Learning to ignore: Single Source Domain Generalization via Oracle Regularization

Dongkyu Cho
Causality Lab, Graduate School of Data Science
Seoul National University
Seoul, South Korea
kulupapa1127@snu.ac.kr

Sanghack Lee
Causality Lab, Graduate School of Data Science
Seoul National University
Seoul, South Korea
sanghack@snu.ac.kr

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 augmented samples. However, prior works frequently overlooked what is learned from such alignment. In this paper, we study the effectiveness of augmentation-based sDG methods via a causal interpretation of the data generating process. We highlight issues in using augmentation for generalization, namely, the distinction between domain invariance and augmentation invariance. To alleviate these issues, we introduce a novel regularization method that leverages pretrained models to guide the learning process via a feature-level regularization, which we name PROF (Progressive mutual information Regularization for Online distillation of Frozen oracles). PROF can be applied to conventional augmentation-based methods to moderate the impact of stochasticity in models repeatedly trained on augmented data, encouraging the model to learn domain-invariant representations. We empirically show that PROF stabilizes the learning process for sDG.

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 [1]. The issue is easily observable when a model trained in a source domain suffers in a target domain that is inconsistent with the source. Single-source Domain Generalization is a task devised to test a model's robustness under domain shift, where the model is given a single labeled dataset at train time and tested across multiple unseen domains. The absence of additional source domains makes sDG challenging, mainly because 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 [2] and learn domain-invariant features through an alignment of the generated domains using self-supervised contrastive loss [3] (hereinafter contrastive loss).

Accepted to the NeurIPS 2023 Workshop on Causal Representation Learning.

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 [4, 5]. 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. [6] 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 [7]. Finally, we devise a regularization method (PROF) under the assumption that generalized oracles can extract domain-invariant representations.

We state our contributions as follows. (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) We empirically show that augmentation-based sDG methods display large fluctuations in OOD performance across various datasets (3) To mitigate the issues brought by the aforementioned obstacles, we introduce a causality-inspired regularization method PROF for sDG, and experimentally display its effectiveness in stabilizing the learning process.

2 Limitations of Augmentation for sDG

In this section, we reveal an overlooked problem of augmentation-based sDG methods. Specifically, we revisit works on S-C disentanglement to analyze the validity of utilizing augmentation for sDG.

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:

$$L := L_{ce} + L_{\text{MaxEnt}}(x, \bar{x}; \Phi). \tag{1}$$

where L_{ce} is 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, $\hat{\mathbf{y}}$ the softmax prediction of the model, and L_{MaxEnt} is an objective that simultaneously aligns the mapped representations $\Phi(x)$ and $\Phi(\bar{x})$ under entropy regularization, where Φ is a feature extractor. Commonly, contrastive loss is used as L_{MaxEnt}. Recently, Von Kügelgen et al. [6] showed that the optimization of a contrastive loss provably minimizes L_{MaxEnt}, learning Φ to extract 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 We illustrate the underlying data generating process (i.e., DGP) using a causal graph and incorporate data augmentation into the causal graph under the 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.

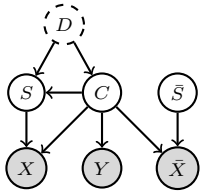


Figure 1: A causal diagram depicting DGP under data augmentation.

Now, we incorporate data augmentation into the picture. Given *label-preserving* augmentations, 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 Von Kügelgen et al. [6] for a detailed discussion). Yet, this does not imply that C and S are independent. C causally affects S (also corroborated by experimental results [8]). A way to understand this separation is by viewing such an augmentation as a soft intervention [9] 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. 1 (W is implicit) *excluding* D.

Von Kügelgen et al. [6] showed that, under certain conditions, the above DGP is sound, and augmentation separates C and S. However, the original 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. Note that C and S are defined by augmentation, not by the domain. Hence, unless the discrepancy between the source and target is moderate, optimizing solely the augment-and-align objective (Eq. 1) would be insufficient to address the issue caused by a large domain gap.

Learning to ignore To address a large domain shift, we begin with some observations. Conventional *augment and align* methods are vulnerable to domain shift in the sense that their effectiveness is 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), it is hard to approximate large domain shifts (e.g., PHOTO $\rightarrow$ SKETCH in PACS) (Appendix B.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 learning of domain-invariant features, while discouraging domain-specific features. Certainly, this requires a condition that the regularizer be an oracle that can distinguish domain-invariant information. Using this oracle regularizer, we aim to solve the phenomena associated with the large domain gap. Especially, the mid-training fluctuation of OOD performance, which was observed in earlier sDG works [10–12] but not discussed in-depth.[1] We view that the fluctuation is strongly correlated with the challenge of 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. 4.1. In the following section, we search for ways to implement the *hypothetical* oracle regularizer.

3 Leveraging Pretrained Models to Learn Domain Invariance

We present a novel sDG method where the aim is to mitigate the issue of mid-train fluctuation. While the principle of our approach is orthogonal to the type of data, in this paper, we focus on image data. The overview of our method is depicted in Fig. 3 (Appendix). At large, the method involves three neural networks, a domain generator G, task model classifier F, and an oracle O. We sequentially learn generators $\{G_k\}_{k=1}^{K}$ and use augmented samples created by the generators to train the task model F. Specifically, the generators provide *challenging* augmented samples to the task model, 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 (Eq. 2) and L_g (Eq. 10) are the loss used to train F and G, respectively, and $w_g \in \{0, 1\}$ controls the training of G.[2] The exact forms for L_f and L_g will become clear at the end of this section.

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 images, labels, and intermediate feature representations.

- **Task model:** The task model $F = C \circ H$ consists of a feature-extractor $H : \mathcal{X} \rightarrow \mathcal{H}$ and a classification head $C : \mathcal{H} \rightarrow \mathcal{Y}$.
- **Oracle:** The oracle model $O = C_o \circ H_o$ consists of a frozen feature-extractor $H_o : \mathcal{X} \rightarrow \mathcal{H}$ and a trainable classification head $C_o : \mathcal{H} \rightarrow \mathcal{Y}$. Task model F and oracle model 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. [3]
- **Generator:** A trainable generator $G : \mathcal{X} \rightarrow \mathcal{X}$ consists of an encoder-decoder architecture with a style-transfer module placed between the encoder and decoder.

[1] On the contrary, the phenomenon has been discussed in the multi-DG literature [13].

[2] Generally, $w_g = 1$ during the first half of the training epochs for G_k, then $w_g = 0$ to stop the training [11].

[3] For experimental purposes, we match the dimension of representation for the oracle and task model.

- **Distillation Head:** The distillation head $V : \mathcal{H} \to \mathcal{V}$ 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 V, following the analysis of Gupta et al. [14] on the efficacy of projection heads.
- **Projection Head:** Similarly, the projection head $P : \mathcal{H} \to \mathcal{Z}$ projects 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.

We train the task model F using a weighted combination of multiple losses, namely, the cross-entropy classification loss of x (L_{ce}) and $\bar{x}$ (L_{cls}; eq. (6)), with L_{PROF} and L_{MDAR} written as:

$$L_f = L_{ce}(C(H(x)), y) + L_{cls} + w_{\text{PROF}} \cdot L_{\text{PROF}} + w_{\text{MDAR}} \cdot L_{\text{MDAR}}, \tag{2}$$

where w_{PROF} and w_{MDAR} are user-set parameters to activate differing methods, PROF and MDAR. When training with the oracle regularizer (PROF) alone, w_{PROF} is non-zero while w_{MDAR} is set as 0. Vice versa, w_{PROF} is 0 in our baseline (MDAR). We explain losses for PROF and MDAR in the next sections.

3.1 Oracle Regularizer

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

$$L_{\text{PROF}}(x, \bar{x}, \lambda_{\text{PROF}}) = \sum\nolimits_{x' \in \{x, \bar{x}\}} \text{BT}(V(H(x')), V(H_o(x')), \lambda_{\text{PROF}}), \tag{3}$$

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)[15] is defined as:

$$\text{BT}(z, z^+, \lambda) = \textstyle\sum_i (1 - M_{ii})^2 \; + \lambda \sum_i \sum_{j \neq i} M_{ij}^2, \tag{4}$$

where M refers to the cross-correlation matrix of the two positive-pair feature representations z, z^+, and λ a user-set parameter.[4] BT (Eq. 4) is a feature-decorrelation loss originally introduced 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 two representations by spurring the diagonal values in 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 on the Regularization via MI Optimization 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 and H_o. PROF functions as a regularizer that guides the task model from deviating too far from the oracle, learning 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 $I(X; Y) = \mathbb{E}_{p(x,y)}[\log p(x \mid y)/p(x)]$ indicates the mutual information (MI). However, directly optimizing MI is challenging, as its exact estimation is intractable [16]. There exists InfoNCE loss [3] which adopts a lower bound of MI [17] as a surrogate objective for MI optimization:

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

However, an issue of InfoNCE as a variational bound of MI is that InfoNCE requires a large batch size for convergence [18, 19], making it doubtful for use in small datasets (e.g., PACS). Consequently, we indirectly approximate InfoNCE with a feature decorrelation loss [15], based on empirical and theoretical results that show its functional proximity [20, 21]. 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 may not be readily available. However, previous studies [22, 23] report that models pretrained from a large dataset or with deeper architectures tend to generalize better at unseen domains. Considering this, we utilize a model pretrained on a larger domain as oracle, and freeze the feature-extractor H_o to preserve its knowledge.

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

3.2 Multi-Domain Alignment with Redundancy Reduction

We now introduce a novel alignment objective MDAR (Multi-Domain Alignment with Redundancy reduction) for sDG. MDAR aims to disentangle latent features that are invariant across multiple augmented views. We design MDAR as a fair baseline of the conventional *augment and align* method. In learning the kth generator G_k, we create an augmented view $\bar{x}$ for a batch of original samples x using the kth generator G_k. We then randomly load two previously learned generators to construct two augmented views $\bar{x}'$ and $\bar{x}''$. With $\{x, \bar{x}, \bar{x}', \bar{x}''\}$, we encourage their representations vary in a similar way. Hence, we use BT (Eq. 4) over the representations for $\{x, \bar{x}, \bar{x}', \bar{x}''\}$ obtained through the projection head and feature extractor, $P \circ H$. That is, their cross-correlation matrix M to be closer to an identity matrix. Our alignment loss L_{MDAR} is written as:

$$L_{\text{MDAR}}(\mathbf{x} = \{x, \bar{x}, \bar{x}', \bar{x}''\}, \lambda_{\text{MDAR}}) = \sum\nolimits_{x_i \neq x_j} \text{BT}(P(H(x_i)), P(H(x_j)), \lambda_{\text{MDAR}}), \tag{5}$$

where λ_{MDAR} a user-set parameter. Intuitively, via optimizing L_{MDAR}, we can train the task model in a way that multiple views (representations) are aligned. In terms of S-C disentanglement, MDAR encourages the retrieval of augment-invariant features. Different from the commonly used InfoNCE loss, our objective (Eq. 5) does not require negative pairs, thus works well on small batch sizes [15, 24], suitable for benchmarks like PACS. In our conventional *augment and align* baseline experiment, we train our model with a variant of Eq. 2: $L_f = L_{ce}(C(H(x)), y) + L_{cls} + w_{\text{MDAR}} \cdot L_{\text{MDAR}}$.

3.3 Learnable Domain Shift Simulators

We sequentially train multiple generators to obtain varying simulated domains. The purpose of this process is to examine the behavior of models repeatedly trained on simulated domains, namely, the mid-train OOD fluctuation. To simulate domain shift, we must ensure that the augmented domain is label-preserved, while different from the source domain. Reflecting this, we adopt methods of Wang et al. [12], Li et al. [11] to assure the consistency of generated samples:

$$L_{cls}(\bar{x}, y) = L_{ce}(C(H(\bar{x})), y) + I(w_{\text{PROF}} > 0) \cdot L_{ce}(C_o(H_o(\bar{x})), y), \tag{6}$$

$$L_{cyc}(x, \bar{x}) = \|x - G_{cyc}(\bar{x})\|_2, \tag{7}$$

where I is an indicator function. L_{cls} is a cross-entropy loss that assures the validity of the generated samples $\bar{x}$ based on predictions from task model F (also from oracle O if PROF is employed.) L_{cyc} ensures that the output of G, can be recovered to the original input image when passed through the inversed generator G_{cyc} [25]. Next, we encourage the generator to create diverse augmentations with:

$$L_{div}(\bar{x}_1, \bar{x}_2) = -\|\bar{x}_1 - \bar{x}_2\|_2, \tag{8}$$

$$L_{adv}(x, \bar{x}, \lambda_{adv}) = -\text{BT}(P(H(x)), P(H(\bar{x})), \lambda_{adv}). \tag{9}$$

L_{div} is a negated L2-norm between two augmented views ($\bar{x}_1$,$\bar{x}_2$) of a batch x created with the generator. Intuitively, optimizing with L_{div} encourages the generator to augment diverse samples, preventing collapse. L_{adv} is an adversarial loss function designed to reverse the alignment process by negating the feature-decorrelation loss used in Eq. 4. We train the generator with the weighted sum L_g of the above four objectives (where L_{adv} is active only if MDAR is used.):

$$L_g = L_{cls} + w_{cyc} \cdot L_{cyc} + w_{div} \cdot L_{div} + I(w_{\text{MDAR}}{>}0) \cdot w_{adv} \cdot L_{adv}. \tag{10}$$

4 Experiment

Datasets & Implementation Following the experimental settings in prior sDG works, we adopted two broadly used benchmarks (e.g., PACS [30] and Digits) for our problem, along with an additional benchmark, Office-Home [31]. Details of the datasets are included in Appendix B.1. In all experiments, we utilized the identical network architectures used in previous sDG works, its details reported in Appendix B.2. Information regarding the pretraining process, training process, and training hyperparameters are reported in Appendix B.4, Appendix B.3, and Appendix B.5, respectively.

Table 1: sDG accuracy on PACS.

Method	A	C	S	Avg.
ERM [26]	54.43	42.74	42.02	46.39
ADA [27]	58.72	45.58	48.26	50.85
ME-ADA [28]	58.96	**51.05**	58.42	51.00
L2D (AN) [12]	56.26	51.04	58.42	55.24
MetaCNN [29]	54.05	53.58	63.88	57.17
Ours (P)	52.46	50.29	**66.79**	56.52
Ours (M)	57.54	46.89	64.93	56.45
Ours (MP)	**58.96**	45.86	64.57	56.46

4.1 Experimental Results and Analysis

Experiment with PACS PACS experiment aims 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 experimental results are reported in Table 1 where M and P stand for MDAR, and PROF. First, we compare the generalization accuracy. Training with PROF (Eq.(2)) showed results close to the current SoTA without additional process of alignment. Our *augment and align* baseline (MDAR) showed a similar accuracy, but displayed a fluctuation of OOD performance after a certain point (i.e. $K > 5$), escalating as training continued. On the contrary, 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 stabilization effect is depicted in Fig. 2(A, C, and S are from PACS). More PACS experiments are reported in Appendix A.1.

Table 2: sDG accuracy on Digits.

Method	SVHN	M-M	S-D	USPS	Avg.
ERM [26]	27.83	52.72	39.65	76.94	49.29
JiGen [32]	33.80	57.80	43.79	77.15	53.14
M-ADA [10]	42.55	67.94	48.95	78.53	59.49
L2D [12]	62.86	87.30	63.72	83.97	74.46
PDEN [11]	62.21	82.20	69.39	85.26	74.77
MetaCNN [29]	66.50	**88.27**	70.66	**89.64**	78.76
Ours M	**68.29**	81.88	**76.24**	88.79	**78.80**
Ours P	**74.50**	87.98	**78.67**	86.15	**81.82**

Experiment with Digits Digits experiment aims to display the efficacy of (PROF) and present the strength of our baseline (MDAR). We share the results on Table 2. We underline that in Digits, we could not obtain a pretrained model fit for use as oracle. Hence, we follow the practice of Cha et al. [22] and use a *true* oracle, a model pretrained on both the source and target domains. Our method with PROF showed a large drop in mid-train OOD fluctuation compared to the baseline (M-M: 2.56 → 1.17, USPS: 3.48 → 1.11, SVHN: 3.58 → 1.95, S-D: 2.36 → 2.10). The OOD stabilization effect is illustrated in Fig. 5 (Appendix A.2). Furthermore, PROF displays superior generalization accuracy.[5] Notably, our baseline (MDAR) surpassed SoTA records. The analysis continues in Appendix A.2.

Experiment with Office-Home Office-Home experiment reconfirms the effectiveness of PROF for mitigating the issues under large domain shifts. The results of the experiment is reported on Table 3. In terms of performance, regularizing with PROF displayed a strong advantage over the conventional baseline (MDAR). In terms of OOD fluctuation, PROF showed a stabilization of the OOD performance, measured as the reduction in variance across the target domain accuracy (Art: 10.63 → 8.23, Clipart: 2.17 → 2.05, Product: 7.46 → 6.41). The stabilization effect is illustrated in Fig. 6 (Appendix A.3). Detailed analysis of the Office-Home experiment is reported in Appendix A.3.

Table 3: sDG accuracy on Office-Home.

Method	Art	Clipart	Product	Avg.
ERM	52.78	40.19	68.73	53.90
Ours (M)	53.39	43.38	66.25	54.34
Ours (P)	**55.25**	**46.69**	**69.26**	**57.07**

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 Digits, it is commonly viewed that the gap between the source (MNIST) and the target is greater in certain datasets (e.g., SVHN and SYNDIGIT) 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 2. Interestingly, in our baseline experiment with MDAR, we find that the fluctuation magnitude follows the same order: USPS(1.211) < MNIST-M(1.1795) < SYNDIGIT(4.938) < SVHN(5.106), measured by the variance of the OOD accuracy after $K > 5$. A similar pattern is observed on PACS (Table 1), where the baseline OOD accuracy order A (54.43%), C (42.74%), and S (42.02%) matches the order of the mid-train fluctuation: A (3.39), C (5.22), and S (7.23). We view that these results empirically support the correlation between domain gap and mid-train fluctuation. We elaborate the domain gap further in Appendix C.

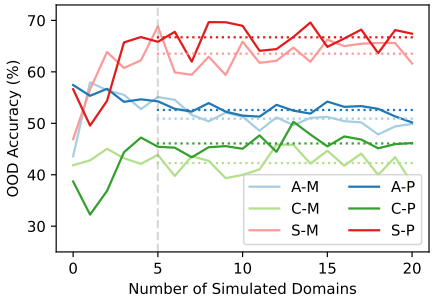


Figure 2: OOD accuracy (%) on PACS

Effect of PROF We study further the effect of PROF on OOD generalization. The stabilization effect of PROF is repeatedly confirmed across many benchmarks including PACS, Digits (Fig. 5), and Office-Home (Fig. 6). In real-world settings, a model with large fluctuation is unreliable since its performance may drop unknowingly. Hence, a reduction in fluctuation is closely synonymous with

[5] As we use the *true* oracle, performance boost is expectable [33]. Hence, we do not claim SoTA for PROF.

model consistency at test time. Furthermore, PROF boosts generalization accuracy. OOD accuracy benefited from using PROF in PACS (Table 4(Appendix A.1)) and Office-Home (Table 3). However, the gain was marginal in PACS with an AlexNet backbone Table 1. Our notion is that the model architecture (e.g., width and depth) affects the knowledge transfer, though further research is required.

5 Discussion

In this section, we discuss the limitations of our work and propose ideas for future work.

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 [22, 23], RegNetY-16GF sufficiently works as an oracle for the PACS benchmark. However, the same model does not fit well with the Digits benchmark. Due to the large gap between the pretrained dataset of the RegNetY-16GF and the Digits dataset. This issue can be explained with the work of Wolpert and Macready [34], in which the authors demonstrated that there exists a trade-off between a model's performance on a certain task and the performance on all remaining tasks.

Future Work As mentioned above, a critical limitation of our work is that the method relies on an external source of knowledge (i.e., Oracle) to regulate the learning process. Naturally, there are concerns that question the necessity of such regularization, suggesting the direct use of the oracle. In our defense, the direct implementation of oracles is discouraged in the sDG setting, as most works follow the same model selection criteria (e.g., AlexNet for PACS, 3-layer MLP for Digits). However, we agree that it is an important concern to address in future work. A possible suggestion is to regulate only the later layers, under the assumption that earlier layers contain general, domain-invariant information [35]. We believe further research is necessary to alleviate this issue.

6 Conclusion

This paper presents PROF, a novel oracle regularizer to address single source domain generalization under large domain discrepancy. 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 multiple datasets (PACS, Digits, Office-Home) 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.

Acknowledgement

This work was partly supported by IITP (2022-0-00953-PICA/50%) and NRF (RS-2023-00211904/50%) grant funded by the Korean government (MSIT).

References

[1] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[2] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[3] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

[4] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun 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.

[5] Aapo Hyvarinen and Hiroshi 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.

[6] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34: 16451–16467, 2021.

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

[8] David A. Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan 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.

[9] Frederick Eberhardt and Richard 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.

[10] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12556–12565, 2020.

[11] 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.

[12] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa 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.

[13] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022.

[14] Kartik Gupta, Thalaiyasingam Ajanthan, Anton van den Hengel, and Stephen Gould. Understanding and improving the role of projection head in self-supervised learning, 2022.

[15] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

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

[17] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[18] Aman Shrivastava, Yanjun Qi, and Vicente 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.

[19] Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019.

[20] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning, 2021.

[21] 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.

[22] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain Generalization by Mutual-Information Regularization with Pre-trained Models. *arXiv e-prints*, art. arXiv:2203.10789, March 2022. doi: 10.48550/arXiv.2203.10789.

[23] Ziyue Li, Kan Ren, XINYANG JIANG, Yifei Shen, Haipeng Zhang, and Dongsheng Li. SIMPLE: Specialized model-sample matching for domain generalization. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] Yao-Hung Hubert Tsai, Shaojie Bai, Louis-Philippe Morency, and Ruslan Salakhutdinov. A note on connecting barlow twins with negative-sample-free contrastive learning, 2021.

[25] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei 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.

[26] Vladimir 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.

[27] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan 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.

[28] Long Zhao, Ting Liu, Xi Peng, and Dimitris 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.

[29] Chaoqun Wan, Xu Shen, Yonggang Zhang, Zhiheng Yin, Xinmei Tian, Feng Gao, Jianqiang Huang, and Xian-Sheng 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.

[30] 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.

[31] 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.

[32] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[33] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent, 2022.

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

[35] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations*, 2022.

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

[37] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew 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`.

[38] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research 17 (2016) 1-35*, 2015.

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

[40] 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.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. 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`.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[43] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

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

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

[46] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016.

[47] Jivat Neet Kaur, Emre Kiciman, and Amit 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`.

[48] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2021.

[49] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

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

[51] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens 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.

[52] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann 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.

[53] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann 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.

[54] Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Challenges in disentangling independent factors of variation, 2017.

[55] Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017. ISBN 0262037319.

[56] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. *ICML*, 2017.

[57] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *ICML*, 2018.

[58] Luigi Gresele, Julius Von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. Independent mechanism analysis, a new concept? *Advances in neural information processing systems*, 34:28233–28248, 2021.

[59] Patrik Reizinger, Luigi Gresele, Jack Brady, Julius von Kügelgen, Dominik Zietlow, Bernhard Schölkopf, Georg Martius, Wieland Brendel, and Michel Besserve. Embrace the gap: Vaes perform independent mechanism analysis, 2022.

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

[61] Kevin H. Huang, Peter Orbanz, and Morgane Austern. Quantifying the effects of data augmentation, 2022.

[62] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2019.

[63] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[64] Zihao Wang and Victor 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.

[65] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok 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.

[66] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[67] Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and B Yoshua. Fitnets: Hints for thin deep nets. *International Conference on Learning Representations*, 2, 2015.

[68] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[69] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

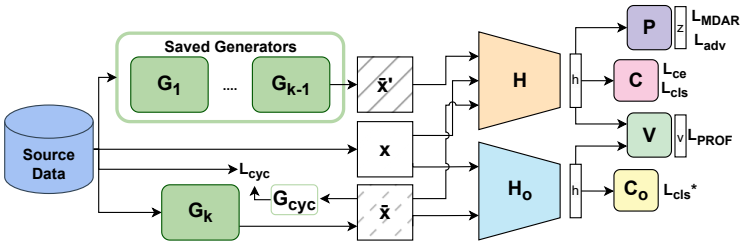


Figure 3: The illustration of our method. We sequentially train multiple generators $G_{1...K}$. The Oracle H_o regulates the task model H's learning process. During the training, multiple modules (e.g., P, V, C) are used for optimization.

A Experimental Results

A.1 Experiments on PACS (Continued)

Here we continue presenting the results of experiments with the PACS benchmark.

Next, we present experimental results where the backbone is switched from the default backbone AlexNet to ResNet. We applied MDAR to an existing sDG method [12] by replacing the InfoNCE loss with MDAR. We observe wide improvement over conventional methods under certain conditions, as recorded in the last rows of Table 4. Furthermore, synergistic methods that apply both PROF and MDAR displayed large improvements in the generalization performance. We will discuss further on the synergistic method further in Appendix A.4.

Table 4: sDG accuracy on PACS (ResNet).

Method	A	C	S	Avg.
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

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.

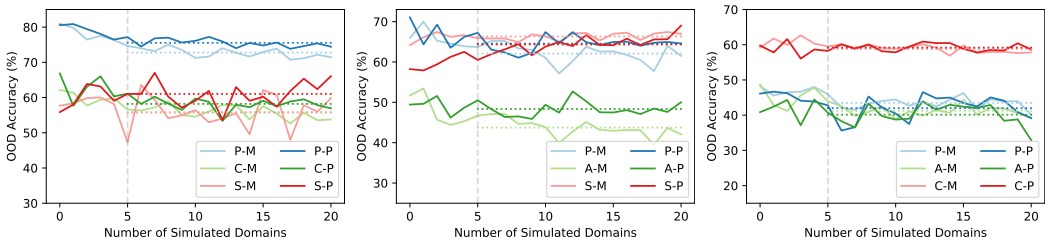
In Table 5, we report the sDG accuracy of our two methods, MDAR and PROF, where AN, M, and P stands for AlexNet, MDAR, and PROF, respectively. Each row in the table displays the source domain, backbone type, and the training method (M/P). In cases where Art or Cartoon is used as source domain, training 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 domain, falling behind the baseline MDAR. Our hypothesis is that this behavior is triggered by the subpar 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%).

Table 5: sDG accuracy on PACS (Full).

Method	P	A	C	S	Avg.
Source: Photo					
Ours (AN+P)	____	52.46	50.29	66.79	**56.52**
Ours (AN+M)	____	57.54	46.89	64.93	56.45
Source: Art					
Ours (AN+P)	**78.07**	____	**66.04**	**63.15**	**69.09**
Ours (AN+M)	77.53	____	59.39	60.04	65.65
Source: Cartoon					
Ours (AN+P)	64.57	**50.02**	____	**69.00**	**62.04**
Ours (AN+M)	65.20	47.10	____	65.81	59.37
Source: Sketch					
Ours (AN+P)	46.25	44.31	61.60	____	50.72
Ours (AN+M)	**48.03**	**47.83**	60.32	____	**52.06**

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. 4a.

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



(a) Source: Art (b) Source: Cartoon (c) Source: Sketch

Figure 4: OOD accuracy (%) on PACS (Additional)

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. 4b.

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. 4c. 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, Appendix D

A.2 Experimental Results on Digits (Continued)

Here we continue our analysis on the results of the Digits Experiment. In Sec. 4, we demonstrated that our regularization method PROF successfully mitigates issues of OOD fluctuation, measured as variance. This is illustrated in Fig. 5 (M and P are from MDAR and PROF.). One notable observation is the significant increase in OOD generalization accuracy (81.82) when using PROF, in Table 2. As mentioned in the footnote, we do not claim this score to be state-of-the-art, as the true oracle is used. From the perspective of knowledge distillation, this is anticipated as the true oracle is already generalized to the target domains. In comparison, the approximated oracle in PACS does not guarantee robustness in the target domains, despite its higher generalizability. This confirms that a gap between the approximated oracle and the true oracle exists, which is a limitation that we acknowledge. We provide further analysis on the oracle in Appendix D

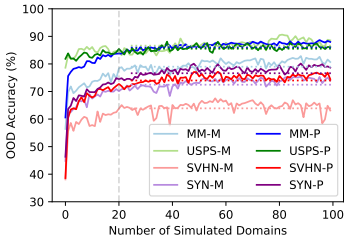


Figure 5: OOD accuracy (%) on Digits

Next, we discuss the results of our baseline experiment using MDAR. As mentioned in the main paper, our baseline surpassed state-of-the-art in Digits. In SVHN and SYNDIGIT (S-D), we show large improvement, while results in MNIST-M (M-M) show slight deficiency. Similar to 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. Both our baseline (MDAR) and PROF benefited from long training ($K > 100$).

A.3 Experimental Results on Office-Home (Continued)

Here we continue our analysis of the results of the Office-Home Experiment. The Office-Home benchmark is not commonly used in the sDG literature, but we include the benchmark to bring attention to an important question: Is augmentation reliable for sDG?

As described in Table 3, augmentation-based approaches do show a boost in OOD accuracy. However, the effect gradually disappears with a sharp decline in OOD accuracy, as depicted in Fig. 6. (A,

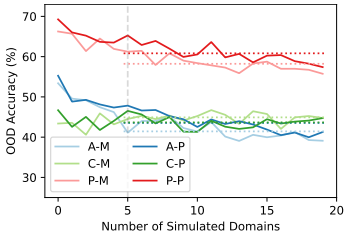


Figure 6: OOD accuracy (%) on Office-Home

C, and P are abbreviations of Art, Clipart, and Product domains, while M and P are from MDAR and PROF.) This downward trend is also spotted on other benchmarks, but not as intense.

We believe that this phenomenon aligns with our analysis of the uncertainty of utilizing augmentation for OOD generalization. Our hypothesis is that the distributional gap within the Office-Home benchmark may be more intense than conventional sDG benchmarks (e.g., Digits, PACS). The phenomenon brings novel questions on the efficacy of augmentation-based generalization methods. We believe that further research is required. Nonetheless, even in this case, PROF continues to stabilize the learning process, showing a smaller variance than our baseline (MDAR).

A.4 A Synergistic Approach: 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.

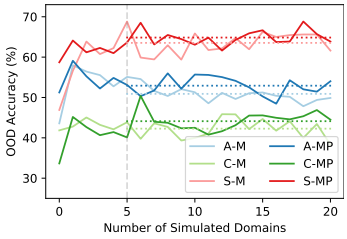


Figure 7: OOD accuracy (%) on PACS (MDAR + PROF)

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 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. On the other hand, applying the synergistic method with a ResNet18 backbone showed a rise in OOD accuracy by a large gap 4. 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 is depicted in Fig. 7 (A, C, and S are from PACS and M and MP from MDAR and MDAR+PROF, the synergistic method.). 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.

A.5 Study of Hyperparameters

We explore our method's sensitivity to hyperparameters. (λ_{PROF}): λ_{PROF} is the hyperparameter used for PROF that operates as the balancing weight of the two functions in Eq. (4). We begin with the value in the original paper of Zbontar et al. [15] with $\lambda_{\text{PROF}} = 0.005$, and an alternate value $\frac{1}{d}$ introduced in Tsai et al. [24] where d is the length of a vector in $\mathcal{D}$ (distillation head output space). We observe that our method is resilient 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}. Switching between $\lambda = 0.005$ and $\frac{1}{p}$ posed no notable impact on the learning process, where p is 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 the $(0, 1)$ range, there is no significant impact on performance.

B Implementation Detail

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

B.1 Datasets

Here, we elaborate on the datasets used in our experiments.

PACS [30] consists of 4 domains of differing styles (Photo, Art, Cartoon, and Sketch) with 7 classes. In default, we train our model with the Photo domain and evaluate the remaining target domains. We also present additional experiments in Appendix A.1. Among the selected benchmarks, PACS is the main target of PROF due to its large gap between domains.

Digits is comprised of 5 different digit classification datasets, MNIST [36], SVHN [37], MNIST-M [38], SYNDIGIT [39], USPS [40]. In our 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.

Office-Home [31] is a common benchmark for DG, but not for sDG. The benchmark consists of 4 datasets (Real-world, Art, Clipart, Product) with differing styles with 65 classes. We train on the Real-world domain and evaluate on the remaining domains.

B.2 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 [41], pretrained on ImageNet [42]. 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 Digits experiment is a multi-layer CNN network (i.e. conv-pool-conv-pool-fc-fc-softmax). 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. As the true oracle in Digits uses an identical network design as the task model, no buffer layer was used.

Lastly, The model used in the Office-Home experiment is a ResNet18 network. The ResNet is torchvision implemented, and pretrained on the ImageNet dataset. Similar to the PACS experiment, The buffer is a 2-layered MLP that maps the output dimension to that of the oracle (RegNetY-16GF), which is 3024. Hence, the final output dimension of the feature extractor is 3024.

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. 8. 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 [43]. 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 [43]. In Fig. 8, we can see that the STN performs spatial transformations, creating the modified image at the middle. An advantage of STN is that no additional requirements are needed for training the module.

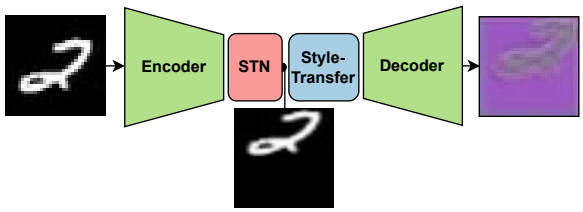
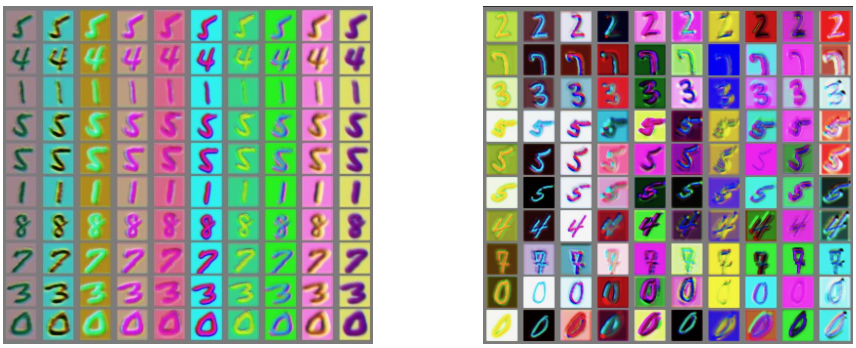


Figure 8: The illustration of the Generator.



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

Figure 9: The illustrated comparison of the generators.

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) [44]. 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 Huang and Belongie [45], where we switched the normalization method from Instance Normalization [46] to BIN for effective style transfer.

We share the results of applying these modifications in Fig. 9. Whilst previous augmentation methods [11, 12] were limited to manipulating certain attributes (e.g., color, stroke), our method further allows spatial manipulations (e.g., shape, location). For instance, in the right image of Fig. 9, we can observe that the images generated using our method displayed a large variance in shape, position, and color. This modification is inspired by recent studies on domain shift [47, 48], which revealed that domain shift occurs on a variety of levels. 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. 10, 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, a RegNetY-16GF for the PACS and Office-Home experiment, and a multi-layer CNN network for the Digits experiment.

The RegNetY-16GF is a variant of the RegNet family, a line of models introduced in [49] 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 capacity or complexity. We implement the model, and its model weights using the torchvision [50] library. We used the weights pretrained via end-to-end fine-tuning of the original SWAG [51] weights on the ImageNet-1K data [42]. 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 SGD optimizer and the Cosine Annealing learning rate scheduler, a batch size of 64. For the Office-Home, we fine-tuned the pretrained model with the Real World domain of Office-Home for 30 epochs, using the SGD optimizer and the Cosine Annealing scheduler, a batch size of 16.

For the PROF experiment in Digits, an identical network architecture was used for both the task model F and the true oracle model O. The true oracle was pretrained on the source and target domains

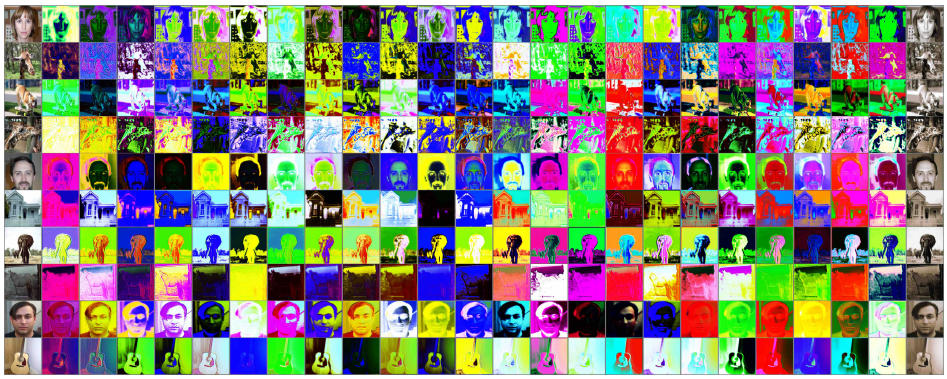


Figure 10: The illustration of generated images (PACS).

of Digits. The pretraining epochs were set as 100, with a learning rate of $1e-4$ using the Adam optimizer. The batch size was set as 256.

B.3 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 [52]. 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.

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, both the projection dimension ($\mathcal{Z}$) and the distillation head projection dimension ($\mathcal{D}$) were as 128.

Office-Home For the Office-Home experiment, we set K as 20, training each generator with 30 inner loops. During the first 15 inner loops we train the generator, and halted training for the remaining 15 loops. Similar to other cases, we set the number of epochs by analyzing the training behavior of the generators. The learning rate was set as $1e-4$, using the Adam optimizer. 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 512.

B.4 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 [30]. 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.

Digits 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.

Office-Home We pretrained the ResNet18 with the train split of the Real World domain. We pretrained the model for 100 epochs, with a learning rate of $1e-4$ using the SGD optimizer. We used no learning rate scheduler. The batch size was set as 64.

B.5 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. [15]. We tuned λ_{PROF} using experimental results of the original paper and [24]. 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 $\mathcal{D}$ (distillation head output space). 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. [11]. 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. Finally, w_{PROF} was set as 0.1. The values were tuned such that the weighted losses (i.e. wL) are situated in a similar range.

C On Domain Gaps

In previous works, there exist different mentions regarding the domain gap within the experimental datasets. We begin this section by comparing such views.

There are contradicting views on the domain gap within the PACS dataset, the authors of Wan et al. [29] 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. [12] viewed that the domain gap is the largest between the source and the Sketch domain, due to its vastly abstracted shapes. Concerning the Digits dataset, the authors of Qiao et al. [10], Wang et al. [12], Li et al. [11] view that USPS displays the smallest domain gap with the source domain (MNIST). This is very similar to the view of Wan et al. [29] 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. 4.1. 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 [10, 12, 11, 29]. In PACS,

the Sketch domain displays the lowest baseline OOD accuracy, which is in line with the view of some previous works [12], while different from the view of Wan et al. [29].

D 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 [23, 22], but first for the task of sDG.

We selected the pretrained RegNetY-16GF as an oracle for PACS. In Cha et al. [22], 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 in PACS, MNIST in Digits, Real World in Office-Home) displayed high validation accuracies across all target domains. To be specific, in PACS, 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. [22], this is an expected behavior as our oracle used only the Photo domain, while the true oracle in [22] utilized all four domains of PACS.

However, we empirically confirm that the RegNetY-16GF is not universally available for use as the oracle. For instance, using the RegNetY-16GF to implement the oracle for the Digits experiment was not satisfactory. When finetuned with the source domain (i.e. MNIST), RegNetY-16GF marked low validation accuracy in the target domain. We believe that this difference is derived from the difference between the two datasets. For instance, PACS is a collection of images without any distortion, while Digits is a dataset solely comprised of digit images. Hence, we view that the large gap between the pretrained dataset of the RegNetY-16GF and the Digit classification datasets is responsible for this behavior.

This issue can be explained with the work of Wolpert and Macready [34], 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 this to be a crucial limitation of our method, and aspire to investigate further.

E 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. While the term style and content originated from the style transfer literature [53, 54], recent works try to push the idea further using concepts of causal inference [7, 55, 56] and Independent Component Analysis (ICA) [57–59]. Notably, disentanglement is used to elucidate the underlying mechanism of data augmentation [6, 60, 61].

Domain Generalization In the multi-source domain generalization field, disentanglement of domain-invariant features has shown great success in training robust domain-agnostic models 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 [62–64]. 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 [2, 10–12, 29, 27]. 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. A recent movement in the multi-DG literature highlights the use of pretrained models for OOD generalization, leveraging the knowledge of the pretrained models [22, 65, 23]. Such works closely resemble the methods introduced in the Knowledge Distillation (KD) literature [66–68, 18, 69].