Dealing with the Evil Twins: Improving Random Augmentation by Addressing Catastrophic Forgetting of Diverse Augmentations

Dongkyu Cho
New York University
New York, New York
dongkyu.cho@nyu.edu

Rumi Chunara
New York University
New York, New York
rumi.chunara@nyu.edu

Abstract

Data augmentation is a promising tool for enhancing out-of-distribution generalization, where the key is to produce diverse, challenging variations of the source domain via costly targeted augmentations that maximize its generalization effect. Conversely, random augmentation is inexpensive but is deemed suboptimal due to its limited effect. In this paper, we revisit random augmentation and explore methods to address its shortcomings. We show that the stochastic nature of random augmentation can produce a set of colliding augmentations that distorts the learned features, similar to catastrophic forgetting [22]. We propose a simple solution that improves the generalization effect of random augmentation by addressing forgetting, which displays strong generalization performance across various single source domain generalization (sDG) benchmarks.

1. Introduction

Learning a robust model from a single source domain mirrors many real-world scenarios where a machine learning model has limited access to training data but is expected to perform well across various out-of-distribution (OOD) test data [10]. Previous studies have shown that expanding the source domain through data augmentation [34] improves generalization on unseen target domains. Fig. 1 illustrates the general concept of these augmentation-based generalization methods. Many of these methods rely on targeted augmentation techniques that select transformations based on designated objectives (e.g., adversarial loss [40, 50]), owing to its efficacy in improving the model's OOD generalization capabilities. In contrast, random augmentation selects transformations stochastically, while showing a limited generalization effect [1].

In this paper, we answer the following question: "Why does random augmentation underperform?" despite its ability to efficiently provide challenging, diverse augmentations.

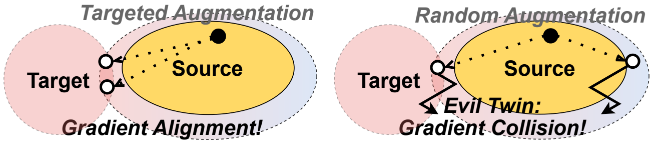


Figure 1. Data Augmentation enhances generalization by generating samples closer to the target data. However, contradicting augmentations can induce gradient collisions, leading to forgetting.

An important motivation for this work is that cost-effective data augmentation is crucial, especially in the era of large foundation models, where training and inference resources are already at a premium. Targeted augmentation methods, while effective, generally demand additional inference passes through the main model as well as an auxiliary module to optimize or select transformations, incurring significant computational overhead. In contrast, random augmentation imposes minimal extra cost during training yet still ensures a broad range of transformations.

We hypothesize that the underperformance of random augmentation stems from its stochastic nature, where colliding augmentations of the same source– evil twin–distort the model's trained features [22]. Throughout this paper, we theoretically demonstrate that the diversity of the random augmentation stimulates gradient collision [38], distorting learned features (i.e., forgetting). We substantiate our hypothesis through empirical analysis and show that conventional continual learning (CL) algorithms can mitigate this phenomenon, but are generally unsuitable. Reflecting on this, we propose a simple solution that directly manipulates the model's weight space to accumulate the learned features of colliding augmentations.

Our contributions are summarized as follows:

- We theoretically show that diverse augmentations can induce a unique variant of catastrophic forgetting induced by colliding augmentations (i.e., evil twin).
- We show that the generalization effect of random augmentation can be enhanced by addressing forgetting, namely through continual learning algorithms.
- We present a simple method that substitutes replay memory

using model weights.

2. Related Works

Data Augmentation: Targeted vs. Random Data augmentation is a common technique for improving model robustness by improving data diversity [15]. It is instrumental when the training data is limited or lacks diversity [2, 20], such as in cases of single-source domain generalization [10]. At large, two categories of data augmentation techniques are used: targeted augmentation and random augmentation. Targeted augmentation techniques aim to generate challenging augmentations, namely via providing adversarial examples [17, 46]. However, this family of data augmentation is computationally demanding due to the additional cost required for adversarial training. On the other hand, random augmentation applies a random combination of transformations (e.g., flipping, noise injection, etc.) [7, 8, 10]. Compared to targeted augmentation techniques, random augmentation does not require additional optimization processes but shows limited generalization effect [1].

Single Source Domain Generalization In single-source domain generalization (sDG), only one domain is available for training, limiting the ability to exploit domain differences for robustness. To address this, many methods simulate diverse domains via data augmentation. For example, Volpi et al. [40] adopted adversarial learning strategies to generate complex, label-preserving perturbations, and Zhao et al. [49] introduced entropy-based regularization to generate challenging augmentations. These strategies inspired subsequent approaches [29, 44, 47, 50]. Meanwhile, Efthymiadis et al. [10] introduced a novel validation strategy for sDG. In this work, we evaluate the efficacy of data augmentation on standard sDG benchmarks.

Catastrophic Forgetting & Continual Learning Catastrophic forgetting [16] is a common phenomenon in machine learning, particularly in the context of continual learning [22, 43]. Forgetting occurs when a pre-trained model forgets previously learned features when learning a new domain. While most studies focus on catastrophic forgetting in a multi-domain setting [4, 36, 42], our work addresses a lesser-explored issue: catastrophic forgetting exacerbated by data augmentation, in a single domain setting [6, 18, 31, 37].

3. Evil Twin Effect: Random Augmentation Exacerbates Catastrophic Forgetting of diverse augmentations.

In this section, we analyze the generalization effect of random augmentation and reveal its overlooked issues. In the sequel, we show that (1) random augmentation provides diverse transformations that benefit generalization, but (2) the diversity of the random augmentation produces contradicting augmentations that exacerbate gradient collision (i.e., evil twin), fostering catastrophic forgetting (3) lastly, we show that conventional continual learning algorithms *generally* cannot address the issue.

Notation and Setup We begin with the notations. Let $x \sim D$ denote the original sample (e.g., image) sampled from the dataset D, and let the finite transformation set $T = \{T_1, T_2, \cdots T_n\}$, represent the set of all n available augmentation transformations that can be applied to x. For simplicity, we assume there is a finite set of transformations that can be applied to the original sample, which reflects the reality of many existing augmentation algorithms [8, 10]

At the ith step of the training stage, random augmentation τ_{ra} selects a transformation $T_i \in T$ with a uniform probability, expressed as:

$$P_{\tau_{ra}}(T_i) = \frac{1}{n}, \forall T_i \in T. \tag{1}$$

The targeted augmentation τ_l selects a transformation T_l that minimizes a predefined objective $\mathcal{L}(\theta; T_i(x))$ – namely adversarial loss [40] – over the model parameters θ:

$$P_{\tau_l}(T_i) = \frac{e^{-\beta \mathcal{L}(\theta; T_i(x))}}{\sum_{j=1}^{n} e^{-\beta \mathcal{L}(\theta; T_j(x))}}, \tag{2}$$

where $\beta > 0$ is a hyperparameter that adjusts how sensitive the probabilities are influenced by the loss $\mathcal{L}$. For instance, setting $\beta = 0$ would make τ_l equivalent to τ_{ra}, as all the transformations would have equal probability. Due to this optimization, transformations under τ_l are selected based on alignment with $\mathcal{L}$, resulting in a non-uniform probability distribution.

3.1. Quantifying Augmentation Diversity: Random vs. Targeted Augmentation

In this section, we show that random augmentation provides a more diverse set of augmentations compared to the targeted augmentation strategies.

Comparing Augmentation Diversity To measure the diversity of the augmentation's outputs (i.e., augmentation diversity), we use the entropy H of the transformation distribution induced by τ_{ra} and τ_l. The entropy $H(P)$ over a discrete probability distribution P over transformations T is defined as:

$$H(P) = -\sum_{i=1}^{n} P(T_i) \log P(T_i), \tag{3}$$

where higher entropy values indicate greater stochasticity and thus, greater diversity in the transformations applied. By comparing the entropy of the two augmentation strategies, we can show that the entropy of targeted augmentation is lower than that of random augmentation.

Proof Sketch. Recall that τ_{ra} selects each transformation $T_i \in \{T_1, \ldots, T_n\}$ with uniform probability Eq. (1). Consequently, its entropy satisfies

$$H(\tau_{ra}) \;=\; -\sum_{i=1}^{n} \frac{1}{n} \log\left(\tfrac{1}{n}\right) \;=\; \log n. \tag{4}$$

In contrast, the targeted augmentation τ_l assigns probabilities based on the loss Eq. (2), leading to a *non-uniform* distribution and thus lower entropy. Consequently,

$$H(\tau_l) \;<\; \log n \;=\; H(\tau_{ra}), \tag{5}$$

which establishes that random augmentation τ_{ra} indeed induces higher entropy, and thus more diverse transformations, than the targeted augmentation τ_l. For the full derivation, refer to Appendix A.

> **Takeaway:** Random Augmentation introduces greater augmentation diversity than any targeted augmentation that relies on a loss objective.

3.2. The Impact of Augmentation Diversity on Catastrophic Forgetting

Next, we analyze how higher diversity from random augmentation can lead to increased catastrophic forgetting [22], compared to targeted augmentation methods with lower augmentation diversity. Specifically, we show that a high augmentation diversity fosters gradient collisions that erode the model's trained features during parameter updates.

Proof Sketch Let $g_x = \nabla_\theta L(\theta; x)$ be the gradient computed on the original sample x and $g_{T(x)} = \nabla_\theta L(\theta; T(x))$ be the gradient computed on the augmented sample $T(x)$. Assume that the augmentation introduces a perturbation δ such that

$$T(x) = x + \delta, \tag{6}$$

with δ drawn from a distribution whose variance reflects the diversity of the augmentation. Using a first-order Taylor expansion, we approximate

$$g_{T(x)} \approx g_x + M\,\delta, \tag{7}$$

where $M = \nabla_\theta \nabla_x L(\theta; x)$ is the matrix of mixed second-order derivatives.

We then analyze the alignment of the gradients via their cosine similarity:

$$\cos(\phi) = \frac{g_x^\top (g_x + M\,\delta)}{\|g_x\| \cdot \|g_x + M\,\delta\|}. \tag{8}$$

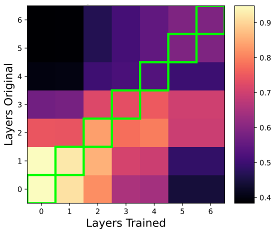


Figure 2. CKA Similarity between models trained with two distinct augmentations of the same dataset.

For random augmentation, the perturbation δ has high variance, which increases the magnitude of the term $M\,\delta$ and hence introduces greater fluctuations in $g_{T(x)}$. Consequently, when taking the expectation over δ, the expected cosine similarity $\mathbb{E}[\cos(\phi)]$ is lower, signifying that the gradients are less aligned (i.e., more likely to collide) compared to the case of targeted augmentation, where δ is smaller.

This misalignment—referred to as *gradient collision*—leads to destructive interference during parameter updates, which gradually erodes the model's learned features [5, 6, 48]. For the full derivation, please see Appendix B.

> **Takeaway:** The augmentation diversity of Random Augmentation exacerbates forgetting by triggering gradient collision.

3.3. Empirical Analysis of the Problem and Solutions

In this section, we provide an empirical analysis of the problem and seek practical ways to address it for the improvement of random augmentation. First, we empirically show the effect of the forgetting syndrome on the model's generalization. Specifically, we study (1) its effect on the prediction of the model, and (2) the model's learned features.

To assess the effect of the colliding augmentations on the catastrophic forgetting, we design an experiment that uses a simple transformation (e.g., rotation) to simulate "evil twin", a set of augmented samples that results in colliding gradients in the parameter space [48], and study the model's behavior when it is trained on such pairs. Specifically, we test a model's prediction accuracy before and after it is trained on a set of evil twin.

A formal definition of Evil Twin Let $\nabla f_1, \nabla f_2 \in \mathbb{R}^d$ be the gradients of the loss function with respect to the model parameters computed from different augmentations of the same data sample (or batch). In our notation, we write

$$\mathbf{g}_1 = \nabla f_1 \quad \text{and} \quad \mathbf{g}_2 = \nabla f_2. \tag{9}$$

We say that $(\mathbf{g}_1, \mathbf{g}_2)$ are *evil twin* if their parameter-wise signs differ extensively, thereby inducing catastrophic interference in the model's parameter updates.

Table 1. Effect of evil twin on catastrophic forgetting. Model sequentially trained on colliding augmentations (i.e., evil twin) suffer from significant forgetting of the previous augmentations, which limit augmentation's generalization effect.

Rotation	$\overline{\mathrm{SD}}$	x_1 Acc	Forgetting
0°	0.261	0.987	0.0013
-10°	0.272	0.985	0.0031
-20°	0.290	0.965	0.0236
-30°	0.306	0.921	0.0668
-40°	0.315	0.797	0.1916
-50°	0.330	0.624	0.3638
-60°	0.337	0.532	0.4557
-70°	0.340	0.389	0.5989
-80°	0.340	0.308	0.6803
-90°	0.341	0.252	0.7365

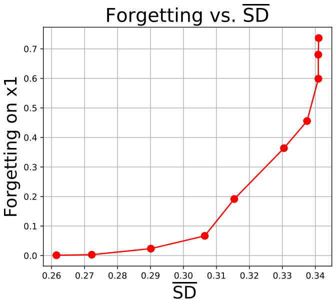


Figure 3. Aggregated Sign Discrepancy ($\overline{\mathrm{SD}}$) vs. Forgetting. Increase in the level of collision between augmented samples (measured as $\overline{\mathrm{SD}}$) coincides with the worsening of the forgetting.

Formally, we define the *sign discrepancy* between $\mathbf{g}_1$ and $\mathbf{g}_2$ as

$$\mathrm{SD}(\mathbf{g}_1, \mathbf{g}_2) \;=\; \frac{1}{d}\sum_{i=1}^{d} \mathbf{1}\Big(\mathrm{sign}(g_{1,i}) \neq \mathrm{sign}(g_{2,i})\Big), \quad (10)$$

where $g_{1,i}$ denotes the ith component of $\mathbf{g}_1$, and d is the total number of parameters.

In practice, to reduce noise from a single mini-batch, we aggregate the sign discrepancy over k mini-batches. Let $\{(\mathbf{x}_b, \mathbf{y}_b)\}_{b=1}^{k}$ be k mini-batches sampled from the same augmentation, and let $\mathbf{g}_{2,b}$ denote the gradient computed on batch b using the second augmentation. We then define the aggregated sign discrepancy as

$$\overline{\mathrm{SD}}(\mathbf{g}_1, \{\mathbf{g}_{2,b}\}_{b=1}^{k}) \;=\; \frac{1}{k}\sum_{b=1}^{k} \mathrm{SD}\big(\mathbf{g}_1, \mathbf{g}_{2,b}\big). \quad (11)$$

This aggregated measure $\overline{\mathrm{SD}}$ provides a smoother and more reliable indication of the level of parameter-wise sign conflict—hence, the extent of "Evil Twin" behavior—than that computed on a single mini-batch.

Effect on Prediction Using this measure (i.e., $\overline{\mathrm{SD}}$), we analyze the effect of the evil twin on the model's prediction in a formal fashion. Specifically, we demonstrate that two colliding augmentations x_1, x_2 of an original sample x can distort the learned features of each other, causing them to forget what it has learned previously. To show this, we design a simple experiment using the MNIST dataset, where the model is first trained on a image rotated clock-wise $45°$ (x_1), and then re-trained on another rotated image (x_2) with a different rotation value ranging from $45°$ to $-45°$ ($0°$ to $-90°$ compared to x_1). We then analyze the drop in x_1 accuracy (i.e., forgetting) and measure the aggregated sign discrepancy ($\overline{\mathrm{SD}}$), where $k = 10$, meaning that we measure the $\overline{\mathrm{SD}}$ over 10 mini-batches. The results are reported in Table 1 and Figure 3.

We can see a general trend in Table 1. As the aggregated sign discrepancy (i.e., evil twin metric) increases, the accuracy on the previously learned samples decreases (i.e., forgetting increases). While our analysis is based on rotation-based augmentations on MNIST and focuses on the effect of a single prior augmentation, the aggregated sign discrepancy metric is generalizable as is rooted in the fundamental dynamics of gradient-based optimization, quantifying parameter-level conflicts independent of augmentation type, and averaging over multiple mini-batches reduces noise.

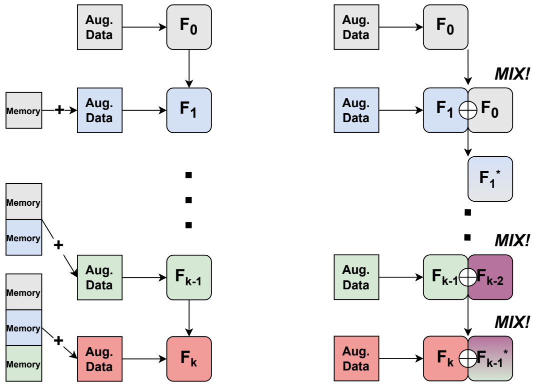
Effect on Learned Features Next, we analyze how the model's learned features are altered during training with random augmentation, namely by using the Centered Kernel Alignment (CKA) similarity metric [23]. Specifically, we compare the intermediate output features of the model that is trained with augmented MNIST but with distinct, non-overlapping augmentations. In Fig. 2, we visualize the results, where the diagonal cells of the box indicate the feature similarity between the corresponding layers between the two models, bright colors indicating high feature similarity. As we can see in Fig. 2, the features learned from augmented data diminish as the model is exposed to distinct augmentations.

Solutions We then seek ways to address forgetting. Our initial idea is to view the evil twin effect as a continual learning (CL) problem. Naturally, we start by using an existing CL algorithm that explicitly mitigates forgetting.

Specifically, we implement a simple replay algorithm that saves subsets of augmented samples from previous steps to explicitly tackle forgetting via exemplar memory replay [36]. We denote this as the *memory* method in our experiments. We find that the memory method effectively boosts performance while stabilizing the learning process (see Fig. 6). In Sec. 5, we have shared the details of the experimental results. However, this approach has some limitations. First, replay

Algorithm 1: Our Method

```
Input: Task model $f$ with parameters $\theta$, source data $D_s$, snapshot
   interval $k$;
Output: Fully updated task model $f$ and its parameters $\theta$
Save snapshot $\theta_s \leftarrow \theta$
while not converge do
    for $i = 1 : n_{iterations}$ do
        if $i\%k = 0$ then
            Select layers to merge based on rank
            $\theta \leftarrow \text{MERGE}(\theta, \theta_s)$
            Update snapshot $\theta_s \leftarrow \theta$
            Randomly select augmentation $g$
        Sample $i$-th mini-batch from data $D_s$
        Augment the mini-batch using $g$
        Train $f$, update $\theta$
```



(a) Memory Method (b) Ours

Figure 4. Replay memory helps address forgetting. In our scenario, we substitute replay memory with model weights.

memory is only effective when the discrepancy between the augmented data and the original data is significant. In other words, when the gap between the augmented sample and the original sample is not significant, they are not reliable. Furthermore, to elicit stronger generalization, more than 50% of the augmented samples were needed to be stored as the replay memory. In the sequel, we present a simple idea to address the forgetting problem, namely by substituting the memory samples with model weights.

4. Method

Notation and Setup Our objective is to train a robust model $F = E \circ C$ using only the source domain dataset D_s. The model is composed of an encoder $E : \mathcal{X} \rightarrow \mathcal{H}$, and a classification head $C : \mathcal{H} \rightarrow \mathcal{Y}$. Additionally, we may implement a projection head $P : \mathcal{H} \rightarrow \mathcal{P}$. Specifically, the encoder receives the input image x and returns a feature representation h, while the classification head predicts the prediction logits $\hat{y}$ from h. In this process, we will use an augmentation module G that provides augmented samples $\bar{x}$ from the original sample x. Previous works that utilize learnable augmentation methods commonly utilize the predictions of the training model $F = E \circ C$ to update the augmentation module [40]. On the other hand, our method utilizes random augmentation [8], which alleviates the need for additional training costs.

Proposed Method Let $\theta \in \mathbb{R}^N$ be the current weights of the model and $\theta_s \in \mathbb{R}^N$ be the snapshot of the weights saved k iterations earlier. Define the absolute difference for each parameter as

$$d_i = \big|(\theta)_i - (\theta_s)_i\big|, \quad i = 1, \dots, N. \tag{12}$$

We then construct a binary mask $\mathbf{m} \in \{0, 1\}^N$ by selecting the indices corresponding to the top $p\%$ (in our case, $p = 80$) of the differences:

$$m_i = \begin{cases} 1, & \text{if } d_i \geq \tau, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where the threshold τ is chosen so that

$$\frac{1}{N} \sum_{i=1}^{N} m_i = \frac{p}{100}. \tag{14}$$

The weight merging is then performed via a selective averaging of the current weights with the snapshot:

$$\theta \leftarrow \mathbf{m} \odot \frac{\theta + \theta_s}{2} + (1 - \mathbf{m}) \odot \theta, \tag{15}$$

where $\odot$ denotes element-wise multiplication.

This expression updates only the 80% of the parameters that have experienced the largest shifts during training, effectively substituting replay memory with model weight averaging to counteract forgetting (the evil twin effect).

Our algorithm leverages the continual learning effects of model merging [32, 38], to address forgetting. Specifically, we accomplish this by averaging the model's weights with the weight snapshot θ_s saved k iterations prior. Our method utilizes θ_s as a substitute for replay memory, explicitly regularizing the model in the weight space. The method is graphically illustrated in Fig. 4, and its algorithm is reported in Algorithm 1. The key difference between our method to other model merging methods [24, 32] is that our algorithm selectively merges the parameters that have shown large changes during training, which we assume that forgetting has occurred. Please note that this naive measure is possible owing to our unique setting (i.e., single source domain). Specifically, we selectively merge 80% of the parameters that have shifted the most during training (compared to the snapshot θ_s). The merging percentage was found using a grid search (see Tab. 4).

Table 2. Target domain accuracy on PACS and Digits.

	PACS				Digits				
Method	A	C	S	Avg.	SVHN	M-M	S-D	USPS	Avg.
ERM	54.43	42.74	42.02	46.39	27.83	52.72	39.65	76.94	49.29
ADA [11]	58.72	45.58	48.26	50.85	35.51	60.41	45.32	77.26	54.62
M-ADA [34]	**58.96**	44.09	49.96	51.00	42.55	67.94	48.95	78.53	59.49
L2D [44]	56.26	**51.04**	58.42	55.24	62.86	87.30	63.72	83.97	74.46
PDEN [29]	57.41	45.77	65.01	56.06	62.21	82.20	69.39	85.26	74.77
MetaCNN [41]	54.05	53.58	63.88	57.17	**66.50**	**88.27**	70.66	89.64	78.76
RandAug. [8]	49.05	41.77	57.52	49.45	57.78	77.09	65.42	87.05	71.84
RandAug. + Mem.	52.22	42.37	62.11	52.23	58.44	77.03	69.92	89.27	73.67
Ours	56.95	47.99	**70.20**	**58.38**	63.93	77.79	**80.54**	**93.42**	**78.92**

Table 3. Target domain accuracy on larger datasets.

	Office-Home				VLCS				Terra Incognita			
Method	Art	Clipart	Product	Avg.	L	C	S	Avg.	L38	L43	L46	Avg.
ERM	52.78	40.19	68.73	53.90	59.06	97.30	**74.25**	76.87	22.90	**15.85**	22.91	20.55
L2D* 44	54.02	41.77	66.30	54.03	56.21	95.52	66.90	72.87	34.82	14.23	21.76	23.60
PDEN* 29	53.39	43.38	66.25	54.34	62.55	96.11	73.52	77.39	**37.52**	14.93	20.80	24.42
RandAug. * 8	43.10	45.47	61.67	50.01	57.58	93.18	66.56	72.44	36.41	12.80	20.41	23.21
RandAug. + Mem.	46.18	45.83	64.09	52.03	59.56	95.14	66.83	73.84	36.64	12.61	22.75	24.00
Ours*	**55.94**	**53.10**	**69.33**	**59.46**	**65.82**	**98.01**	70.69	**78.17**	37.45	15.29	**25.63**	**26.12**

5. Experiment

5.1. Experimental Setting

Dataset There are a number of single-source domain generalization benchmarks, many of them deriving from domain generalization benchmarks.

PACS [27] consists of four domains of varying styles (Photo, Art, Cartoon, and Sketch) with 7 classes. Following previous works, we train our model with the Photo domain and evaluate the remaining target domains. We use the train/test split provided by the original paper [27]. **Digits** is also a very common sDG benchmark, composed of 5 different digit classification datasets, MNIST [9], SVHN [33], MNIST-M [14], SYNDIGIT [13], USPS [26]. In our experiment, we train our model with the first 10,000 samples of the MNIST dataset and assess the model's average accuracy across the remaining four domains.

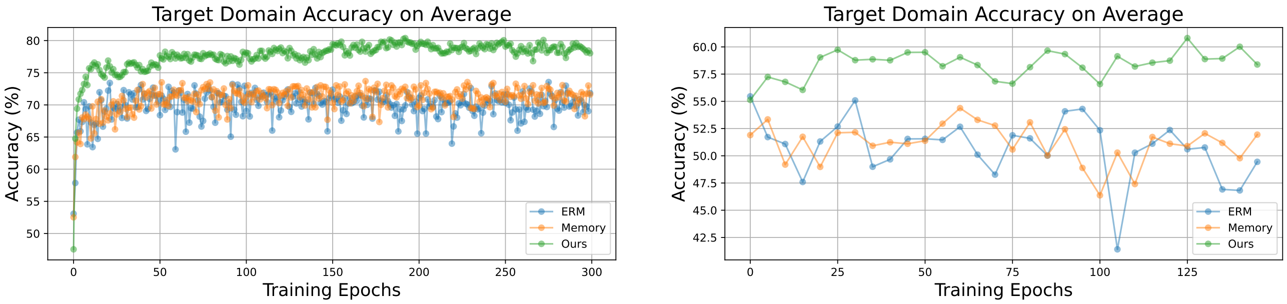
We also include larger benchmarks e.g., Office-Home [39] VLCS [12] and Terra Incognita [3]. **Office-Home** is a common multi-DG benchmark consisting of 4 datasets (Real-world, Art, Clipart, Product) with differing styles with 65 classes. We train on the Real-world domain and evaluate the remaining domains. **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. **Terra Incognita** is comprised of 4 datasets of wildlife photos taken in different locations (L38, L43, L46, L46). We used the L38 dataset as the source and the rest as targets.

Model Architecture Following previous works on sDG, we match the model architecture for a fair comparison with existing methods [28, 34, 41]. Specifically, for PACS, we use an AlexNet [25] architecture, while for Digits, we use a 3-layer MLP. For larger benchmarks, we used the ResNet-(18/50) [19] architecture.

Evaluation metrics Following previous sDG works, we evaluate the model's average accuracy across multiple target domain datasets. We will also report the accuracy of each target domain. Please refer to Tab. 2 for a better understanding of the evaluation metric.

5.2. Main Results

We share our main experimental results in Tab. 2. In PACS and Digits, our method with Random Augmentation displayed strong average accuracy, outperforming many previous works that utilize complex, learnable augmentation modules. More importantly, it is worth noting that a simple model merging procedure can significantly boost the generalization effect of random augmentation (49.45 → 58.38 in PACS, 71.84 → 78.03). Furthermore, we share the experimental results on larger benchmarks in Tab. 3. Please note that the Office-Home, VLCS, and Terra Incognita are not conventional benchmarks in the sDG literature, hence we could only compare with methods that we could replicate.



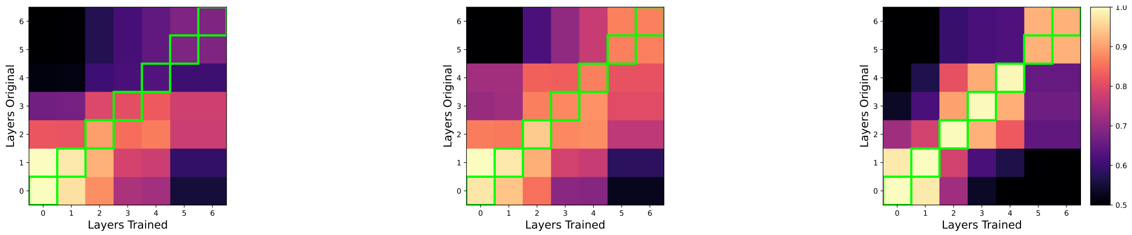
(a) Digits

(b) PACS

Figure 5. Average target domain accuracy on the Digits (left) and PACS (right) datasets.

Table 4. Effect of selective merging on target domain accuracy

	Datasets				
Merge %	PACS	Digits	Office-Home	VLCS	Terra Incognita
Full Merge	77.82	57.01	59.23	76.95	24.47
Top 20%	75.11	56.48	57.94	75.84	24.71
Top 40%	76.44	57.45	59.80	76.82	24.32
Top 60%	77.69	58.11	58.31	77.85	26.48
Ours (top 80%)	78.92	58.38	59.46	78.17	26.12



(a) Without Memory.

(b) With Memory.

(c) With Ours.

Figure 6. Layer-wise Feature Similarity (CKA) of the model between two models trained with two distinct augmentations of the same dataset. (1) Vanilla training (2) Training with replay memory (3) Training with our method. Brighter diagonal boxes indicate the robustness against forgetting.

Nevertheless, in all benchmarks, our selective merging approach showed effectiveness in boosting the generalization effect of random augmentation, showing competitiveness against all competing methods.

Furthermore, we report the results of the ablation study of our selective merging process. Specifically, we perform a study on the percentage of how many parameters are merged. The results of this experiment are shared in Tab. 4. We observe that the selective merging of the top $60-80\%$ parameter (ranked by the activeness of the parameter) outperforms others. However, there is a lack of understanding of why such a simple metric can be used as a selection criterion for merging. We believe further study is required on how to design reliable criteria for selective merging.

Lastly, we show the effect of continual learning algorithms (e.g., replay memory, ours) on preserving the trained features. Similar to Fig. 2, we measure the layer-wise feature similarity between two models (a) a model trained with a set of random augmentations T_1, and (b) the first model retrained on a set of non-identical set of augmentations T_2. Using this, we can show the model's robustness against forgetting induced by random augmentation. Specifically, we compare three cases (1) vanilla training without replay memory, (2) training with replay memory, and (3) training with our merging method. In Fig. 6, we show that our method is significantly better in addressing the forgetting (or feature distortion).

6. Limitations & Future Work

In this section, we discuss the limitations of our work and propose future directions for our research. We begin this discussion by mentioning some remaining limitations. First, our theoretical framework currently assumes a finite, discrete set of augmentation transformations (Appendix A). This assumption may restrict the full potential of augmentation diversity. Extending our framework to allow more expressive perturbations—such as perturbing latent representations directly—could provide a richer variety of augmented samples, potentially improving out-of-distribution robustness further. Additionally, our experiments have primarily focused on classification benchmarks with limited domain diversity. Future work could examine more realistic or large-scale datasets, as well as non-classification tasks (e.g., detection or segmentation), to evaluate whether the proposed solution generalizes to more complex settings. Exploring these directions could yield deeper insights into how random data augmentation can be harnessed for more robust and versatile model training. Lastly, our partial merging method relies on some degree of heuristics that require manual adjustments (e.g. the measure of the parameter drift Eq. (12), the selection of the top $p\%$ parameters to be reset). We can think of an adaptive approach that would improve the method's robustness. We aim to reflect these ideas in our final copy.

7. Concluding Remarks

In this paper, we investigate why random augmentation underperforms in enhancing generalization in a limited data setting. Our findings reveal that the stochastic nature of random augmentation induces significant catastrophic forgetting during training. While continual learning strategies can mitigate this issue, their naive application in single-source domain generalization has limitations. To address this, we proposed a simple model merging method that counteracts this phenomenon, effectively leveraging random augmentation for improved generalization. Our approach achieved strong performance across multiple benchmarks—PACS, Digits, VLCS, Office-Home, and Terra Incognita-while significantly reducing computational costs.

References

[1] Masih Aminbeidokhti, Fidel A Guerrero Pena, Heitor Rapela Medeiros, Thomas Dubail, Eric Granger, and Marco Pedersoli. Domain generalization by rejecting extreme augmentations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2215–2225, 2024. 1, 2

[2] Randall Balestriero, Ishan Misra, and Yann LeCun. A data-augmentation is worth a thousand samples: Analytical moments and sampling-free training. *Advances in Neural Information Processing Systems*, 35:19631–19644, 2022. 2

[3] Sara Beery, Grant van Horn, and Pietro Perona. Recognition in terra incognita, 2018. 6

[4] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *International Conference on Machine Learning*. PMLR, 2019. 2

[5] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020. 3

[6] Dong Kyu Cho, Inwoo Hwang, and Sanghack Lee. Peer pressure: Model-to-model regularization for single source domain generalization, 2025. 2, 3

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 2

[8] 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. 2, 5, 6

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

[10] Nikos Efthymiadis, Giorgos Tolias, and Ondřej Chum. Crafting distribution shifts for validation and training in single source domain generalization, 2024. 1, 2

[11] 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. 6

[12] 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. 6

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015. 6

[14] 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. 6

[15] Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. *arXiv preprint arXiv:2210.06441*, 2022. 2, 11

[16] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 2

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 2

[18] Liangxuan Guo, Yang Chen, and Shan Yu. Out-of-distribution forgetting: vulnerability of continual learning to intra-class distribution shift. *arXiv preprint arXiv:2306.00427*, 2023. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[20] Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Data augmentation revisited: Rethinking the distribution gap between clean and augmented data, 2019. 2

[21] 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. 12

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1, 2, 3

[23] 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. 4

[24] Jędrzej Kozal, Jan Wasilewski, Bartosz Krawczyk, and Michał Woźniak. Continual learning with weight interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4187–4195, 2024. 5, 12

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 6

[26] 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. 6

[27] 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. 6

[28] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in neural information processing systems*, 33:3118–3129, 2020. 6

[29] 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, 6

[30] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 11

[31] Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *IJCAI*, pages 2182–2188, 2022. 2

[32] Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. *arXiv preprint arXiv:2312.08977*, 2023. 5

[33] 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. 6

[34] 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. 1, 6

[35] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:10821–10836, 2022. 12

[36] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019. 2, 4

[37] Mohammad Rostami, Soheil Kolouri, and Praveen K Pilly. Complementary learning for overcoming catastrophic forgetting using experience replay. *arXiv preprint arXiv:1903.04566*, 2019. 2

[38] Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9677–9685, 2023. 1, 5, 12

[39] 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. 6

[40] 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, 5

[41] 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. 6

[42] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. *arXiv preprint arXiv:2204.04662*, 2022. 2

[43] L Wang, X Zhang, H Su, and J Zhu. A comprehensive survey of continual learning: Theory, method and application (2023). *arXiv preprint arXiv:2302.00487*, 1(5), 2023. 2

[44] 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, 6

[45] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks, 2016. 12

[46] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V. Le. Adversarial examples improve image recognition, 2020. 2

[47] 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

[48] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020. 3

[49] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020. 2

[50] 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. 1, 2

A. Quantifying Augmentation Diversity: Random vs. Targeted Augmentation

Entropy of Random Augmentation and Targeted Augmentation For a random augmentation τ_{ra}, each transformation is selected with uniform probability $\frac{1}{n}$. Therefore, the entropy (augmentation diversity) in this case is:

$$H(\tau_{ra}) = -\sum_{i=1}^{n} \frac{1}{n} \log\left(\frac{1}{n}\right) = \log n. \tag{16}$$

This entropy is maximized since all transformations in T are equally probable under τ_{ra}, resulting in the highest possible diversity in transformation selection.

For targeted augmentation τ_l, transformations are selected based on their loss values $\mathcal{L}(\theta; T_i(x))$, resulting in a non-uniform probability distribution. The entropy here is:

$$H(\tau_l) = -\sum_{i=1}^{n} P_{\tau_l}(T_i) \log P_{\tau_l}(T_i). \tag{17}$$

As $P_{\tau_l}(T_i)$ assigns higher probabilities to transformations that optimize $\mathcal{L}$, the distribution is skewed, leading to $H(\tau_l) < H(\tau_{ra}) = \log n$. The exact value of $H(\tau_l)$ depends on the loss values and the hyperparameter β. As β increases, the distribution becomes sharper, further reducing $H(\tau_l)$.

Conclusion Since $H(\tau_l) < H(\tau_{ra}) = \log n$, the entropy of the targeted augmentation is lower than that of the random augmentation. This indicates that random augmentation τ_{ra} provides greater diversity by uniformly exploring the full transformation set T, whereas targeted augmentation τ_l focuses on a subset of transformations that align with the objective $\mathcal{L}$. This reduced diversity in τ_l can lead to less exploration during training, potentially resulting in better optimization to $\mathcal{L}$ but possibly at the cost of the generalization effect. Conversely, the higher diversity in τ_{ra} may enhance generalization by exposing the model to a wider range of augmented samples, while it could also exacerbate unexpected issues, namely catastrophic forgetting.

B. The Impact of Augmentation Diversity on Catastrophic Forgetting

Recap on Augmentation Diversity In Sec. 3 and Appendix A, we showed that the random augmentation τ_{ra} uniformly selects the transformations from all n possible transformations, resulting in diverse augmented samples $T(x)$. Consequently, the gradients $\nabla_\theta(\theta; T(x))$ have high variance due to the diverse $T(x)$. On the other hand, the targeted augmentation τ_l selects its transformations based on the loss $\mathcal{L}$, leading to augmented samples $T(x)$ that are limited and closer to the original sample x e.g., label-preserving augmentations [15]. This results in a lower variance in the gradients (i.e. gradient alignment).

Connecting Augmentation Diversity to Forgetting: Gradient Collision Now, we investigate how increased augmentation diversity can lead to a higher chance of catastrophic forgetting by focusing on gradient collision during training.

When training a model with parameters θ on augmented data, the parameter update is given by: $\Delta\theta = -\eta\nabla_\theta\mathcal{L}(\theta; T(x))$, where η is the learning rate, and $T(x)$ the transformed version of x using the augmentation set T. Let $g_x = \nabla_\theta\mathcal{L}(\theta; x)$ denote the gradient with respect to x, and $g_{T(x)} = \nabla_\theta\mathcal{L}(\theta; T(x))$ denote the gradient with respect to the augmented data $T(x)$.

To measure the alignment of the gradients g_x and $g_{T(x)}$, we use the cosine similarity, formulated as: $\cos(\phi) = \frac{g_x^\intercal g_{T(x)}}{\|g_x\| \cdot \|g_{T(x)}\|}$, where $\cos(\phi) > 0$ indicates aligned gradients, and $\cos(\phi) < 0$ indicates gradient collision, which can contribute to catastrophic forgetting [30].

Assuming the transformation $T(x)$ introduces a perturbation δ on the input x, we can write:

$$T(x) = x + \delta, \tag{18}$$

where δ is a random variable with zero mean ($\mathbb{E}[\delta] = 0$) and covariance $\sum_T$. Here, the covariance matrix $\sum_T$ describes how the perturbations vary across different dimensions of the input space, its diagonal elements representing the variance of δ in each input dimension i.

Using a first-order Taylor expansion around x, we approximate $g_{T(x)}$ as:

$$g_{T(x)} = \nabla_\theta\mathcal{L}(\theta; x+\delta) \approx \nabla_\theta(\theta; x) + \nabla_\theta\nabla_x\mathcal{L}(\theta; x)\delta = g_x + M\delta, \tag{19}$$

where $M = \nabla_\theta\nabla_x\mathcal{L}(\theta; x)$ is a matrix representing the mixed second-order derivatives.

Grounding on this, we analyze the expected cosine similarity (alignment) over δ. We derive this by plugging in Eq. (19) to the cosine similarity $cos(\phi)$:

$$\mathbb{E}_\delta[\cos(\phi)] = \mathbb{E}_\delta\left[\frac{g_x^\intercal(g_x + M\delta)}{\|g_x\| \cdot \|(g_x + M\delta)\|}\right]. \tag{20}$$

While the exact computation is complex, the trend of the expected cosine similarity can be analyzed by expanding the numerator and the denominator.

Numerator of Eq. (20)

$$g_x^\intercal(g_x + M\delta) = \|g_x\|^2 + g_x^\intercal M\delta. \tag{21}$$

Since the expected value $\mathbb{E}_\delta[g_x^\intercal M\delta] = g_x^\intercal M\mathbb{E}[\delta] = 0$ is zero, the expected numerator is $\|g_x\|^2$. The variance of $g_x^\intercal M\delta$ is:

$$Var[g_x^\intercal M\delta] = g_x^\intercal M \sum\nolimits_T M^\intercal g_x. \tag{22}$$

Denominator of Eq. (20) In $\|g_x\| \cdot \|(g_x + M\delta)\|$,

$$\|(g_x + M\delta)\| = \sqrt{\|g_x\|^2 + 2g_x^\intercal M\delta + \|M\delta\|^2}. \tag{23}$$

The expected value of $\|(g_x + M\delta)\|$ increases with the variance introduced by δ, specifically due to $\|M\delta\|^2$. Consider a case where the transformation covariance $\sum_T$ of δ increases:

- Numerator: The variance of $g_x^\intercal M\delta$ increases, causing greater fluctuations around the expected value $\|g_x\|^2$.
- Denominator: The term $\|M\delta\|^2$ increases, leading to a larger denominator.

Overall, the expected cosine similarity $\mathbb{E}_\delta[\cos(\phi)]$ (Eq. (20)) decreases because the denominator grows faster than the numerator. This decrease indicates increased gradient collision.

Conclusion An increase in the variance $\sum_T$ of the transformation perturbations δ leads to decreased expected cosine similarity between g_x and $g_{T(x)}$. This misalignment of gradients can result in gradient collision, effectively increasing the risk of catastrophic forgetting during parameter updates [38]. Therefore, higher augmentation diversity of random augmentation, characterized by larger perturbations, may exacerbate forgetting, whereas targeted augmentation with lower diversity suffers less from this issue.

C. Why is model merging effective for continual learning in a single domain?

The continual learning effect of model merging (i.e. weight-averaging) has been discussed in the recent continual learning literature [24, 38]. In general, combining the weights of neural networks trained on different tasks enables the model to retain knowledge from previous tasks while learning new ones.

Weight-averaging helps in creating a parameter space that balances the representations learned from different tasks. By averaging the weights, the model finds a middle ground in the parameter landscape that is effective for multiple tasks [21]. This results in a smoother loss surface, which enhances the model's generalization capabilities. Furthermore, model merging is analogous to ensemble methods, where combining multiple models leads to improved performance and robustness [35]. In the context of continual learning, merging models trained on different tasks can be seen as creating an ensemble that captures diverse features from all tasks.

Moreover, we view that weight averaging acts as an implicit regularizer, preventing the model from becoming too specialized on any particular augmented version of the data [45]. This regularization is crucial for improving the model's performance on unseen data and ensuring stability during training. In summary, while augmented data does not constitute different tasks, model merging through weight averaging remains effective for continual learning within the same task. It facilitates the integration of knowledge from various data augmentations, enhancing the model's generalization capabilities and robustness.