# PEER pressure: Model-to-Model Regularization for Single Source Domain Generalization

Dong Kyu Cho
New York University
New York, USA
dongkyu.cho@nyu.edu

Inwoo Hwang*†
Columbia University
New York, USA
ih2455@columbia.edu

Sanghack Lee*
Seoul National University
Seoul, South Korea
sanghack@snu.ac.kr

## Abstract

*Data augmentation is a popular tool for single source domain generalization, which expands the source domain by generating simulated ones, improving generalization on unseen target domains. In this work, we show that the performance of such augmentation-based methods in the target domains universally fluctuates during training, posing challenges in model selection under realistic scenarios. We argue that the fluctuation stems from the inability of the model to accumulate the knowledge learned from diverse augmentations, exacerbating feature distortion during training. Based on this observation, we propose a novel generalization method, coined Parameter-Space Ensemble with Entropy Regularization (PEER), that uses a proxy model to learn the augmented data on behalf of the main model. The main model is updated by averaging its parameters with the proxy model, progressively accumulating knowledge over the training steps. Maximizing the mutual information between the output representations of the two models guides the learning process of the proxy model, mitigating feature distortion during training. Experimental results demonstrate the effectiveness of PEER in reducing the OOD performance fluctuation and enhancing generalization across various datasets, including PACS, Digits, Office-Home, and VLCS. Notably, our method with simple random augmentation achieves state-of-the-art performance, surpassing prior approaches on sDG that utilize complex data augmentation strategies.*

## 1. Introduction

Real-world deployment of deep neural networks frequently encounters domain shift, which refers to the discrepancy between the training domain and the unseen target domain on which the model is tested. An important aspect of domain shift is that it hinders the generalization of trained models

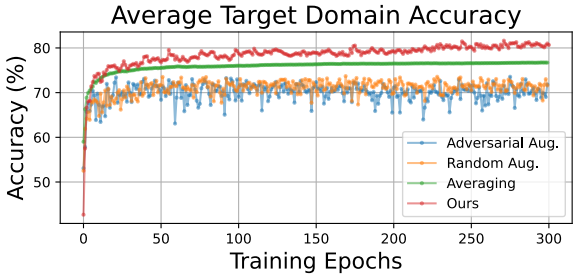



Figure 1. Despite its generalization effect, data augmentation induces fluctuations in target domain accuracy during the training. This phenomenon becomes more pronounced as the complexity of the augmentation increases, complicating model selection. We address this issue of fluctuation with a simple model-to-model regularization method that cushions the effect of data augmentation.

[36]. Nevertheless, a trained model is expected to perform well on various OOD data, given a limited source of training data. Similarly, single source domain generalization (sDG) is the task of building a robust model that performs well across multiple target domains, trained from a single source domain [63]. Existing approaches commonly utilize data augmentation to generate simulated target domains [60] and attempt to learn domain-invariant features from the augmented data.

This paper highlights an overlooked issue of leveraging data augmentation for sDG, particularly focusing on the fluctuation of OOD target domain performance amidst training, referred to as *mid-train OOD fluctuation* (Fig. 1). We find that this phenomenon stems from the model's incapability to accumulate the knowledge obtained from diverse augmentations and demonstrate that the features obtained from previous steps are largely distorted during training (see Fig. 2). We further illustrate that the fluctuation worsens when the model's trained features are distorted by augmented samples discrepant from the previously trained data and show that augmented samples are surprisingly inconsistent from their original state. This complicates *model selection* and poten-

*Corresponding authors.
†Work done at Seoul National University.

tially undermines generalization at test time, and thus, it is crucial to mitigate this issue.

Based on our observations, we suggest a novel generalization method coined PEER (Parameter-Space Ensemble with Entropy Regularization), that mitigates the augmentation-induced feature distortion by averaging parameters at various points along the model's learning trajectory [28]. Specifically, our method leverages two interacting modules, i.e., the task model and the proxy model, to accumulate the knowledge acquired during training. The parameter-averaged task model guides the learning process of the proxy model, significantly reducing the aforementioned mid-train OOD fluctuation. Consequently, our framework stacks the generalization effect of varying data augmentation into the task model, reaching state-of-the-art performance in conventional sDG benchmarks (e.g., PACS, Digits), even in benchmarks where conventional sDG methods do not guarantee generalization (e.g., Office-Home, VLCS).

Our contributions are summarized as follows:

- We highlight an overlooked issue of the mid-train OOD fluctuation of augmentation-based sDG methods which poses serious issues in model selection and reveal that it stems from the distortion in the trained features.
- Based on our observation, we introduce PEER, a novel framework for sDG that stabilizes the learning process and boosts the target domain accuracy by accumulating the generalization effect of diverse augmentations using a parameter-space ensemble model.
- Our method achieves state-of-the-art performance across a wide range of benchmarks against existing sDG methods.

## 2. Related Works

**Domain generalization.** In the multi-source domain generalization (DG) literature, learning domain-invariant features has shown success in training robust models [4]. Specifically, these algorithms aim to disentangle the knowledge shared across domains [31, 52]. A recent line of work highlighted the use of pre-trained models for model-to-model regularization, e.g., Cha et al. [9] used an external pre-trained model to encourage the learning of domain-invariant features, and Li et al. [40] expanded this approach by using multiple pre-trained models. In contrast, we refrain from using an external model and show that a training model can effectively perform regularization. On a different note, Arpit et al. [5] studied the instability of the model's OOD performance and suggested an ensemble algorithm to alleviate the stochastic nature of the learning process. In contrast, we relieve the computational burden of ensembles by using a single parameter-averaged model [1, 29, 50] and incorporate an alignment strategy [10, 18] to assist this.

**Single source domain generalization.** In the sDG setting, only one domain is available for training, which makes it hard to apply conventional approaches developed for DG. To tackle this, a line of work focused on generating diverse domains using sophisticated data augmentation strategies, e.g., adversarial augmentation [60] or learnable augmentation modules [16, 39, 48, 65, 68, 70]. On the other hand, we reveal a universal phenomenon (i.e., mid-train OOD fluctuation) associated with utilizing data augmentation for generalization, and present a simple strategy to alleviate it.

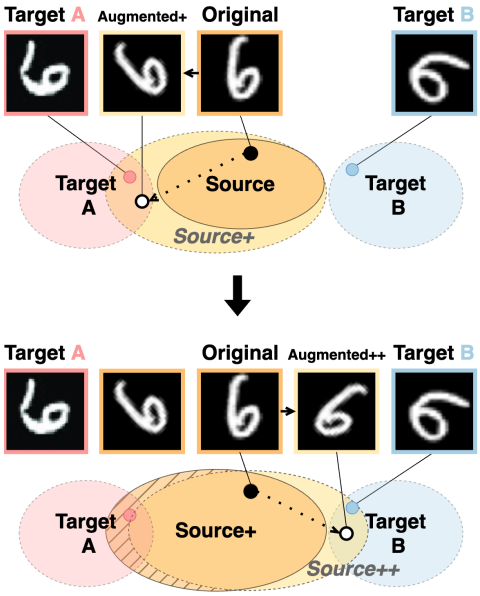



Figure 2. Illustration of pitfalls of augmentation in generalizing to unseen target domains. (a) Augmentation-based methods expand the source domain by providing diverse augmented samples (i.e., Source+). This enhances the model's generalization capability towards the unseen target domain (i.e., Target A). (b) Throughout the course of training, it iteratively simulates diverse unseen domains. However, at the same time, diverse augmentations lead to the distortion of the learned representations, thereby triggering OOD fluctuation.

**Mode connectivity and parameter-space ensembles.** Our work draws inspiration from the mode connectivity [18] property of neural networks, which refers to the presence of a continuous manifold of non-increasing error that connects the minima identified by two global minimizers (i.e., trained models) [21, 41]. The concept is commonly used to justify how individual models can be merged to produce parameter-space ensembles [50, 67] and also form the basis for designing model alignment methods to encourage mode connectivity between models [1, 10, 14, 51]. To analyze mode connectivity between models, a common practice is to measure the loss barrier [18], quantified as the rise in loss values when the parameters of two models are averaged. Extending this, we suggest an effective alignment method to encourage mode connectivity between models trained with

varying augmented data.

## 3. Observation: Pitfalls of Augmentation for Generalization

In this section, we reveal an overlooked problem in augmentation-based sDG methods. We first provide a brief background on the augmentation-based approaches to sDG (Sec. 3.1). Then, we highlight the performance fluctuation of models trained with data augmentation (Sec. 3.2).

### 3.1. Augment-and-Align: Augmentation-based Approaches to sDG

Let $\mathcal{D}_S = \{(x_i, y_i)\}_{i=1}^N$ be a source domain where $x_i \in \mathcal{X}$ is an input image and $y_i \in \mathcal{Y}$ is its corresponding label. The goal of sDG is to build a model $F$ from $\mathcal{D}_S$ that is capable of generalizing to unknown target domains $\{\mathcal{D}_T^{(1)}, \cdots, \mathcal{D}_T^{(t)}\}$ distributionally different from the source domain. The model $F = C \circ H$ consists of a feature extractor $H : \mathcal{X} \to \mathcal{H}$ and the classifier $C : \mathcal{H} \to \mathcal{Y}$. Clearly, the classifier relying on the domain-specific features would not generalize to unseen target domains, and thus it is crucial to learn domain-invariant features from the source domain.

Existing approaches utilize data augmentation to simulate domain shift and aim to extract domain-invariant features by aligning the feature distribution between the original sample $x$ and its augmented view $\bar{x} = G(x)$, where $G$ is the augmentation function. The objective of such augmentation-based sDG approaches, omitting some arguments for simplicity, can be written as:

$$\arg\min_{H,C} \mathbb{E}_{(x,y)\in\mathcal{D}_S} \Big( \mathcal{L}_{\text{CE}}(C(H(x)), y) + \mathcal{L}_{\text{align}}(x, \bar{x}; H) \Big), \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss and $\mathcal{L}_{\text{align}}$ is an alignment loss for capturing domain-invariant features by comparing $H(x)$ and $H(\bar{x})$. The commonly used alignment loss is InfoNCE [45], which lower bounds the mutual information $I(H(x), H(\bar{x}))$. Importantly, such alignment only guarantees to retrieve *augmentation-invariant* features [61], and simple input transformations for generating the augmented views are often insufficient to capture *domain-invariant* ones [3]. Therefore, recent methods devise more complex data augmentation strategies [39, 65] to simulate diverse shifts in distribution.

However, it is still unclear whether such augmentation strategies can guarantee generalization to the target domain, especially given that it is unseen. In the sequel, we illustrate that this discrepancy makes the model performance fluctuate in the target domain.

### 3.2. Mid-train OOD fluctuation of Augmentation-Based sDG Methods

Recall Fig. 1, we find that augmentation-based sDG methods commonly exhibit large fluctuation of OOD performance throughout training, dubbed mid-train OOD fluctuation. Then, the following questions naturally arise: *"How does the fluctuation relate to the generalization performance? Where does the fluctuation stem from?"* Here, we investigate the relationships between the fluctuation and target domain accuracy through the lens of source-target dataset distance and examine the impact of data augmentation on the fluctuation.

We begin by observing that the target domain accuracy is closely related to the mid-train OOD fluctuation by comparing two augmentation-based sDG methods: random augmentation (RandAug, Cubuk et al. [11]) and adversarial augmentation (AdvAug, Li et al. [39]). As shown in the last column (Avg.) of Tab. 1-(a) and (b), the models with better generalization performance also display larger fluctuation. Clearly, the complexity of the augmentation the models employed aligns with the target domain accuracy and fluctuation.

To further investigate their relationships, we adopt a similarity metric that measures the geometric distance between datasets (i.e., OTDD [2]). By comparing different target domains (e.g., SVHN and USPS), we observe that the source-target discrepancy shown in Tab. 1-(c) is closely associated with the target domain accuracy and fluctuation. In other words, the models exhibit relatively small fluctuation on the target domain that is similar to the source domain (i.e., USPS) and vice versa (i.e., SVHN). Similarly, the models tend to show higher accuracy on target domains with smaller discrepancies (i.e., USPS) and vice versa (i.e., SVHN).

To better understand our observations above, we examine the discrepancy between the original dataset (MNIST) and its augmented view across varying degrees of random augmentation [11]. As shown in Fig. 3, we observe that the discrepancy becomes more significant as the augmentation becomes diverse and its magnitude becomes stronger. Notably, such discrepancies often even exceed the source-target distance (i.e., 0.92 in Tab. 1-(c)).

Lastly, we find that parameter-averaging of multiple points along the model's learning trajectory [28] can drastically reduce the OOD fluctuation, although with only limited gains in generalization. This is illustrated by the green line in Fig. 1. Intuitively, this aligns with our idea that the model's learned features are consistently distorted during training, and parameter-averaging could alleviate the distortion [42].

Our observations suggest that data augmentation improves generalization capacity by simulating diverse domain shifts, but simultaneously leads to the distortion of the learned features and triggers mid-train OOD fluctuation, as depicted in Fig. 2. Based on our findings, we now proceed to present our method that retains the knowledge accumulated throughout the training, thereby alleviating fluctuations while achieving better generalization performance.

Table 1. Empirical study of (a) target domain accuracy, (b) mid-train OOD fluctuation, and (c) source-target dataset distance. We use MNIST as a source. Large source-target distance (red) coincided with low target accuracy and high OOD fluctuation during training, and vice versa (blue).

| Method | SVHN | M-M | S-D | USPS | Avg. |
|---|---|---|---|---|---|
| (a) Target domain accuracy | | | | | |
| NoAug | 27.83 | 52.72 | 39.65 | 76.94 | 49.29 |
| RandAug [11] | 57.76 | 77.15 | 73.65 | 87.94 | 73.98 |
| AdvAug [39] | 62.21 | 82.20 | 69.39 | 85.26 | 74.77 |
| (b) Variance of the target domain accuracy | | | | | |
| NoAug | 4.76 | 2.77 | 1.72 | 0.32 | 1.33 |
| RandAug [11] | 2.51 | 1.04 | 1.05 | 1.49 | 1.52 |
| AdvAug [39] | 3.58 | 2.56 | 2.36 | 3.48 | 2.99 |
| (c) Source-target dataset distance [2] ($\times 10^3$) | | | | | |
| - | *3.46* | *2.65* | *2.75* | *0.92* | *2.45* |

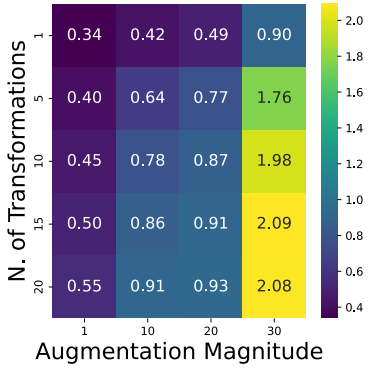



Figure 3. OTDD distance [2] between the original data (MNIST) and its augmented view.

## 4. Method

We now present a novel generalization method for sDG, coined Parameter-Space Ensemble with Entropy Regularization (PEER), that mitigates the augmentation-induced feature distortion and its associated issues (e.g., mid-train OOD fluctuation). Our approach involves two interacting modules with identical architectures: a frozen task model $F$ and a trainable proxy model $P$. The task model guides the proxy model's learning process through entropy regularization of feature representations (Sec. 4.1). Subsequently, the task model is updated via parameter-averaging with the regularized proxy model, progressively accumulating the proxy model's knowledge throughout training (Sec. 4.2). The concept of our method is depicted in Fig. 6. The pseudo-code of our method is provided in Algorithm 1.

### 4.1. Regulating the Proxy Model with PEER

Our goal is to learn a robust task model $F$ from a single source domain that can generalize to multiple unseen target domains, where the task model consists of a frozen encoder $H_f : \mathcal{X} \to \mathcal{H}$ and a frozen classification head $C_f : \mathcal{H} \to \mathcal{Y}$, i.e., $F = C_f \circ H_f$. However, directly training the task model with varying augmented data is prone to feature distortion. Our key idea is to introduce a proxy model $P$ that trains on behalf of the task model and under the its guidance. Specifically, the proxy model $P = C_p \circ H_p$ shares the same architecture as the task model and consists of an encoder $H_p : \mathcal{X} \to \mathcal{H}$ and a classification head $C_p : \mathcal{H} \to \mathcal{Y}$. The proxy model is initialized by copying the task model at the beginning of training, i.e., $\theta_p \leftarrow \theta_f^{(0)}$ where $\theta_p$ is the parameters of the proxy model $P$ and $\theta_f^{(n)}$ is the parameters of the task model $F$ at $n$-th training epoch.

Our method PEER imposes regularization to the proxy model at the intermediate feature level. Instead of directly comparing the intermediate representation in $\mathcal{H}$, we map the representations from $H_f$ and $H_p$ using a shared projection head $R : \mathcal{H} \to \mathcal{R}$, following the empirical analysis by Gupta et al. [24] and our experimental findings (Tab. 15) regarding its optimization efficacy.

The objective for PEER is then defined as:

$$\mathcal{L}_{\text{PEER}}(H_f(x), H_p(\bar{x})) = -I(R(H_f(x)); R(H_p(\bar{x}))), \quad (2)$$

where $x$ denotes the original sample and $\bar{x}$ the augmented view created by an augmentation function $G$. The loss function $\mathcal{L}_{\text{PEER}}$ is designed to maximize the mutual information ($I$) between the two representations $R(H_f(x))$ and $R(H_p(\bar{x}))$. Since the exact mutual information is intractable, we use practical lower bounds $\tilde{I}$ such as the InfoNCE [45] or the Barlow Twins [69] loss functions, both effective in optimizing mutual information between feature representations [47]. The details of the mutual information optimization are included in Appendix B.2.

Intuitively, regularizing with PEER guides the proxy model $P$ to learn features selected by the task model $F$. Notably, in Eq. (2), the task model and the proxy model receive nonidentical inputs $x$ and $\bar{x}$, respectively, reflecting our idea that the frozen task model is expected to provide a rich feature representation of the original sample $x$, while the training proxy model can better comprehend the newly augmented sample $\bar{x}$.

We train *only* the proxy model $P$ using a classification loss (i.e., cross-entropy) with the regularization:

$$\mathcal{L}_P = \sum\nolimits_{x' \in \{x, \bar{x}\}} \mathcal{L}_{\text{CE}}(C_p(H_p(x')), y) + w \cdot \mathcal{L}_{\text{PEER}}(H_f(x), H_p(\bar{x})), \quad (3)$$

where $w$ is a balancing coefficient. In Sec. 4.3, we further elaborate on the PEER regularization as an optimization of the mutual information (MI).

**Algorithm 1:** Parameter-space Ensemble with Entropy Regularization (PEER)

**Input:** Task model $F$ and its parameter $\theta_f$, augmentation function $G$, data from source domain $D_s$, augmentation reinitialization criteria $k$;
**Output:** Fully updated task model $F$ and its parameter $\theta_f$
Pre-train $F$ with $D_s$ without $G$
Initialize $P$ by setting its parameter $\theta_p$ with $\theta_f$ from $F$
Initialize trajectory $\Theta \leftarrow \{\,\}$
**while** *not converge* **do**
  **if** $n \,\%\, k$=0 **then**
    Reinitialize $G$ `// for random augmentation, change augmentation strength`
    $\Theta \leftarrow \Theta \cup \{\theta_p^{(n)}\}$ `// save a snapshot of` $P$
    $\theta_f \leftarrow$ AVERAGE$(\Theta)$ `// update` $F$ (Equation (4))
  **for** $i = 1 : n_{iterations}$ **do**
    Augment the $i$-th mini-batch sampled from $D_s$ with $G$
    Train $P$ with PEER following Equation (3)

### 4.2. Accumulating Knowledge in the Task Model with PEER

The task model $F$ is gradually updated through parameter-averaging with the proxy model $P$. This updating process progressively improves the task model's generalization throughout training, ensuring it remains effective as the regulator of the ever-growing proxy model [8]. Specifically, we update the task model by parameter-averaging with the proxy model for every $k$ epoch through the proxy model's learning trajectory i.e., $\Theta = \left\{\theta_p^{(k)}, \theta_p^{(2k)} \cdots, \theta_p^{(\lfloor \frac{n}{k} \rfloor \cdot k)}\right\}$ where $n$ is the current training epoch, and update the task model with:

$$\theta_f \leftarrow \frac{1}{|\Theta|} \sum_{\theta \in \Theta} \theta. \tag{4}$$

Also, we reinitialize the augmentation function $G$ for every $k$ epoch (e.g., changing the policy – number and of transformations/ magnitude – of random augmentation). This periodic update of the task model allows it to stack the effect of diverse augmentations, similar to an ensemble model [50].

For the parameter-averaged task model to enjoy ensemble effects, it's crucial to ensure mode connectivity [18] between the task model and the proxy model, which can be sufficed by sharing an identical initialization or backbone [44]. As our proxy model is initialized from the task model, it naturally satisfies this requirement. To further benefit parameter-averaging, the two models must be closely located in the feature space, which can be obtained by tuning the models on an identical source data [10, 51]. Our regularization with PEER (Eq. (2)) encourages the proxy model to be aligned with the task model in the feature space by treating the augmented domain similarly to the source domain. In Sec. 5, we show that the task model and the proxy model benefit from the regularization's alignment effect. In Appendix A, we empirically demonstrate that the task model cannot function as an effective regulator of the proxy model without the updating process (w/o ParamAvg. in Tab. 5).

### 4.3. Discussion

**PEER as mutual information (MI) maximization.** The idea of PEER is that we can leverage the frozen task model to regularize the proxy model by maximizing the shared information between the two models. PEER aims to maximize the MI between the intermediate output features of the two encoders $H_f$ and $H_p$. The entropy regularization aligns the proxy model to the task model, preventing the proxy model from deviating too far from the task model. From this perspective, an intended objective for PEER could be formulated as $\max_H \; I(H_f(\bar{x}); H_p(x))$ where $I(X;Y) = \mathbb{E}_{p(x,y)}[\log p(x \mid y)/p(x)]$ indicates the mutual information i.e., MI. In our implementation, PEER uses a feature decorrelation loss Eq. (6) [69] to maximize the lower bound of MI as a surrogate objective for MI optimization under a Gaussian assumption [57]. We further elaborate on the adequacy of Eq. (6) for MI optimization in Appendix A and report comparative results of different objectives e.g., InfoNCE [45] and Barlow Twins [69] (Tab. 7). In Sec. 5, we provide experimental analysis on the effect of PEER by showing its effectiveness in alleviating augmentation-induced feature distortion.

## 5. Experiment

In this section, we investigate the following questions: (1) How effective is our method compared to prior sDG approaches? (Tabs. 2 and 3) (2) Does our method reduce the fluctuation of OOD performance? (Tab. 4) (3) What effect does our method have on the model's learned features and loss landscape connectivity? (Figs. 4, 5 and 7) (4) How effective is our method compared to previous model-to-model regularization approaches (Tab. 5) or ensemble methods (Tab. 6)?

### 5.1. Experimental Setup

**Datasets.** Following prior works [39, 62], we evaluate our method on two standard benchmarks for sDG. **PACS** [38] consists of 4 domains of differing styles (Photo, Art, Cartoon, and Sketch) with 7 classes. By default, we train our model with the Photo domain and evaluate it on the remaining target domains. **Digits** comprises of 5 different digit classification datasets, MNIST [12], SVHN [43], MNIST-M (M-M) [20], SYNDIGIT (S-D) [19], and USPS [37]. We train our model with the first 10,000 samples of the MNIST dataset and assess its generalization accuracy across the remaining domains.

We also include Office-Home [58] and VLCS [17], challenging benchmarks for sDG methods. **Office-Home** is a

Table 2. Target domain accuracy on PACS and Digits († indicates numbers are from original authors).

| Method | PACS | | | | Digits | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A | C | S | Avg. | SVHN | M-M | S-D | USPS | Avg. |
| ERM [32] | 54.43 | 42.74 | 42.02 | 46.39 | 27.83 | 52.72 | 39.65 | 76.94 | 49.29 |
| ADA† [16] | 58.72 | 45.58 | 48.26 | 50.85 | 35.51 | 60.41 | 45.32 | 77.26 | 54.62 |
| M-ADA† [48] | 58.96 | 44.09 | 49.96 | 51.00 | 42.55 | 67.94 | 48.95 | 78.53 | 59.49 |
| L2D† [65] | 56.26 | 51.04 | 58.42 | 55.24 | 62.86 | 87.30 | 63.72 | 83.97 | 74.46 |
| PDEN [39] | 57.41 | 45.77 | 65.01 | 56.06 | 62.21 | 82.20 | 69.39 | 85.26 | 74.77 |
| SimDE† [68] | – | – | – | 59.32 | 66.08 | 84.90 | 70.04 | 86.56 | 76.89 |
| AdvST [70] | 53.95 | 46.11 | 49.63 | 49.90 | 67.50 | 79.80 | 78.10 | 94.80 | 80.10 |
| MetaCNN† [62] | 54.05 | **53.58** | 63.88 | 57.17 | 66.50 | **88.27** | 70.66 | 89.64 | 78.76 |
| RandAug [11] | 54.17 | 47.48 | 65.11 | 55.59 | 57.76 | 77.15 | 73.65 | 87.94 | 73.98 |
| PEER (ours) | **62.66** | 47.40 | **68.21** | **59.42** | **70.79** | 76.84 | **83.05** | **93.57** | **81.06** |

common multi-DG benchmark consisting of 4 datasets (Real-world, Art, Clipart, Product) with differing styles with 65 classes. We train on the Real-world domain and evaluate with the remaining domains. **VLCS** is also a benchmark for multi-DG, comprised of 4 datasets, PASCAL-VOC (V), LabelMe (L), Caltech-101 (C), and SUN09 (S) with varying styles. We used the PASCAL-VOC dataset as the source and the rest as target domains.

**Baselines.** We first consider ERM [32] and also compare our method with several strong augmentation-based approaches, i.e., M-ADA [48], L2D [65], PDEN [39], SimDE [68] and AdvST [70]. Some recent works [68, 70] have reported the results using a different backbone (ResNet-18 in PACS) from the standard setting (AlexNet), thus we have used the authors' codes (if applicable) for reassessment.

**Implementation.** We use the same backbone architecture as prior works to ensure fair comparison. Specifically, we used AlexNet and multi-layer CNN for PACS and Digits, respectively, following earlier works [39, 59, 62]. For Office-Home and VLCS, we used ResNet-18. Additional experimental results across various backbone models (e.g., ResNet-18/50) are provided in Appendix (Tabs. 13 and 14). For the implementation of our method, we use random augmentation [11] to generate augmented samples. We set $k = 10$ and the balancing coefficients $\lambda = 0.005$, and $w = 2$ for all experiments. Hyperparameter studies are provided in Appendix D.2. We report the final test accuracy of the task model and report the OOD fluctuation measured as the variance of the target domain accuracy for every $k$-th epoch (Tab. 4). Throughout this section, we use the abbreviation RA for Random Augmentation and P for PEER.

## 5.2. Main Results

In Tabs. 2 and 3, we report experimental results using the accuracy for each target domain and the mean accuracy across all target domains. In standard sDG benchmarks (i.e., PACS, Digits; Tab. 2), our method achieves state-of-the-art target domain accuracy in many of the target domains and outperforms all baselines in terms of mean accuracy.

Table 3. Target domain accuracy on Office-Home and VLCS.

| Method | Office-Home | | | | VLCS | | | |
|---|---|---|---|---|---|---|---|---|
| | Art | Clipart | Product | Avg. | L | C | S | Avg. |
| ERM [32] | 52.78 | 40.19 | 68.73 | 53.90 | 59.06 | 97.30 | **74.25** | 76.87 |
| M-ADA [48] | 54.36 | 40.41 | 65.11 | 53.29 | 57.84 | 97.88 | 64.42 | 73.38 |
| L2D [65] | 54.02 | 41.77 | 66.30 | 54.03 | 56.21 | 95.52 | 66.90 | 72.87 |
| PDEN [39] | 53.39 | 43.38 | 66.25 | 54.34 | 62.55 | 96.11 | 73.52 | 77.39 |
| RandAug [11] | 43.10 | 45.47 | 61.67 | 50.01 | 57.58 | 93.18 | 66.56 | 72.44 |
| PEER (ours) | **56.81** | **54.23** | **70.84** | **60.63** | **67.00** | **97.73** | 72.56 | **79.10** |

Please note that SimDE and AdvST have used more robust backbones (ResNet-18) than the standard setting (AlexNet), which makes direct comparison challenging. Notably, our method outperforms current SoTA methods (using the same backbone) by $2.30\%$ and $0.96\%$. It is worth noting that our simple method boosted the mean accuracy of random augmentation (RandAug) by $7.08\% \uparrow$ in Digits and $3.76\% \uparrow$ in PACS.

In more challenging benchmarks (i.e., Office-Home, VLCS; Tab. 3), previous augmentation-based methods (e.g., PDEN, RandAug) show either small gains or negative effects in enhancing generalization. Similarly, naively applying random augmentation for these benchmarks lowered the target domain accuracy. In contrast, applying random augmentation with PEER, shows a significant performance gain of $10.62\%$ in Office-Home and $6.66\%$ in VLCS.

Finally, Tab. 4 demonstrates the fluctuation of OOD performance, measured as the variance across the target domain accuracy. We observe that our method successfully reduces the mid-train OOD fluctuation across all benchmarks. In our framework, the task model accumulates knowledge of the proxy model throughout the training. Thus, regularizing with the task model encourages the proxy model to preserve the knowledge of previous steps, similar to a memory buffer used in continual learning [64]. In the next section, we illustrate that the task model indeed preserves the knowledge of the proxy model through parameter averaging.

## 5.3. Detailed Analysis on PEER

### 5.3.1. Advantages of PEER in Model-to-Model Regularization

In Tab. 5, we demonstrate the advantages of PEER compared to previous approaches that utilize a pre-trained model (i.e., teacher) for regularization, where T+RA and P+RA refer to applying the teacher and the PEER regularization, respectively. We observe that both the teacher and the task model in PEER reduce the OOD fluctuation, while the fully-trained teacher (T+RA) often displays a stronger regularization effect compared to PEER (P+RA). However, PEER achieves superior sDG target domain accuracy in both datasets compared to the teacher. This is due to the teacher model's static nature, which limits its capability to process newly augmented samples. In contrast, our task model, evolving

Table 4. Variance of the target domain accuracy.

| | PACS | | | | Digits | | | | | Office-Home | | | | VLCS | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | A | C | S | Avg. | SVHN | M-M | S-D | USPS | Avg. | Art | Clipart | Product | Avg. | L | C | S | Avg. |
| L2D [65] | 3.70 | 5.30 | 13.37 | 7.46 | 3.53 | 3.01 | 2.59 | 4.44 | 3.39 | 5.22 | 1.90 | 5.58 | 4.23 | 5.72 | 0.59 | 1.66 | 2.66 |
| PDEN [39] | 3.39 | 5.22 | 7.23 | 5.28 | 3.58 | 2.56 | 2.36 | 3.48 | 2.99 | 10.63 | 2.17 | 7.46 | 6.75 | 2.44 | 2.39 | 2.81 | 2.55 |
| RandAug [11] | 2.23 | 4.81 | 5.01 | 4.02 | 2.51 | **1.04** | 1.05 | 1.49 | 1.52 | **3.49** | 2.17 | 2.74 | 1.89 | 3.02 | 1.61 | **1.96** | 2.20 |
| PEER (ours) | **2.01** | **3.98** | **4.77** | **3.59** | **2.03** | 1.11 | **1.04** | **1.24** | **1.36** | 3.99 | **1.41** | **1.80** | **1.31** | **2.05** | **1.61** | 2.10 | **1.92** |
| Metric | Source-target dataset distance ($\times 10^3$) | | | | | | | | | | | | | | | | |
| OTDD [2] | 13.37 | 29.52 | 49.94 | 30.94 | 3.46 | 2.65 | 2.75 | 0.92 | 2.45 | 19.53 | 19.29 | 20.63 | 19.82 | 11.79 | 10.14 | 11.77 | 11.23 |

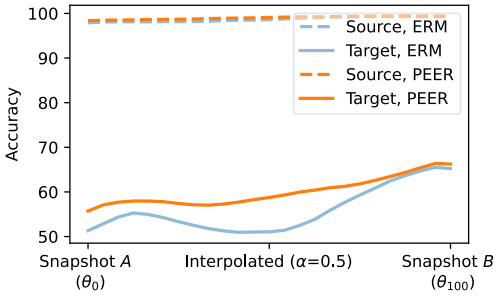



Figure 4. Mode connectivity in the proxy model's trajectory. PEER benefits parameter-averaging between snapshots of $P$ through its regularization effects.

with the proxy model, is less vulnerable to these limitations.

We further validate the effectiveness of the updating process by ablating parameter-averaging (w/o ParamAvg. in Tab. 5). Instead of updating the task model by parameter-averaging, we simply freeze a snapshot of the proxy model for every $k$ epoch and use the latest snapshot as the regulator. As shown in Tab. 5, the non-averaged task model sacrifices the target domain accuracy for addressing OOD fluctuation, which illustrates the effectiveness of parameter-averaging.

#### 5.3.2. Effect of PEER on Parameter-Averaging

Here, we investigate the effect of PEER regularization in benefiting parameter-averaging for the task model $F$ update. We observe that the regularization brings forth an alignment between different steps of the proxy model in its learning trajectory $\Theta$. To clarify, we find different steps of the proxy model $\theta_p^{(i)}, \theta_p^{(j)}$ to be aligned by the regularization. To show this, we follow the practice of Frankle et al. [18] and analyze the loss barrier between snapshots of the proxy model in its learning trajectory. Fig. 4 illustrates the mode connectivity of the proxy model training with data augmentation with/without PEER on Digits (source: MNIST, target: SVHN). Here, we analyze the connectivity of the proxy model in its early stage of training ($\theta_p^{(0)}$) and at the late stage ($\theta_p^{(100)}$) by interpolating the two $\alpha\theta_p^{(0)} + (1-\alpha)\theta_p^{(100)}$, where $\alpha \in [0, 1]$ be the interpolation weight. We note that PEER aligns the model's snapshots($\theta_p^{(0)}, \theta_p^{(100)}$) in its learning trajectory, gifting a stronger performance gain when it is interpolated ($\alpha = 0.5$), especially in the OOD target domain. In other words, PEER's regularization enables the task model to function as a robust parameter-space ensemble, which can guide the proxy model's generalization to target domains.

We further investigate the PEER's role in parameter-averaging in Tab. 6, specifically showing the failure cases of parameter-averaging without model alignment. Here, P-ENS refers to the parameter-space ensembles. In both PACS and Digits, parameter-space ensembling without regularization (P-ENS w/o PEER) falls behind ensembling with regularization. Notably in PACS, we observe failure cases of parameter-space ensembling without regularization, where the ensemble effect (i.e., gain in generalization ability) was very marginal. This failure case in parameter-averaging is an interesting observation as averaging the parameters between different training step snapshots of the same model has shown great success in many previous works [23, 28]. In Appendix C.2, we provide a deeper analysis of this topic.

#### 5.3.3. Effect of PEER on Learned Features

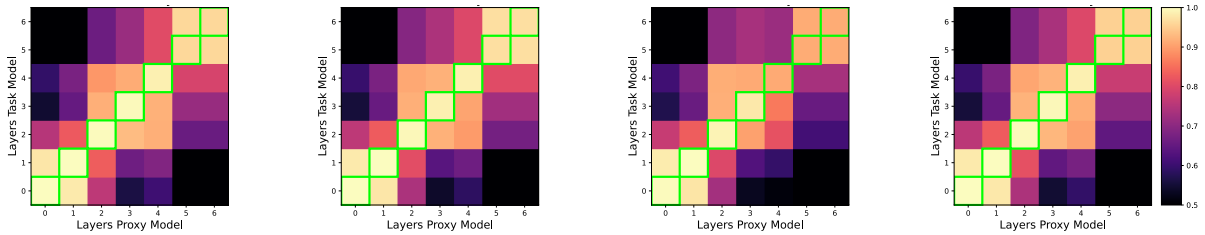
In this section, we analyze the PEER's effect on the learned feature representations. In detail, we share two results: (1) parameter-averaging allows the task model to accumulate the proxy model's knowledge, (2) the PEER regularization addresses the proxy model's feature distortion (Appendix C.1).

To show this, we follow the practice of Neyshabur et al. [44] and compute the Centered Kernel Alignment (CKA) metric [33] between trained models. The CKA metric measures the similarity between feature representations, where 1.0 indicates perfect alignment. Specifically, we compute and visualize the CKA similarity for different layers of the multi-layer CNN network trained on the Digits setting (see Appendix E.4 for details). Each matrix in Figs. 5 and 7 displays the similarity between the two models, its diagonal values indicating the similarity between corresponding layers' features, i.e. brighter boxes indicate more shared knowledge.

We report that the parameter-averaging allows the task model to function similarly to a buffer which accumulates the knowledge of the proxy model across previous training steps. Fig. 5, we illustrate the feature similarity between the task model $F$ ($\theta_f$) and the proxy model $P$ ($\theta_p$). We can see that

Table 5. Comparitive study on PEER vs. Teacher.

| Method | Regulator | PACS | | | | Digits | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | A | C | S | Avg. | SVHN | M-M | S-D | USPS | Avg. |
| Variance of the target domain accuracy (OOD Fluctuation) | | | | | | | | | | |
| RandAug [11] | N/A | 2.23 | 4.81 | 5.01 | 4.02 | 2.51 | 1.04 | 1.05 | 1.49 | 1.52 |
| T+RA | Teacher | **1.27** | **2.49** | 5.30 | **3.02** | 1.95 | 1.17 | **1.10** | **1.11** | **1.33** |
| P+RA | PEER (w/o ParamAvg.) | 1.69 | 3.38 | **4.62** | 3.23 | **1.93** | **1.10** | 1.11 | 1.22 | 1.34 |
| P+RA | PEER | 2.01 | 3.98 | 4.77 | 3.59 | 2.03 | 1.11 | 1.04 | 1.24 | 1.36 |
| Target Domain Accuracy | | | | | | | | | | |
| RandAug [11] | N/A | 54.17 | **47.48** | 65.11 | 55.59 | 57.76 | 77.15 | 73.65 | 87.94 | 73.98 |
| T+RA | Teacher | 58.61 | 46.66 | 64.23 | 56.50 | 63.37 | 72.63 | 77.91 | 87.39 | 75.33 |
| P+RA | PEER (w/o ParamAvg.) | 57.73 | 46.69 | 61.33 | 55.25 | 59.99 | **77.26** | 72.3 | 88.28 | 74.46 |
| P+RA | PEER | **62.66** | 47.40 | **68.21** | **59.42** | **70.79** | 76.84 | **83.05** | **93.57** | **81.06** |




(a) Epoch 30 (b) Epoch 60 (c) Epoch 90 (d) Epoch 120

Figure 5. Layer-wise feature similarity between the fully updated task model and the proxy model at different epochs. The task model gradually accumulates the knowledge of the proxy model.

Table 6. The target domain accuracy of the parameter-space ensemble ($^\dagger$ indicates numbers are from original authors).

| Method | Ensemble | PACS | | | | Digits | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | A | C | S | Avg. | SVHN | M-M | S-D | USPS | Avg. |
| ERM [32] | ✗ | 54.43 | 42.74 | 42.02 | 46.39 | 27.83 | 52.72 | 39.65 | 76.94 | 49.29 |
| MetaCNN$^\dagger$ [62] | ✗ | 54.05 | **53.58** | 63.88 | 57.17 | 66.50 | **88.27** | 70.66 | 89.64 | 78.76 |
| P-ENS w/o PEER | ✓ | **63.20** | 41.08 | 56.25 | 53.51 | **71.87** | 76.42 | 82.36 | 92.23 | 80.72 |
| P-ENS PEER (ours) | ✓ | 62.66 | 47.40 | **68.21** | **59.42** | 70.79 | 76.84 | **83.05** | **93.57** | **81.06** |

the fully updated task model is closely aligned with different stages of the proxy model's trajectory (indicated by bright diagonal values in Fig. 5), suggesting that the parameter-averaging effectively consolidates knowledge from various augmentations and preserves features that might otherwise be distorted during training. Continuing this discussion, on Appendix C.1, we show that PEER plays an important role in addressing the feature distortion during training (Fig. 7).

### 5.4. Ablation Study

We conduct an ablation study to evaluate the impact of various components on overall performance, including the regularization objective (Tab. 7), hyperparameters $w$, $\lambda$, and $k$ (Tabs. 9a and 9b) , model size (Tabs. 13 and 14) , and the role of the projection head (Tab. 15).

## 6. Conclusion

This paper presents PEER, a novel generalization method to address the issues of augmentation-based approaches to single source domain generalization. We highlight the feature distortion induced by augmentation, which triggers fluctuations in the target domain performance during training. Based on our observations, we propose a parameter-averaged task model that accumulates the generalization effect of the training proxy model. Entropy regularization on their learned feature representation aligns the two models, addressing feature distortion. Experiments on various datasets (PACS, Digits, Office-Home, VLCS) demonstrate the effectiveness of our method in stabilizing the learning process and enhancing the generalization performance.

## Acknowledgment

We thank anonymous reviewers for constructive comments to improve the manuscript. This work was partly supported by the IITP (RS-2022-II220953/25%) and NRF (RS-2023-00211904/50%, RS-2023-00222663/25%) grant funded by the Korean government. This work was supported in part through the NYU IT High-Performance Computing resources, services, and staff expertise.

## References

[1] Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries, 2023. 2, 15

[2] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020. 3, 4, 7

[3] Masih Aminbeidokhti, Fidel A. Guerrero Peña, Heitor Rapela Medeiros, Thomas Dubail, Eric Granger, and Marco Pedersoli. Domain generalization by rejecting extreme augmentations, 2023. 3

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2019. 2

[5] 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. 2, 13

[6] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023. 14

[7] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent, 2022. 12

[8] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. 5, 12

[9] 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, 2022. 2, 12, 20

[10] Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*, 2022. 2, 5, 15

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 3, 4, 6, 7, 8, 18, 19

[12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5, 12, 18, 21

[13] Nikos Efthymiadis, Giorgos Tolias, and Ondřej Chum. Crafting distribution shifts for validation and training in single source domain generalization. *arXiv:2409.19774*, 2024. 16

[14] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021. 2

[15] Daniel Falbel. *torchvision: Models, Datasets and Transformations for Images*, 2023. https://torchvision.mlverse.org, https://github.com/mlverse/torchvision. 20, 21

[16] 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. 2, 6

[17] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013. 5, 18, 21

[18] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 2, 5, 7

[19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015. 5, 18

[20] 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. 5, 18

[21] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018. 2

[22] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 12

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 7, 15

[24] Kartik Gupta, Thalaiyasingam Ajanthan, Anton van den Hengel, and Stephen Gould. Understanding and improving the role of projection head in self-supervised learning, 2022. 4

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 12

[26] 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, 2019. 14

[27] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning, 2021. 14

[28] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 2, 3, 7, 15

[29] Alexia Jolicoeur-Martineau, Emy Gervais, Kilian Fatras, Yan Zhang, and Simon Lacoste-Julien. Population parameter averaging (papa). *arXiv preprint arXiv:2304.03094*, 2023. 2

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 20

[31] 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. 2

[32] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: École d'Été de Probabilités de Saint-Flour XXXVIII-2008*. Springer Berlin Heidelberg, 2011. 6, 8, 18

[33] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 7

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 19

[35] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. 13

[36] 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. 1

[37] 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*, page 396–404, Cambridge, MA, USA, 1989. MIT Press. 5, 18

[38] 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. 5, 18, 20, 21

[39] 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, 2021. IEEE Computer Society. 2, 3, 4, 5, 6, 7, 18, 21

[40] 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. 2, 12

[41] Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, pages 22965–23004. PMLR, 2023. 2

[42] Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. *arXiv preprint arXiv:2312.08977*, 2023. 3

[43] 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. 5, 18

[44] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. 5, 7

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. 3, 4, 5, 13, 15

[46] Liam Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15(6):1191–1253, 2003. 13

[47] 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. 4, 13

[48] 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. 2, 6, 18

[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. 12, 20

[50] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022. 2, 5

[51] Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 28656–28679. PMLR, 2023. 2, 5, 15

[52] 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. 2

[53] 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. 19

[54] Haizhou Shi and Hao Wang. A unified approach to domain incremental learning with memory: Theory and algorithm.

*Advances in Neural Information Processing Systems*, 36, 2024. 13

[55] 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. 14

[56] 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, 2022. IEEE Computer Society. 14

[57] 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. 5, 15, 20

[58] 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. 5, 18, 21

[59] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation, 2018. 6

[60] 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. 1, 2

[61] 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. 3

[62] 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. 5, 6, 8, 16, 18

[63] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization, 2021. 1

[64] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 6

[65] 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, 2021. 2, 3, 6, 7, 18

[66] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. 13, 14

[67] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. 2

[68] Qinwei Xu, Ruipeng Zhang, Yi-Yan Wu, Ya Zhang, Ning Liu, and Yanfeng Wang. Simde: A simple domain expansion approach for single-source domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4798–4808, 2023. 2, 6

[69] 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. 4, 5, 14, 15, 20

[70] Guangtao Zheng, Mengdi Huai, and Aidong Zhang. Advst: Revisiting data augmentations for single domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21832–21840, 2024. 2, 6, 18