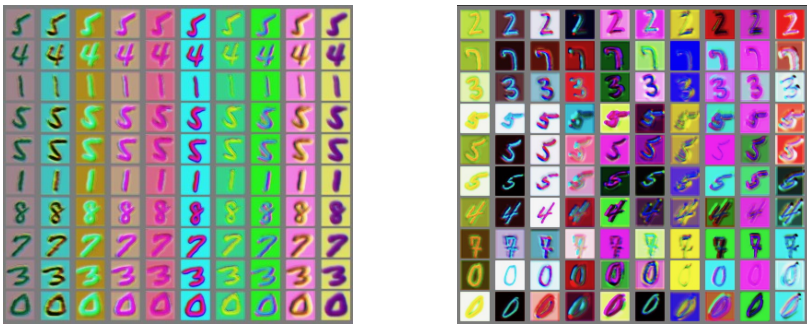
creating the modified image at the middle.

The style-transfer module modifies the features of the input image by adjusting the mean and standard deviation of the image features. This is performed using a normalization technique called Batch-Instance Normalization (i.e. BIN) [58]. BIN selectively normalizes the features of the input image that are of less significance, while preserving features that are important. Note that this module is a modified version of the AdaIN method introduced in [28], where we switched the normalization method from Instance Normalization [87] to BIN for effective style transfer.

We share the results of applying these modifications in Fig. 5.2. When compared to conventional style-transfer methods [28], our generator modifies a wider range of attributes. For instance, in the right image of Fig. 5.2, we can observe that the images generated using our method displayed a large variance in shape, position, and color. However, an observable limitation is that the STN cannot transform complex images as in PACS, as small spatial modifications vastly change the semantics of the image. As depicted in Fig. 5.3, the effect of the spatial modification is limited on PACS images.

Oracle Here, we report the architecture of the oracle. The oracle varies on the type of the experiment, (1) a RegNetY-16GF for the PACS experiment, (2) a ResNet50 for the corrupted CIFAR-10 experiment.

The RegNetY-16GF is a variant of the RegNet family, a line of models introduced in [69] for image classification. The name of the model indicates its configurations, where the "Y" indicates the convolution method, and the "16GF" represents the model's



(a) Conventional Style-transfer (b) STN + Style-transfer

Figure 5.2: The illustrated comparison of the generators.

Figure 5.3: The illustration of generated images (PACS).

capacity or complexity. We implement the model, and its model weights using the torchvision [12] library. We used the weights pretrained via end-to-end fine-tuning of the original SWAG [80] weights on the ImageNet-1K data [75]. We then fine-tuned the pretrained model again with the Photo domain of PACS for 200 epochs, with a learning rate of $1e-4$ using the SDG optimizer and the Cosine Annealing learning rate scheduler, a batch size of 64. The same configuration was used for the additional experiments where the source domain was switched to the Art, Cartoon, Sketch domain in PACS.

The ResNet50 is a variant of the ResNet family, a series of image classification models introduced in He et al. [21]. The name of the model indicates its depth, where

"50" marks the number of layers. We implemented the model and its model weights using the torchvision library. For ResNet50, we used the weights pretrained with the ImageNet-1K dataset. We finetuned the pretrained ResNet50 with the CIFAR-10 dataset, the source domain of the corrupted CIFAR-10 experiment. In detail, we trained for 100 fine-tuning epochs, with a learning rate of $1e-4$ with the SDG optimizer and the Cosine Annealing learning rate scheduler, a batch size of 64.

5.2.2 Model Training

In this section, we elaborate on the details of the training process. We explicitly state the training hyperparameters (e.g., number of simulated domains (K), number of inner training loops for each generator, learning rate, the type of the optimizer, learning rate scheduler, and batch size). We further state the configurations of the projection heads (e.g., projection dimension ($\mathcal{Z}$) of the projection head P, projection dimension ($\mathcal{D}$) of the distillation head D).

PACS For the PACS experiment, we set K as 20, training each generator with 30 inner loops. During the first 15 inner loops we train the generator, and stop the training during the last 15 loops. We manually set the number of epochs by analyzing the training behavior of the generators. We set the learning rate as $1e-4$, using the Adam optimizer [38]. The batch size was set as 64. Regarding the model architecture, both the projection dimension ($\mathcal{Z}$) and the distillation head projection dimension ($\mathcal{D}$) were set as 1024.

Corrupted CIFAR-10 For the Corrupted CIFAR-10 experiment, we set K as 20, and 20 inner loops. During half (10) of the inner loops, we trained the generator and stopped the training during the remaining 10 inner loops. We set the learning rate as $1e-4$, with the Adam optimizer. The batch size was set as 256. The projection dimension ($\mathcal{Z}$) and the distillation head projection dimension ($\mathcal{D}$) were both set as 512.

Digits For the Digits experiment, we set K as 100, with 10 inner loops. Similar to the above two experiments, we trained the generator for 5 epochs and stopped the training for the other 5. Furthermore, the learning rate was tuned as $1e-4$, using the Adam optimizer. The batch size was set as 128. Finally, the projection dimension ($\mathcal{Z}$) was set as 128.

5.2.3 Model Pretraining

In this section, we report the information regarding the pretraining process. As mentioned above, we pretrained our task model with the source domain prior to the main training procedure. We announce the number of pretraining epochs, the learning rate, the optimizer, the learning rate scheduler, and the batch size.

PACS We pretrained the AlexNet with the train data of the Photo domain, using the train split introduced in the original paper [47]. We pretrained the model for 60 epochs, with a learning rate of $5e-3$ using the SGD optimizer. We further used the Step learning rate scheduler with a gamma rate (i.e. the strength of the learning rate decay) of 0.5. The batch size was set as 32. The same pretraining method was used for additional experiments on PACS where the source domain was changed to Art, Cartoon, Sketch.

Corrupted CIFAR-10 For the corrupted CIFAR-10 experiment, we pretrained the WRN with the train split of CIFAR-10. The pretraining epochs was set as 200, with a learning rate of $1e-1$ using the SGD optimizer. We used the Multi-Step LR scheduler, setting the gamma rate as $2e-1$, with milestones set as $\{60, 120, 160\}$. Hence, every time the training epoch reaches the milestone, the learning rate was reduced to one-fifth of the previous rate. The batch size was set as 128.

Digits Lastly, for the Digits experiment, we set the number of pretraining epochs as 100, with a learning rate of $1e-4$ using the Adam optimizer. The batch size was set as

256.

5.2.4 Hyperparameters

In this part, we state the hyperparameters used in our experiments.

λ_{PROF} λ_{PROF} is a balancing coefficient for L_{PROF}, an objective adopting the feature-decorrelation loss introduced in Zbontar et al. [99]. We tuned λ_{PROF} using experimental results of the original paper and [86]. In the original paper, the author reported the optimal value of the balancing term as 0.005, which remains consistent under varying projection dimensions. We set this as a starting point for hyperparameter tuning. We find that if λ_{PROF} balances the off-diagonal term (i.e. redundancy reduction term) and the diagonal term (i.e. alignment term) to a similar degree, no significant differences are observed. Furthermore, switching λ_{PROF} to $\frac{1}{d} \approx 0.0001$ showed no significant changes to the learning process. Here, d denotes the projection dimension of the distillation head. While we cannot guarantee an optimal value for λ_{PROF}, we set $\lambda_{\text{PROF}} = 0.005$ for our two experiments using PROF.

$\lambda_{\text{MDAR}}, \lambda_{adv}$ The hyperparameters λ_{MDAR} and λ_{adv} is used together for adversarial learning, hence we report the two together. λ_{MDAR} was set in a similar way as λ_{PROF}. For our experiments, λ_{adv} was set as 0.005. λ_{adv} was searched under a fixed value of $\lambda_{\text{MDAR}} = 0.005$. We experimented with varying values of λ_{adv}: $\{0.005, 0.05, 0.5\}$, which showed no significant difference to the training process, while 0.05 showed slightly better results in the validation set of the source domain. Hence, in our experiments, λ_{adv} was set to 0.05. To explicate, generally, L_{adv} displayed a value approximately 10 times larger than L_{MDAR}. We believe that this behavior is correlated to 0.05 being a good value for λ_{adv} under a fixed value of $\lambda_{\text{MDAR}} = 0.005$.

All other hyperparameters (e.g., w_{cyc}, w_{div}, w_{adv}, w_{PROF}) are searched with a similar method to Li et al. [48]. For all experiments, we set w_{cyc} as 20.0, w_{cyc} as 2.0, and w_{adv} as 0.1 in Digits, and 0.02 in PACS and Corrupted CIFAR-10. Finally, w_{PROF}

Table 5.1: sDG accuracy on PACS.

Method	A	C	S	Avg.
ERM [40]	54.43	42.74	42.02	46.39
JiGen [4]	54.98	42.62	40.62	46.07
RSC [29]	56.26	39.59	47.13	47.66
ADA [13]	58.72	45.58	48.26	50.85
ME-ADA [101]	58.96	**51.05**	58.42	51.00
L2D (AN) [93]	56.26	51.04	58.42	55.24
MetaCNN [90]	54.05	53.58	63.88	57.17
Ours (AN+P)	52.46	50.29	**66.79**	56.52
Ours (AN+M)	57.54	46.89	64.93	56.45
Ours (AN+MP)	**58.96**	45.86	64.57	56.46
L2D (RN)	68.41	43.56	48.84	53.60
L2D (RN+M)	57.57	**50.09**	65.51	**57.72**
Ours (RN+M)	58.25	47.35	**67.81**	57.80
Ours (RN+P)	58.42	48.29	66.68	57.80
Ours (RN+MP)	64.06	42.06	**73.98**	60.03

was set as 0.1. The values were tuned such that the weighted losses (i.e wL) are situated in a similar range.

5.3 Experimental Results and Analysis

Here we present experimental results over the three benchmark datasets and examination of domain gaps and the effect of PROF.

5.3.1 Experimental Results

Image experiment with PACS The aim of the PACS experiment is to show that PROF functions as a stable regularizer for sDG, reducing the mid-train OOD fluctuation reported in conventional *augment and align* methods. The results of the PACS experiment

Table 5.2: sDG accuracy on Corrupted CIFAR-10.

Method	W	B	N	D	Avg.
ERM [40]	67.28	56.73	30.02	62.30	54.08
CCSA [57]	67.66	57.81	28.73	61.96	54.04
d-SNE [97]	67.90	56.59	33.97	61.83	55.07
M-ADA [68]	75.54	63.76	54.21	65.10	64.65
L2D [93]	75.98	69.16	73.29	72.02	72.61
MetaCNN [90]	77.44	76.80	**78.23**	**81.26**	**78.45**
Ours M	**77.10**	**76.35**	67.94	76.57	**74.49**
Ours P	72.61	70.30	54.26	71.97	67.28

are reported in Table 5.1 where AN, RN, M, and P stands for AlexNet, ResNet, MDAR, and PROF, respectively.

First, we compare the downstream task accuracy. Training Alexnet with PROF (4.9) showed results close to the current SOTA [90] without the use of alignment. Furthermore, we showed state-of-the-art performance in Sketch domain, where domain gap is considered to be the largest.

Our *augment and align* baseline using MDAR also showed an accuracy close to SOTA. However, we observe that the method using MDAR displays a fluctuation of OOD performance after a certain point (i.e. $K > 5$). The behavior worsened as training continues. More importantly, training with PROF resulted in stabilization of the OOD performance, mitigating fluctuations, quantified as the reduction in variance across the target domain accuracy in $K > 5$ (Art: 3.39→1.27, Cartoon: 5.22→2.49, Sketch: 7.23→5.30). The display of mid-train OOD stabilization is depicted in Fig. 5.4. The phenomenon of OOD performance fluctuation was discussed earlier [48, 68, 93] but only using the Digits dataset. Finally, we show the competitiveness of our baseline with MDAR. We applied MDAR to an existing sDG method [93] by replacing InfoNCE loss [61] with MDAR. We observe a wide improvement over the conventional methods under certain conditions, as recorded in the last rows of Table 5.1.

Table 5.3: sDG accuracy on Digits.

Method	SVHN	M-M	S-D	USPS	Avg.
ERM [40]	27.83	52.72	39.65	76.94	49.29
d-SNE [97]	26.22	50.98	37.83	93.16	52.05
JiGen [4]	33.80	57.80	43.79	77.15	53.14
M-ADA [68]	42.55	67.94	48.95	78.53	59.49
ME-ADA [102]	42.56	63.27	50.39	81.04	59.32
L2D [93]	62.86	87.30	63.72	83.97	74.46
PDEN [48]	62.21	82.20	69.39	85.26	74.77
MetaCNN [90]	66.50	**88.27**	70.66	**89.64**	78.76
Ours M	**68.29**	81.88	**76.24**	88.79	**78.80**

Image experiment with Corrupted CIFAR-10 We present results over CIFAR-10-C (Table 5.2) where we compare the effectiveness of (1) conventional *augment and align* method (PROF) and (2) PROF under *small* domain shifts. We report the average accuracy (%) of each corruption category (Weather, Blur, Noise, Digits) [23], and the average accuracy of all categories.

Our method using MDAR marked scores close to the current SOTA [90] in two categories W (Weather) and B (Blur) while falling behind in others N (Noise) and D (Digital). We report that the OOD performance of the CIFAR-10-C is greatly affected by the design of the domain simulator G. The generator architecture will be On the contrary, our method using PROF marked results lower than our baseline MDAR. This is anticipated as we view the domain gap to be small between different datasets in the corrupted CIFAR-10, whereas PROF is designed for use under large domain discrepancies.

Digits experiment with MNIST The aim of the Digits experiment is too show that our new alignment objective is a strong baseline to compare with PROF. We share our results on the digit experiment on Table 5.3. Our method using MDAR and our updated domain simulation method outcompeted state-of-the-art records. In SVHN and SYNDIGIT (S-D), we show impressive improvement, while results in MNIST-M (M-

M) show slight deficiency. For effective comparison with existing methods, we refrain from using any form of manual data augmentation. We find that in Digits, increasing the number of simulated domains (K) helps OOD generalization. Our method with MDAR benefited from long training ($K > 100$).

Experiment on domain gaps We show results that display a strong correlation between the level of domain gap and the magnitude of mid-train fluctuation. In the Digits benchmark Sec. 5.1, it is commonly viewed that the gap between the source (MNIST) and the target is greater in certain datasets (e.g., SVHN and SYN-DIGIT) over others (e.g., MNIST-M and USPS). For instance, the baseline OOD accuracy is much higher in some target domains as opposed to others, in the order of: USPS(76.94%) > MNIST-M(52.72%) > SYNDIGIT(39.65%) > SVHN(27.83%), as recorded in (Table 5.3) We elaborate the domain gap further in Sec. 5.3.2. Interestingly, in our baseline experiment using the conventional *augment and align* method, we find that the mid-train fluctuation follows the same order: USPS(1.211) < MNIST-M(1.1795) < SYNDIGIT(4.938) < SVHN(5.106), measured with the variance of the OOD accuracy after $K > 5$. The phenomenon occurs similarly on PACS (Table 5.1), where the baseline OOD accuracy order Art (54.43%), Cartoon (42.74%), and Sketch (42.02%) matches the order of the mid-train fluctuation: Art (3.39) Cartoon (5.22), and Sketch (7.23). We view that these results empirically support that domain gap is correlated with mid-train fluctuation.

Effect of PROF We study further the effect of PROF on OOD performance. Experimental results are illustrated in Fig. 5.4 (A, C, and S are from PACS and M and P from MDAR and PROF.) As reported in Sec. 5.3.1, we observe that PROF functions as an effective regularizer for sDG. On the other hand, using PROF showed a limited effect in boosting of OOD performance. In experiments performed with AlexNet, the increase in OOD performance was not significant. However, using the ResNet18 architecture, OOD performance on both Art and Sketch domains benefited from using PROF. Our

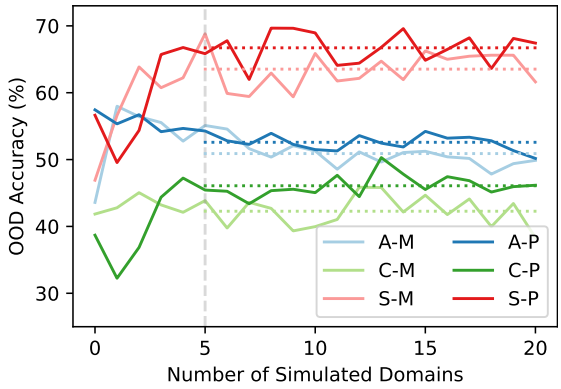


Figure 5.4: OOD accuracy (%) on PACS

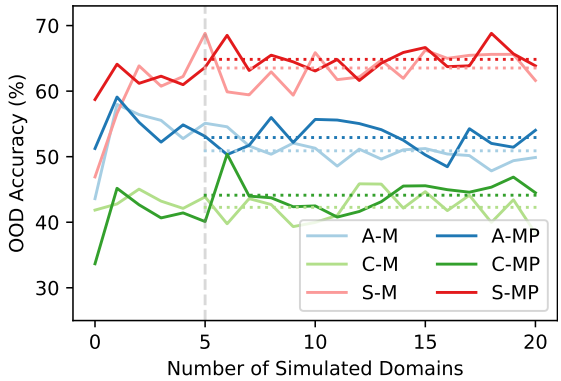


Figure 5.5: OOD accuracy (%) on PACS (MDAR + PROF)

hypothesis is that the model-parameter size and the depth of the task model affect the knowledge transfer capability, although further research is required.

A synergistic method: Combined use of MDAR and PROF In this section, we report the effect of using MDAR and PROF simultaneously. While PROF was designed for use without an alignment term (e.g., MDAR), we tested the effect of combining the two terms together. We observe that the synergistic method of PROF and MDAR triggered some differences in the training process.

Regarding the OOD accuracy, the synergistic method marked Art: 58.96%, Cartoon: 45.86%, Sketch: 64.57%, an average of 56.46% with AlexNet, as seen in Table 5.1.

While the accuracy is slightly higher than using MDAR alone (56.45%), we view that the synergistic method does not significantly benefit the OOD performance. In contrast, applying the synergistic method with a ResNet18 backbone showed a rise in OOD accuracy by a large gap 5.1. Further research is necessary to provide an understanding of this behavior as no definitive explanation currently exists, while our hypothesis is that the model architecture may have caused the phenomenon.

Regarding the mid-train OOD fluctuation, the synergistic method was not able to reduce fluctuations across Art and Cartoon, while reducing the fluctuation in Sketch. (Art: 3.39→4.50, Cartoon: 5.22→5.86, Sketch: 7.23→3.52) Similar to previous experiments, the mid-train OOD fluctuation was quantified with the variance across the target domain accuracy in $K > 5$. The mid-train OOD fluctuation of the synergistic method is depicted in Fig. 5.5. Our hypothesis is that the two terms may have disrupted each other, while a clear explanation for this phenomenon remains elusive. We believe that additional research is needed to produce an effective synergy of both methods.

Additional Experiments on PACS Here we present the results of additional experiments with the PACS benchmark.

Previous experiments on the PACS benchmark only used the Photo dataset as the source domain. In the following section, we report other cases where the source domain is changed (e.g., Art, Cartoon, Sketch). Here, we will denote each experiment as *Art as source*, *Cartoon as source* and *Sketch as source*, respectively. The results of the PACS experiment are reported in Table 5.4 where AN, M, and P stands for AlexNet, MDAR, and PROF, respectively. Each row in the table displays the source domain, backbone type, and the training method (M/P).

In Table 5.4, we report the sDG accuracy of our two methods, MDAR and PROF, on varying source domains. In cases where Art or Cartoon is used as source domain, training with our oracle regularization PROF marked higher OOD accuracy then its counterpart. On the other hand, PROF suffered when Sketch was set as the source

Table 5.4: sDG accuracy on PACS (Additional).

Method	P	A	C	S	Avg.
Source: Art					
Ours (Art+AN+P)	**78.07**	____	**66.04**	**63.15**	**69.09**
Ours (Art+AN+M)	77.53	____	59.39	60.04	65.65
Source: Cartoon					
Ours (Cartoon+AN+P)	64.57	**50.02**	____	**69.00**	**62.04**
Ours (Cartoon+AN+M)	65.20	47.10	____	65.81	59.37
Source: Sketch					
Ours (Sketch+AN+P)	46.25	44.31	61.60	____	50.72
Ours (Sketch+AN+M)	**48.03**	**47.83**	60.32	____	**52.06**

domain, falling behind the baseline MDAR. Our hypothesis is that this behavior is triggered by the inferior performance of the oracle. To elaborate, the oracle used on the *Sketch as source* experiment displayed low OOD accuracy on the target domains, unsuitable for effective oracle regularization (Photo: 51.61%, Art: 39.39%, Cartoon: 56.85%).

Next, we present the analysis on mid-train OOD fluctuation in each experimental configuration. When the source domain is set as Art, employing PROF resulted in yielded a stabilization of the OOD performance, effectively mitigating fluctuations. The fluctuation was quantified as the reduction in variance across the target domain accuracy in $K > 5$. When compared with the conventional *augment & align* method MDAR, our regularization method PROF displayed large reductions in variance (Photo: 1.71→1.17, Cartoon: 3.13→2.97, Sketch: 21.50→11.22). The mid-train OOD fluctuation when source is set as Art, is depicted in Fig. 5.6.

Similarly, when the source domain is configured as Cartoon, PROF displays similar stabilization of the mid-train OOD performance. Using PROF allows a reduction in fluctuation, measured as variance (Photo: 5.15 → 3.06 , Art: 5.00 → 3.07, Sketch: 0.70 → 3.91). We note that the stabilization effect in Sketch is relatively lower than that of other target domains, even lower than our *augment & align* baseline MDAR. The mid-train fluctuation is demonstrated in Fig. 5.7.

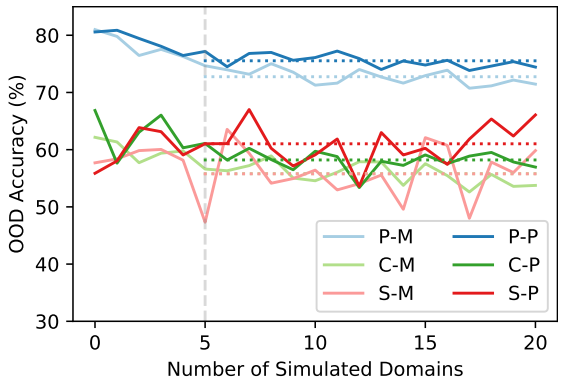


Figure 5.6: OOD accuracy (%) on PACS (Source: Art)

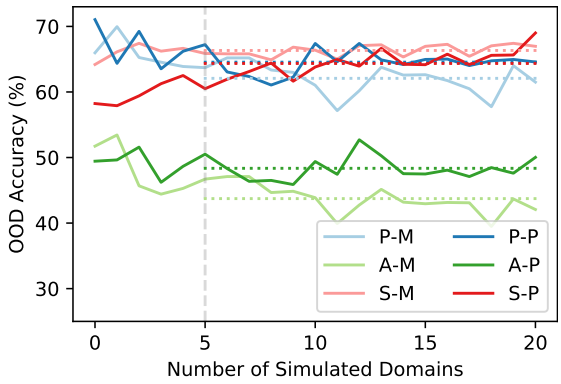


Figure 5.7: OOD accuracy (%) on PACS (Source: Cartoon)

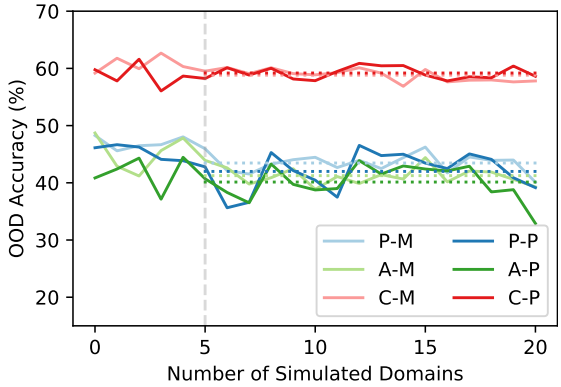


Figure 5.8: OOD accuracy (%) on PACS (Source: Sketch)

Lastly, we report the experimental results where the source was set as Sketch. In the *Sketch as source* experiment, we observe that PROF not only suffers in terms of performance, but also exhibits instability. PROF displayed high variance in mid-train performance when compared to the baseline (Photo: 2.46 $\rightarrow$ 10.41 , Art: 2.33 $\rightarrow$ 7.99, Cartoon: 1.01 $\rightarrow$ 1.04). The fluctuation is illustrated in Fig. 5.8. While a clear explanation is absent, we view that this phenomenon is caused by the under-performance of the oracle in the *Sketch as source* experiment. This result, displays a clear example of the problems associated with the obstacles regarding the oracle, where obtaining an oracle may not be readily available. We further discuss the issue with oracles in the following section, Sec. 5.3.2

Study of Hyperparameters We explore our method's sensitivity to hyperparameters. (λ_{PROF}): λ_{PROF} is the hyperparameter used for PROF that functions as the balancing weight of the two functions in Eq. (4.2). We begin with the value introduced in the original paper of [99] with $\lambda_{\text{PROF}} = 0.005$, and an alternate value $\frac{1}{d}$ introduced in Tsai et al. [86] where d is the length of a vector in $\mathcal{D}$ (distillation head output space). We observe that our method is not sensitive to the switch between two candidate values of λ_{PROF} although we cannot guarantee they are optimal. (λ_{MDAR} and λ_{adv}): The study on λ_{MDAR} and λ_{adv} is processed similar to λ_{PROF}. We find that switching between $\lambda = 0.005$ and $\frac{1}{p}$ has no significant impact on the learning process, where is p the length of a vector in $\mathcal{P}$ (projection head output space). While we cannot guarantee an optimal value. (w_{adv}, w_{cyc}, w_{div}): We optimize the hyperparameters w_{adv}, w_{cyc}, w_{div} using grid search. We find that as long as the weight-multiplied loss (wL) is situated on $(0, 1)$ range, there is no significant impact on performance.

Computing Resource We conducted experiments using NVIDIA RTX A6000 GPU devices for our experiments. For the Digits experiment, we trained approximately 4 hours using 1 GPU. For the CIFAR-10-C, we trained about 2,000 minutes using 2 GPUs. For the PACS experiment with PROF, we trained with 4 GPUs for about 43 hours. To

reproduce all the experiments, we expect 2–3 weeks.

5.3.2 Analysis

Here, we provide further analysis regarding the domain gap and the oracle.

On Domain Gaps In previous works, there exist different mentions regarding the domain gap within the experimental datasets. Here, we analyze such views.

There are contradicting views on the domain gap within the PACS dataset, the authors of Wan et al. [90] view that the domain gap is significant between the Art domain and the source domain (Photo), while relatively smaller with the Sketch and Cartoon domain. In contrast, Wang et al. [93] viewed that the domain gap is the largest between the source and the Sketch domain, due to its vastly abstracted shapes. On the contrary, there exists a shared consensus regarding the domain gap within corrupted CIFAR-10 dataset, where researchers view that the domain gap between the source (CIFAR-10) and the target (corruption datasets) is defined by the severity level of the corruption [48, 68, 90, 93]. Concerning the Digits dataset, the authors of Li et al. [48], Qiao et al. [68], Wang et al. [93] view that USPS displays the smallest domain gap with the source (MNIST). This is very similar to the view of Wan et al. [90] that USPS and SYNDIGIT datasets are closer to the source, while there is a large domain gap between the MNIST-M and the source domain.

In our paper, we used a different measure to observe the domain gap between datasets: the OOD classification accuracy on unseen domains. Our view on domain discrepancy is that it can be indirectly observed through the downstream task performance. This is closely tied to realistic settings, where task performance is the leading motive behind the study of sDG. The method is simple: using a fixed model, we train the model with the train split of the source domain. Then, using the trained model, we test the classification accuracy on unseen domains. We reported the results in Sec. 5.3.1. Using the baseline OOD accuracy as a measure for domain gap matches the view of

many existing works, while differences exist. For instance, USPS displays the highest OOD accuracy, matching the view of previous works that USPS shows the smallest discrepancy with the source [48, 68, 90, 93]. In PACS, the Sketch domain displays the lowest baseline OOD accuracy, which is in line with the view of some previous works [93], while different from others Wan et al. [90].

On Oracles In this section, we discuss the implementation of the oracle using pretrained models. Using pretrained models for OOD generalization is not an entirely novel idea [5, 49], but first for the task of sDG.

We selected the pretrained RegNetY-16GF as an oracle for PACS. In Cha et al. [5], a pretrained RegNetY-16GF model displayed high MI with the true oracle, a model that is trained on all source and target domains). The authors reported that the true oracle displayed an average validation accuracy of 98.4% on all PACS domains.

Similar to this, our implementation of the oracle with a pretrained RegNetY-16GF finetuned on the source domain (i.e. Photo) displayed high validation accuracies across all target domains. The finetuned RegNetY-16GF marked 75.16%, 75.30%, 69.00% on Art, Cartoon, Sketch, and an average validation accuracy of 73.15. While the average accuracy is lower than the true oracle in Cha et al. [5], this is an expected behavior as our oracle only uses the Photo domain, while the true oracle in [5] utilized all four domains.

However, using the RegNetY-16GF to implement the oracle for the Corrupted CIFAR-10 experiment was not satisfactory. When finetuned with the source domain (i.e. CIFAR-10), RegNetY-16GF marked low validation accuracy in the target domain with an average of 60.65%. This is similar for the implementation with ResNet50, which marked an average accuracy of 61.25% on the target domains, performing worse than the task model. We view that this phenomenon is derived from the difference between the two datasets. To elaborate, PACS is a collection of non-corrupted images, while Corrupted CIFAR-10 is a dataset generated by distorting CIFAR-10. As the RegNetY-

16GF is not specifically trained to withstand distortions, its performance decrease in Corrupted CIFAR-10 is understandable.

Chapter 6

Conclusion

This paper presents PROF (Progressive mutual information Regularization for Online distillation of Frozen oracles), a novel regularizer to address single source domain generalization under large domain discrepancy. Throughout the paper, we underscore the vulnerability of learning robustness via augmentation, which is observed as large fluctuations in the OOD performance during the training process. To mitigate this issue, PROF leverages pretrained oracles to guide the model to learn features that are less domain-specific, via maximization of the feature-level mutual information between the learning model and the oracle. Experiments on the PACS dataset demonstrate that PROF can stabilize the fluctuations associated with large domain gaps. We further introduce a strong baseline method with MDAR for a fair comparison with PROF. Training with MDAR showed State of the Art performance in Digits, and displayed a boost in performance when applied to existing methods.

Limitations PROF leverages pretrained models under the hypothesis that it can approximate an oracle that can generalize to all domains. As displayed in previous studies [5, 49], RegNetY-16GF sufficiently works as an oracle for the PACS benchmark. However, the same model does not fit well to the Digits benchmark. Due to the large gap between the pretrained dataset of the RegNetY-16GF and the Digit classification dataset.

This issue can be explained with the work of Wolpert and Macready [95], where the authors demonstrate that there exists a trade-off between a model's performance on a certain task and the performance on all remaining tasks. We believe further research is necessary.

Bibliography

[1] R. Adriana, B. Nicolas, K. S. Ebrahimi, C. Antoine, G. Carlo, and B. Yoshua. Fitnets: Hints for thin deep nets. *International Conference on Learning Representations*, 2, 2015.

[2] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[3] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization, 2019.

[4] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[5] J. Cha, K. Lee, S. Park, and S. Chun. Domain Generalization by Mutual-Information Regularization with Pre-trained Models. *arXiv e-prints*, art. arXiv:2203.10789, Mar. 2022. doi: 10.48550/arXiv.2203.10789.

[6] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994. ISSN 0165-1684. doi: https://doi.org/10.1016/0165-1684(94)90029-9. URL `https://www.sciencedirect.com/science/article/pii/0165168494900299`. Higher Order Statistics.

[7] I. Cugu, M. Mancini, Y. Chen, and Z. Akata. Attention consistency on visual corruptions for single-source domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4165–4174, June 2022.

[8] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[9] Y. Ding, L. Wang, B. Liang, S. Liang, Y. Wang, and F. Chen. Domain generalization by learning and removing domain-specific features. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=37Rf7BTAtAM.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[11] F. Eberhardt and R. Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007. ISSN 00318248, 1539767X. URL http://www.jstor.org/stable/10.1086/525638.

[12] D. Falbel. *torchvision: Models, Datasets and Transformations for Images*, 2023. https://torchvision.mlverse.org, https://github.com/mlverse/torchvision.

[13] X. Fan, Q. Wang, J. Ke, F. Yang, B. Gong, and M. Zhou. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8208–8217, 2021.

[14] A. Gabbay and Y. Hoshen. Demystifying inter-class disentanglement. In *International Conference on Learning Representations (ICLR)*, 2020.

[15] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015.

[16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research 17 (2016) 1-35*, 2015.

[17] B. Gao, H. Gouk, Y. Yang, and T. Hospedales. Loss function learning for domain generalization by implicit gradient. In *International Conference on Machine Learning*, pages 7002–7016. PMLR, 2022.

[18] L. Gresele, J. Von Kügelgen, V. Stimper, B. Schölkopf, and M. Besserve. Independent mechanism analysis, a new concept? *Advances in neural information processing systems*, 34:28233–28248, 2021.

[19] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=lQdXeXDoWtI`.

[20] H. Hälvä, S. Le Corff, L. Lehéricy, J. So, Y. Zhu, E. Gassiat, and A. Hyvarinen. Disentangling identifiable features from noisy data with structured nonlinear ica. *Advances in Neural Information Processing Systems*, 34:1624–1633, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.90.

[22] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[23] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[24] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[25] D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019.

[26] K. H. Huang, P. Orbanz, and M. Austern. Quantifying the effects of data augmentation, 2022.

[27] W. Huang, M. Yi, and X. Zhao. Towards the generalization of contrastive self-supervised learning, 2021.

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

[29] Z. Huang, H. Wang, E. P. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020.

[30] A. Hyvarinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

URL https://proceedings.neurips.cc/paper_files/paper/2016/file/d305281faf947ca7acade9ad5c8c818c-Paper.pdf.

[31] A. Hyvarinen, H. Sasaki, and R. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

[32] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(98)00140-3. URL https://www.sciencedirect.com/science/article/pii/S0893608098001403.

[33] M. Ilse, J. M. Tomczak, and P. Forré. Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pages 4555–4562. PMLR, 2021.

[34] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[35] J. N. Kaur, E. Kiciman, and A. Sharma. Modeling the data-generating process is necessary for out-of-distribution generalization. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL https://openreview.net/forum?id=KfB7QnuseT9.

[36] I. Khemakhem, D. P. Kingma, R. P. Monti, and A. Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. *AISTATS*, 2019.

[37] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[38] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[39] D. A. Klindt, L. Schott, Y. Sharma, I. Ustyuzhaninov, W. Brendel, M. Bethge, and D. Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EbIDjBynYJ8.

[40] V. Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: École d'Été de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Berlin Heidelberg, 01 2011. ISBN 978-3-642-22146-0. doi: 10.1007/978-3-642-22147-7.

[41] L. Kong, S. Xie, W. Yao, Y. Zheng, G. Chen, P. Stojanov, V. Akinwande, and K. Zhang. Partial disentanglement for domain adaptation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11455–11472. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kong22a.html.

[42] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira,

C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[44] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

[45] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[46] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, NIPS'89, page 396–404, Cambridge, MA, USA, 1989. MIT Press.

[47] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[48] L. Li, K. Gao, J. Cao, Z. Huang, Y. Weng, X. Mi, Z. Yu, X. Li, and B. Xia. Progressive domain expansion network for single domain generalization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–233, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.00029. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.00029.

[49] Z. Li, K. Ren, X. JIANG, Y. Shen, H. Zhang, and D. Li. SIMPLE: Specialized

model-sample matching for domain generalization. In *The Eleventh International Conference on Learning Representations*, 2023.

[50] C. Lin, Z. Yuan, S. Zhao, P. Sun, C. Wang, and J. Cai. Domain-invariant disentangled network for generalizable object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8751–8760, 2021. doi: 10.1109/ICCV48922.2021.00865.

[51] C. Liu, X. Sun, J. Wang, H. Tang, T. Li, T. Qin, W. Chen, and T.-Y. Liu. Learning causal semantic representation for out-of-distribution prediction. *Advances in Neural Information Processing Systems*, 34:6155–6170, 2021.

[52] Q. Liu, C. Chen, Q. Dou, and P.-A. Heng. Single-domain generalization in medical image segmentation via test-time adaptation from shape dictionary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1756–1764, 2022.

[53] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *ICML*, 2018.

[54] D. Mahajan, S. Tople, and A. Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[55] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper_files/paper/2016/file/ef0917ea498b1665ad6c701057155abe-Paper.pdf`.

[56] J. Mitrovic, B. McWilliams, J. C. Walker, L. H. Buesing, and C. Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9p2ekP904Rs.

[57] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[58] H. Nam and H.-E. Kim. Batch-instance normalization for adaptively style-invariant neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[59] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[60] Z. Niu, J. Yuan, X. Ma, Y. Xu, J. Liu, Y.-W. Chen, R. Tong, and L. Lin. Knowledge distillation-based domain-invariant representation learning for domain generalization. *IEEE Transactions on Multimedia*, pages 1–11, 2023. doi: 10.1109/TMM.2023.3263549.

[61] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2018.

[62] C. Ouyang, C. Chen, S. Li, Z. Li, C. Qin, W. Bai, and D. Rueckert. Causality-inspired single-source domain generalization for medical image segmentation. *IEEE Transactions on Medical Imaging*, 2022.

[63] L. Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15 (6):1191–1253, jun 2003. ISSN 0899-7667. doi: 10.1162/089976603321780272.

[64] G. Parascandolo, N. Kilbertus, M. Rojas-Carulla, and B. Schölkopf. Learning independent causal mechanisms. *ICML*, 2017.

[65] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

[66] J. Peters, D. Janzing, and B. Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017. ISBN 0262037319.

[67] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[68] F. Qiao, L. Zhao, and X. Peng. Learning to learn single domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12556–12565, 2020.

[69] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[70] P. Reizinger, L. Gresele, J. Brady, J. von Kügelgen, D. Zietlow, B. Schölkopf, G. Martius, W. Brendel, and M. Besserve. Embrace the gap: Vaes perform independent mechanism analysis, 2022.

[71] X. Ren, T. Yang, Y. Wang, and W. Zeng. Rethinking content and style: Exploring bias for unsupervised disentanglement. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1823–1832, 2021. doi: 10.1109/ICCVW54120.2021.00209.

[72] X. Ren, T. Yang, Y. Wang, and W. Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *ICLR*, 2022.

[73] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1): 1309–1342, 2018.

[74] E. Rosenfeld, P. K. Ravikumar, and A. Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=BbNIbVPJ-42`.

[75] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[76] O. Sener and V. Koltun. Domain generalization without excess empirical risk. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[77] X. Shen, F. Liu, H. Dong, Q. Lian, Z. Chen, and T. Zhang. Weakly supervised disentangled generative causal representation learning. *Journal of Machine Learning Research*, 2020.

[78] A. Shrivastava, Y. Qi, and V. Ordonez. Estimating and maximizing mutual information for knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 48–57, 2023.

[79] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019.

[80] M. Singh, L. Gustafson, A. Adcock, V. de Freitas Reis, B. Gedik, R. P. Kosaraju, D. Mahajan, R. Girshick, P. Dollár, and L. Van Der Maaten. Revisiting weakly supervised pre-training of visual perception models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 804–814, 2022.

[81] Z. Su, K. Yao, X. Yang, Q. Wang, J. Sun, and K. Huang. Rethinking data augmentation for single-source domain generalization in medical image segmentation, 2022.

[82] A. Szabó, Q. Hu, T. Portenier, M. Zwicker, and P. Favaro. Challenges in disentangling independent factors of variation, 2017.

[83] C. Tao, H. Wang, X. Zhu, J. Dong, S. Song, G. Huang, and J. Dai. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14411–14420, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01403.

[84] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

[85] N. Tomasev, I. Bica, B. McWilliams, L. Buesing, R. Pascanu, C. Blundell, and J. Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet?, 2022.

[86] Y.-H. H. Tsai, S. Bai, L.-P. Morency, and R. Salakhutdinov. A note on connecting barlow twins with negative-sample-free contrastive learning, 2021.

[87] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016.

[88] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[89] J. Von Kügelgen, Y. Sharma, L. Gresele, W. Brendel, B. Schölkopf, M. Besserve, and F. Locatello. Self-supervised learning with data augmentations provably

isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

[90] C. Wan, X. Shen, Y. Zhang, Z. Yin, X. Tian, F. Gao, J. Huang, and X.-S. Hua. Meta convolutional neural networks for single domain generalization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4672–4681, 2022. doi: 10.1109/CVPR52688.2022.00464.

[91] Y. Wang, H. Li, L.-p. Chau, and A. C. Kot. Embracing the dark knowledge: Domain generalization using regularized knowledge distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2595–2604, 2021.

[92] Z. Wang and V. Veitch. A unified causal view of domain invariant representation learning. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL `https://openreview.net/forum?id=-l9cpeEYwJJ`.

[93] Z. Wang, Y. Luo, R. Qiu, Z. Huang, and M. Baktashmotlagh. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 834–843, October 2021.

[94] O. Wiles, S. Gowal, F. Stimberg, S.-A. Rebuffi, I. Ktena, K. D. Dvijotham, and A. T. Cemgil. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2021.

[95] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. doi: 10.1109/4235.585893.

[96] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, et al. Robust fine-tuning of

zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022.

[97] X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *CVPR 2019*, 2019.

[98] S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

[99] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[100] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing. Towards principled disentanglement for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8024–8034, 2022.

[101] L. Zhao, T. Liu, X. Peng, and D. Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[102] L. Zhao, T. Liu, X. Peng, and D. Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020.

[103] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[104] R. S. Zimmermann, Y. Sharma, S. Schneider, M. Bethge, and W. Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

초 록

일반적으로, 머신러닝 모델은 데이터 분포 상의 변화에 취약한 경향을 보인다. 단일 소스 도메인 일반화(sDG)는 이러한 문제를 해결하기 위해 고안된 연구 태스크로, 인공적으로 설정된 분포 변화에도 강건한 모델을 만드는 것을 목표로 한다. 기존의 sDG 연구는 다양한 데이터 증강 기법을 통해 모델의 일반화 성능을 향상시키는 데 집중하였으나, 이와 같은 증강 기반 접근법의 유효성은 깊이 논의되지 않았다. 본 논문은 최근 Von Kügelgen et al. [89]의 연구결과를 이용하여 기존에 간과된 증강 기반 접근법의 문제들을 인과적 관점에서 조명하고, 그에 대한 해결책을 탐구한다. 본 연구진은 증강 기반 sDG 방식의 불안정성을 해소하기 위한 "PROF: 상호 정보 정규화를 통한 지식 증류 기반 단일 소스 도메인 일반화" 기법을 제시한다. PROF는 선학습된 모델의 지식을 이용한 증류 기반의 정규화 기법을 통해 모델의 훈련 과정을 지도한다. PROF는 증강 기반 sDG 방식에 추가되어, 모델의 일반화 성능이 안정적으로 증가할 수 있도록 한다. 나아가, 본 논문은 기존 방식에 비해 경제적인 정렬 함수와 개선된 데이터 증강 방식을 제안하였다.

주요어: 인과적 표상 학습, 도메인 정규화

학번: 2021-20711